```
                        ; *****
                        ; *
                        ; * ADD-WHATEVER GRAPHICS AND I/O EXTENSIONS
                        ; *
                        ; *  PROGRAMMED BY
                        ; *
                        ; *    JAY FENTON
                        ; *
                        ; * FEB - MAR 1979
                        ; *
                        ; *****
                                .XLINK
                                .PABS
                                .PHEX
                        ; *****
                        ; *
                        ; * VERY IMPORTANT EQUATES !!
                        ; * CHANGE THESE ANY TIME AL MCNEILS VERSION IS
                        CHANGED!!
                        ; *
                        ; *****
  0CA0                  GRADDR=0CA0H    ; ** START OF GRAPHICS GOODIES *
                        *
  8000                  RAMBEG=08000H   ; ** RAM AREA START **
  8060                  DPADDR=08060H   ; ** ADDR OF DICTIONARY POINTER
                        **
  8080                  DPVAL=08080H    ; ** VALUE OF DICTIONARY POINTER
                         **
  8064                  LASTADDR=08064H ; ** ADDR OF LAST VAR IN TERSE *
                        *
  0C86                  .LINK.=0C86H    ; ** VALUE OF LAST FROM TERSE **

                        ; *****
                        ; *
                        ; * TERSE MACROS
                        ; *
                        ; *****
                        ; SET IY = SP, CREATING A POINTER TO STACK FRAME

                        .DEFINE FRAME=[
                                PUSH    Y
                                LXI     Y,0
                                DADY    SP]
                        ; RESTORE OLD IY FROM STACK
                        .DEFINE UNFRAME=[
                                POP     Y]
                        .DEFINE NEXT=[
                                PCIY]
                        .DEFINE SYSTEM[STRING,%LEN]=[
                                .BYTE   %LEN-.-1
                                .ASCII  [2EH] STRING [2EH]
                        %LEN:]
                        .DEFINE VERB[STRING,%LEN]=[
                                .WORD   .LINK.
                        .LINK.=.-2
```

```
                                 .BYTE    %LEN-.-1
                                 .ASCII   STRING
                        %LEN:]
                        .DEFINE IMMED [STRING,%LEN]=[
                                 .WORD    .LINK.
                        .LINK.=.-2
                                 .BYTE    %LEN-.-1+80H
                                 .ASCII   STRING
                        %LEN:]
                        .DEFINE DHEAD=[
                                RST       1]
                        .DEFINE DEND=[
                                 .WORD    %END]
                        .DEFINE CONSTANT[NAME,VALUE,LABEL]=[
                                VERB      "NAME"
                        .IFB    [LABEL],[
                        NAME:][
                        LABEL:] CALL      CONST
                                 .WORD    VALUE]
                        .DEFINE VARIABLE[NAME,VAL(0)]=[
                                VERB      "NAME"
                        NAME:   CALL      SHI
                                 .WORD    VAL
                        ]
                        .DEFINE TEXT [STR]=[
                        ..S:     .ASCII   [..E-..S-1]STR
                        ..E=     .
                        ]
                                 .DEFINE WASTE[TIME,%LAB]=[
                                MVI       A,TIME
                        %LAB:   DCR       A
                                JRNZ      %LAB]
                        ; *****
                        ; *
                        ; * IO PORT EQUATES
                        ; *
                        ; *****
  000C                  MAGIC=0CH
  000D                  INFBK=0DH
  000E                  INMOD=0EH
  000F                  INLIN=0FH
  0019                  XPAND=19H
  0099                  TAPEIO=99H       ; PORT USED FOR BIT BANGER AUDIO
                         INTERFACE
                        ; BITS AND MASKS FOR MAGIC REGISTER VALUES
                        ;
  0003                  MRXPND=3         ; MR EXPAND BIT NUMBER
  0020                  XORWMR=20H       ; XOR MASK
  0008                  XPWMR=8H         ; EXPAND MASK
                        ; *****
                        ; *
                        ; * SPECIAL CHARACTERS
                        ; *
                        ; *****
  0009                  TAB=9
```

```
 001B                    ESC=1BH
 0022                    DQUOTE=22H
 0027                    SQUOTE=27H
 005E                    UPA=5EH
 005F                    LEFTA=5FH
 0000                    DOWNA=0
 0000                    FORWA=0
 0000                    XORCHR=0         ; CHARACTER CODE FOR KEYBOARD ´X
                         OR´ KEY
 000D                    NL=0DH           ; CARRAIGE RETURN
 000A                    LF=0AH           ; LINE FEED
 005F                    RUBKEY=5FH       ; RUBOUT CODE FOR KEYBOARD READ
 0018                    LINKIL=´X´-40H   ; LINE KILL
 0008                    RUBOUT=08H       ; RUBOUT CODE FOR CHAR DISPLAY
 00FD                    LOCK=0FDH        ; SHIFT LOCK KEY CODE
 00FE                    FLIP=0FEH        ; UPPER-LOWER SENSE SHIFT
 00FF                    BREAK=0FFH       ; PANIC BUTTON
                         ;
                         ; *****
                         ; *
                         ; * OTHER EQUATES FOR KEYBOARD SCANNER
                         ; *
                         ; *****
 0044                    SHKMSK=01000100B         ; SHIFT KEY POSITIONS IN
                          COL 7
 0030                    CNTMSK=00110000B         ; CONTROL KEY POSITIONS
                         IN COL 7
 0003                    KEYTRV=3                 ; DEBOUNCE TIME CONSTANT

 0020                    KEYBSZ=32                ; SIZE OF KEYBOARD INPUT
                          BUFFER
                         ; *****
                         ; *
                         ; * DISPLACEMENTS FOR PARAMETERS IN STACK FRAME
                         CREATED
                         ; * BY ´FRAME´ MACRO
                         ; *
                         ; *  FOR EXAMPLE, TO GET THE FIFTH PARAMETER INT
                         O DE:
                         ; *  MOV E,P5(Y)
                         ; *  MOV D,P5+1(Y)
                         ; *
                         ; *****
 0002                    FR.P1=2
 0004                    FR.P2=4
 0006                    FR.P3=6
 0008                    FR.P4=8
 000A                    FR.P5=10
 000C                    FR.P6=12
 000E                    FR.P7=14
 0010                    FR.P8=16
                         ; DISPLACEMENTS FOR PARAMETERS TO ´BOX´ COMMAND
 000A                    BX.X=10
 0008                    BX.Y=8
 0006                    BX.XS=6
```

```
  0004                     BX.YS=4
  0002                     BX.MOD=2
                           ; AND PARAMETERS TO CLIP SUBROUTINE
  0006                     CLP.S=6
  000A                     CLP.C=10
                           ; FIELDS IN WINDOW DESCRIPTOR TABLE
  0000                     WXR=0    ; XRIGHT
  0002                     WXL=2    ; X LEFT
  0004                     WYU=4    ; Y UPPER
  0006                     WYL=6    ; Y LOWER
                           ; *****
                           ; *
                           ; * MISC STUFF
                           ; *
                           ; *****
  0028                     BYTEPL=40                ; BYTES PER LINE OF DISP
                           LAY
  0FFF                     URINAL=0FFFH             ; STUPID MEMORY CELL
  4FFF                     CRAPPER=4FFFH
                           ; *****
                           ; *
                           ; * PLUG IN JUMP TO INITIALIZATION CODE
                           ; *
                           ; *****
  0100                             .LOC    100H
  0100    C3 0CA0                  JMP     POWERUP
  0CA0                             .LOC    GRADDR
                           ; *****
                           ; *
                           ; * RESET INITIALIZATION
                           ; *
                           ; *****
  0CA0                     POWERUP:
  0CA0    F3                       DI
  0CA1    AF                       XRA     A
  0CA2    D30E                     OUT     INMOD
  0CA4    21 0CE0                  LXI     H,INICOL        ; INITIALIZE COL
                           OR MAP REGISTERS
  0CA7    0E0A                     MVI     C,0AH
  0CA9    7E               ..OUTR: MOV     A,M
  0CAA    ED79                     OUTP    A
  0CAC    23                       INX     H
  0CAD    0D                       DCR     C
  0CAE    F2 0CA9                  JP      ..OUTR
  0CB1    21 1CF6                  LXI     H,INIVAL        ; ZERO OUT VARIA
                           BLES
  0CB4    11 8000                  LXI     D,RAMBEG
  0CB7    01 000C                  LXI     B,ZEROUT-RAMBEG
  0CBA    EDB0                     LDIR
  0CBC    063E                     MVI     B,ZERSIZ
  0CBE    AF                       XRA     A
  0CBF    EB                       XCHG
  0CC0    77               ..CLR:  MOV     M,A
  0CC1    23                       INX     H
  0CC2    10FC                     DJNZ    ..CLR
```

```
 0CC4    21 1CE8                LXI     H,LSTLNK        ; INITIALIZE 'LA
                         ST' POINTER
 0CC7    22 8064                SHLD    LASTADDR
 0CCA    21 0D18                LXI     H,INTVEC        ; SET INTERRUPT
                         REGISTERS
 0CCD    7C                     MOV     A,H
 0CCE    ED47                   STAI
 0CD0    7D                     MOV     A,L
 0CD1    D30D                   OUT     INFBK
 0CD3    ED5E                   IM2
 0CD5    3E08                   MVI     A,8
 0CD7    D30E                   OUT     INMOD
 0CD9    AF                     XRA     A
 0CDA    D30F                   OUT     INLIN
 0CDC    FB                     EI
 0CDD    C3 0103                JMP     103H    ; ENTER AL'S INITIALIZAT
                         ION CODE
                         ; *****
                         ; *
                         ; * TABLE OF INITIAL COLOR VALUES
                         ; * STORED IN REVERSE ORDER
                         ; *
                         ; *****
 0CE0                    INICOL:
 0CE0    CC                     .BYTE   0CCH    ; VERT BLANK
 0CE1    2C                     .BYTE   2CH     ; HORI BOUND
 0CE2    00                     .BYTE   0       ; CON-COM
 0CE3    08                     .BYTE   8       ; 7
 0CE4    5B                     .BYTE   5BH     ; 6
 0CE5    A5                     .BYTE   0A5H    ; 5
 0CE6    07                     .BYTE   7       ; 4
 0CE7    08                     .BYTE   8       ; 3
 0CE8    5B                     .BYTE   5BH     ; 2
 0CE9    A5                     .BYTE   0A5H    ; 1
 0CEA    07                     .BYTE   7       ; 0
                         ; *****
                         ; *
                         ; * CONSTANT ROUTINE - LOCAL TO THIS MODULE
                         ; *  DELETE THIS GUY WHEN YOU RUN OUT OF SPACE
                         ; *  HE IS DUPLICATED IN AL'S MODULE
                         ; *
                         ; *****
 0CEB                    CONST:
 0CEB    E1                     POP     H
 0CEC    5E                     MOV     E,M
 0CED    23                     INX     H
 0CEE    56                     MOV     D,M
 0CEF    D5                     PUSH    D
                                NEXT[
 0CF0    FDE9          +        PCIY]
                         ; *****
                         ; *
                         ; * INCREMENT INTERRUPT LOCKOUT FLAG
                         ; * CALLED BY PROCEDURES THAT CAN NOT BE INTERRU
                         PTED
```

```
                        ; * DUE TO NON REENTRANCY
                        ; *
                        ; *****
0CF2                    INCLOK:
0CF2    E5                      PUSH    H
0CF3    21 800C                 LXI     H,INTLOK
0CF6    34                      INR     M
0CF7    E1                      POP     H
0CF8    C9                      RET .
                        ; *****
                        ; *
                        ; * DECREMENT INTERRUPT LOCKOUT FLAG
                        ; * RELEASES LOCKOUT SET BY CALL TO INCLOK
                        ; *
                        ; *****
0CF9                    DECLOK:
0CF9    E5                      PUSH    H
0CFA    21 800C                 LXI     H,INTLOK
0CFD    35                      DCR     M
0CFE    E1                      POP     H
0CFF    C9                      RET
                        ; *****
                        ; *
                        ; * INTERRUPT SCHEDULING VERB
                        ; * SCHEDULES A VERB TO BE EXECUTED AT INTERRUPT
                         LEVEL
                        ; * interruptline routineaddress DOIT .
                        ; *
                        ; *****
                                VERB    "DOIT"[
0D00    0C86            +       .WORD   .LINK.
0D02    04              +       .BYTE   ..0001-.-1
0D03    444F4954        +       .ASCII  "DOIT"
0D07                    +..0001:]
0D07    E1                      POP     H               ; H=VERB ADDR TO
                         DO
0D08    22 8019                 SHLD    INTVRB
0D0B    E1                      POP     H               ; H=LINE TO DOIT
                         ON
0D0C    7D                      MOV     A,L
0D0D    D30F                    OUT     INLIN
0D0F    FDE5                    PUSH    Y ; ** CLUDGE TO REMEMBER IY FOR
                         INT USE **
0D11    E1                      POP     H
0D12    22 801B                 SHLD    IYVALU
                                NEXT[
0D15    FDE9            +       PCIY]
0D18                            .LOC    (.+1)&0FFFEH    ; TO BYTE BOUNDA
                        RY
0D18                    INTVEC:
0D18    0D1A                    .WORD   SCRINT
                        ; *****
                        ; *
                        ; * SCREEN INTERRUPT ROUTINE
                        ; * PERFORMS KEYBOARD SCAN, THEN TRANSFERS CONTR
```

```
                           OL TO
                           ; * USERS INTERRUPT VERB, IF ONE HAS BEEN SPECIF
                           IED
                           ; *
                           ; *****
 0D1A                      SCRINT:
 0D1A     F5                       PUSH    PSW
 0D1B     C5                       PUSH    B
 0D1C     D5                       PUSH    D
 0D1D     E5                       PUSH    H
 0D1E     D9                       EXX
 0D1F     08                       EXAF
 0D20     F5                       PUSH    PSW
 0D21     C5                       PUSH    B
 0D22     D5                       PUSH    D
 0D23     E5                       PUSH    H
 0D24     FDE5                     PUSH    Y
                           ; CHECK KEYBOARD
 0D26     CD 17B5                  CALL    KEYSCN
                           ; INTERRUPTS ON?
 0D29     3A 800C                  LDA     INTLOK
 0D2C     A7                       ANA     A
 0D2D     2018                     JRNZ    GOBACK
                           ; HAVE WE SOMETHANG TO DO?
 0D2F     2A 8019                  LHLD    INTVRB
 0D32     7C                       MOV     A,H
 0D33     B5                       ORA     L
 0D34     2811                     JRZ     GOBACK
 0D36     FD2A 801B                LIYD    IYVALU
 0D3A     01 0D45                  LXI     B,GOBAKV
 0D3D     11 0000                  LXI     D,0
 0D40     ED53 8019                SDED    INTVRB
 0D44     E9                       PCHL
 0D45                      GOBAKV:
 0D45     0D47                     .WORD   GOBACK
 0D47                      GOBACK:
 0D47     FDE1                     POP     Y
 0D49     E1                       POP     H
 0D4A     D1                       POP     D
 0D4B     C1                       POP     B
 0D4C     F1                       POP     PSW
 0D4D     08                       EXAF
 0D4E     D9                       EXX
 0D4F     E1                       POP     H
 0D50     D1                       POP     D
 0D51     C1                       POP     B
 0D52                      INTNOGO:
 0D52     F1                       POP     PSW
 0D53     FB                       EI
 0D54     C9                       RET
                           ; *****
                           ; *
                           ; *  VERB TO RE-INITIALIZE INTERRUPT REGISTERS
                           ; *  AND RE-ENABLE INTERRUPTS
                           ; *
```

```
                        ; *****
                                VERB    "ENABLE"[
 0D55   0D00        +           .WORD   .LINK.
 0D57   06          +           .BYTE   ..0002-.-1
 0D58   454E41424C45+           .ASCII  "ENABLE"
 0D5E               +..0002:]
 0D5E   21 0D18                 LXI     H,INTVEC
 0D61   7C                      MOV     A,H
 0D62   ED47                    STAI
 0D64   7D                      MOV     A,L
 0D65   D30D                    OUT     INFBK
 0D67   ED5E                    IM2
 0D69   FB                      EI
                                NEXT[
 0D6A   FDE9        +           PCIY]
                        ; *****
                        ; *
                        ; * VERB TO READ BLOCK INTO MEMORY
                        ; * memaddrress TAPEIN .
                        ; *
                        ; *****
                                VERB    "TAPEIN"[
 0D6C   0D55        +           .WORD   .LINK.
 0D6E   06          +           .BYTE   ..0003-.-1
 0D6F   54415045494E+           .ASCII  "TAPEIN"
 0D75               +..0003:]
                        ; ADDR TAPEIN
 0D75   E1                      POP     H
 0D76   CD 0D97                 CALL    TAPGET
                                NEXT[
 0D79   FDE9        +           PCIY]
                        ; *****
                        ; *
                        ; * LOAD DICTIONARY INTO MEMORY
                        ; *
                        ; *****
                                VERB    "TLOAD"[
 0D7B   0D6C        +           .WORD   .LINK.
 0D7D   05          +           .BYTE   ..0004-.-1
 0D7E   544C4F4144  +           .ASCII  "TLOAD"
 0D83               +..0004:]
 0D83   21 8080                 LXI     H,DPVAL ; HL=DICTIONARY START AD
                        DR
 0D86   CD 0D97                 CALL    TAPGET
 0D89   2B                      DCX     H       ; DE = .LAST. READ IN
 0D8A   56                      MOV     D,M
 0D8B   2B                      DCX     H
 0D8C   5E                      MOV     E,M
 0D8D   22 8060                 SHLD    DPADDR  ; STORE UPDATED DICT PTR

 0D90   ED53 8064               SDED    LASTADDR ; AND NEW LAST
 0D94   C3 139F                 JMP     CLEARE  ; CLEAR SCREEN AND GO HO
                        ME
                        ; *****
                        ; *
```

```
                         ; * READ BLOCK INTO MEMORY
                         ; * HL=READ ADDRESS
                         ; *
                         ; *****
  0D97                   TAPGET:
  0D97    C5                     PUSH    B
  0D98    F3                     DI
  0D99                   ..SENW:
  0D99    CD 0DB9                CALL    INCHAR          ; AWAIT SENTINEL
                          CHARACTER
  0D9C    79                     MOV     A,C
  0D9D    28FA                   JRZ     ..SENW
  0D9F    FEA5                   CPI     0A5H
  0DA1    20F6                   JRNZ    ..SENW
  0DA3    11 4000                LXI     D,4000H         ; DE=FEEDBACK ST
                         ORE ADDR
  0DA6                   ..CHRL:
  0DA6    D5                     PUSH    D
  0DA7    CD 0DB9                CALL    INCHAR
  0DAA    D1                     POP     D
  0DAB    2809                   JRZ     ..DONE
  0DAD    71                     MOV     M,C
  0DAE    79                     MOV     A,C
  0DAF    12                     STAX    D               ; GIVE FEEDBACK
                         ON SCREEN
  0DB0    13                     INX     D               ; BUMP FEEDBACK
                         ADDR
  0DB1    CBA2                   RES     4,D             ; CONSTRAIN TO 4
                         000H-4FFFH
  0DB3    23                     INX     H
  0DB4    18F0                   JMPR    ..CHRL
  0DB6                   ..DONE:
  0DB6    FB                     EI
  0DB7    C1                     POP     B
  0DB8    C9                     RET
                         ; *****
                         ; *
                         ; * SUBROUTINE TO INPUT A CHARACTER
                         ; * RETURNS CHARACTER IN C
                         ; * AND STATUS OF NONZERO UNLESS A TIMEOUT HAPPE
                         NED
                         ; * IN WHICH CASE ZERO STATUS IS RETURNED
                         ; *
                         ; *****
  0DB9                   INCHAR:
  0DB9    01 0810                LXI     B,810H  ; B=BIT CTR, C=TIMEOUT F
                         ACTOR
  0DBC    CD 0DCD        ..SBW:  CALL    INBIT   ; AWAIT START BIT
  0DBF    2804                   JRZ     ..GETL
  0DC1    0D                     DCR     C       ; NOT YET - DCR TIMEOUT
  0DC2    20F8                   JRNZ    ..SBW   ; IF COUNTED DOWN
  0DC4    C9                     RET             ; RETURN ZERO SET
  0DC5    CD 0DCD        ..GETL: CALL    INBIT
  0DC8    0F                     RRC             ; BIT GOT TO CY
  0DC9    CB19                   RARR    C       ; SHIFT INTO C HO
```

```
 0DCB    10F8                   DJNZ    ..GETL
                        ; NOW FALL INTO INBIT TO EAT THE STOP BIT
 0DCD                   INBIT:
 0DCD    DB99                   IN      TAPEIO  ; WAIT TILL WE GET A TRA
                        NSITION
 0DCF    5F                     MOV     E,A
 0DD0    DB99           ..INBW: IN      TAPEIO
 0DD2    AB                     XRA     E
 0DD3    0F                     RRC
 0DD4    30FA                   JRNC    ..INBW
                                WASTE   31      ; WAIT UNTIL SAMPLE POIN
                        T[
 0DD6    3E1F          +        MVI     A,31
 0DD8    3D            +..0005: DCR     A
 0DD9    20FD          +        JRNZ    ..0005]
 0DDB    DB99                   IN      TAPEIO
 0DDD    AB                     XRA     E       ; COMPARE TO OLDER STUFF

 0DDE    E601                   ANI     1
 0DE0    C8                     RZ              ; 0 IF TRANSITION HAPPEN
                        ED
                                WASTE   29      ; ELSE WAIT UNTIL MIDDLE
                         OF NEXT CYCLE[
 0DE1    3E1D          +        MVI     A,29
 0DE3    3D            +..0006: DCR     A
 0DE4    20FD          +        JRNZ    ..0006]
 0DE6    3C                     INR     A       ; RETURN VAL OF 1
 0DE7    C9                     RET
                        ; *****
                        ; *
                        ; * VERB TO WRITE OUT DICTIONARY
                        ; *
                        ; *****
                                VERB    "TSAVE"[
 0DE8    0D7B          +        .WORD   .LINK.
 0DEA    05            +        .BYTE   ..0007-.-1
 0DEB    5453415645    +        .ASCII  "TSAVE"
 0DF0                  +..0007:]
 0DF0    ED5B 8064              LDED    LASTADDR ; PUT .LAST. AT END
 0DF4    2A 8060                LHLD    DPADDR
 0DF7    73                     MOV     M,E
 0DF8    23                     INX     H
 0DF9    72                     MOV     M,D
 0DFA    23                     INX     H
 0DFB    11 8080                LXI     D,DPVAL ; COMPUTE SIZE
 0DFE    A7                     ANA     A
 0DFF    ED52                   DSBC    D
 0E01    EB                     XCHG            ; HL=ADDR,DE=SIZE
 0E02    180C                   JMPR    SAVEE
                        ; *****
                        ; *
                        ; * VERB TO WRITE OUT A BLOCK OF BYTES TO TAPE
                        ; * memaddress numbytes TAPEOUT .
                        ; *
                        ; *****
```

```
                                VERB    "TAPEOUT"[
 0E04   0DE8        +           .WORD   .LINK.
 0E06   07          +           .BYTE   ..0008-.-1
 0E07   544150454F55+           .ASCII  "TAPEOUT"
 0E0E               +..0008:]
 0E0E   D1                      POP     D
 0E0F   E1                      POP     H
 0E10                SAVEE:
 0E10   C5                      PUSH    B
 0E11   F3                      DI
 0E12   CD 0E5A                 CALL    LEADER  ; WRITE OUT LEADER
 0E15   0EA5                    MVI     C,0A5H  ; WRITE SENTINEL
 0E17   CD 0E2F                 CALL    OUTBYT
 0E1A   CD 0E24                 CALL    WRBLOC  ; AND DATA BLOCK
 0E1D   CD 0E5A                 CALL    LEADER  ; THEN TRAILER
 0E20   FB                      EI
 0E21   C1                      POP     B
                                NEXT[
 0E22   FDE9        +           PCIY]
                     ; *****
                     ; *
                     ; * SUBROUTINE TO WRITE OUT BLOCK OF BYTES
                     ; * HL=LIST, DE=# OF BYTES
                     ; *
                     ; *****
 0E24                WRBLOC:
 0E24                ..BYTL:
 0E24   4E                      MOV     C,M
 0E25   CD 0E2F                 CALL    OUTBYT
 0E28   23                      INX     H
 0E29   1B                      DCX     D
 0E2A   7A                      MOV     A,D
 0E2B   B3                      ORA     E
 0E2C   20F6                    JRNZ    ..BYTL
 0E2E   C9                      RET
                     ; *****
                     ; *
                     ; * WRITE OUT A BYTE ONTO TAPE
                     ; *
                     ; * THIS ROUTINE IS TIME SENSITIVE! CHANGE CAREF
                     ULLY!
                     ; *
                     ; *****
 0E2F                OUTBYT:
 0E2F   CD 0E7A                 CALL    WRZERO          ; WRITE START BI
                     T
                                WASTE   19              ; FINISH START B
                     IT[
 0E32   3E13        +           MVI     A,19
 0E34   3D          +..0009:    DCR     A
 0E35   20FD        +           JRNZ    ..0009]
 0E37   0608                    MVI     B,8             ; WRITE OUT 8 BI
                     TS
 0E39   CB09         ..WRL:     RRCR    C       ; SHIFT NEXT BIT TO CY
 0E3B   380A                    JRC     ..WR1   ; BRANCH ON BIT VALUE
 0E3D    CD 0E7A                 CALL    WRZERO  ; THIS GUY ZERO
                                 WASTE   19[
 0E40    3E13        +           MVI     A,19
 0E42    3D          +..0010:    DCR     A
 0E43    20FD        +           JRNZ    ..0010]
 0E45    1808                    JMPR    ..WRE
 0E47                 ..WR1:
                      ; WRITE ONE BIT
 0E47    CD 0E70                 CALL    WRONE
                                 WASTE   42[
 0E4A    3E2A        +           MVI     A,42
 0E4C    3D          +..0011:    DCR     A
 0E4D    20FD        +           JRNZ    ..0011]
 0E4F                 ..WRE:
 0E4F    10E8                    DJNZ    ..WRL   ; LOOP TILL BYTE DONE
 0E51    CD 0E70                 CALL    WRONE   ; WRITE STOP BIT
                                 WASTE   43[
 0E54    3E2B        +           MVI     A,43
 0E56    3D          +..0012:    DCR     A
 0E57    20FD        +           JRNZ    ..0012]
 0E59    C9                      RET
                      ; SUBROUTINE TO WRITE OUT 3 SECONDS WORTH OF LEA
                      DER
 0E5A                 LEADER:
 0E5A    01 0E10                 LXI     B,3600
 0E5D                 ..LDR1:
                                 WASTE   43[
 0E5D    3E2B        +           MVI     A,43
 0E5F    3D          +..0013:    DCR     A
 0E60    20FD        +           JRNZ    ..0013]
 0E62    CD 0E70                 CALL    WRONE
 0E65    0B                      DCX     B
 0E66    78                      MOV     A,B
 0E67    B1                      ORA     C
 0E68    20F3                    JRNZ    ..LDR1
                                 WASTE   42[
 0E6A    3E2A        +           MVI     A,42
 0E6C    3D          +..0014:    DCR     A
 0E6D    20FD        +           JRNZ    ..0014]
 0E6F    C9                      RET
                      ; SUBROUTINE TO WRITE 1 HALF CYCLE OF A ONE BIT
                      ; 1/1200 SEC
 0E70                 WRONE:
 0E70    D399                    OUT     TAPEIO
                                 WASTE   46[
 0E72    3E2E        +           MVI     A,46
 0E74    3D          +..0015:    DCR     A
 0E75    20FD        +           JRNZ    ..0015]
 0E77    D399                    OUT     TAPEIO
 0E79    C9                      RET
                      ; SUBROUTINE TO WRITE 1 HALF CYCLE OF A ZERO BIT

                      ; 1/2400 SEC
 0E7A                 WRZERO:
 0E7A    D399                    OUT     TAPEIO
```

```
                                        WASTE   23[
 0E7C   3E17        +                   MVI     A,23
 0E7E   3D          +..0016: DCR     A
 0E7F   20FD        +                   JRNZ    ..0016]
 0E81   D399                            OUT     TAPEIO
 0E83   C9                              RET
                     ; *****
                     ; *
                     ; * INPUT FROM PORT
                     ; * portnumber INPUT inputvalue .
                     ; *
                     ; *****
                                        VERB    "INPUT"[
 0E84   0E04        +                   .WORD   .LINK.
 0E86   05          +                   .BYTE   ..0017-.-1
 0E87   494E505554  +                   .ASCII  "INPUT"
 0E8C               +..0017:]
 0E8C   E1                              POP     H               ; HL=PORT #
 0E8D   C5                              PUSH    B
 0E8E   44                              MOV     B,H
 0E8F   4D                              MOV     C,L
 0E90   ED68                            INP     L
 0E92   2600                            MVI     H,0
 0E94   C1                              POP     B
 0E95   E5                              PUSH    H
                                        NEXT[
 0E96   FDE9        +                   PCIY]
                     ; *****
                     ; *
                     ; * OUTPUT TO PORT
                     ; * value portnumber OUTPUT .
                     ; *
                     ; *****
                                        VERB    "OUTPUT"[
 0E98   0E84        +                   .WORD   .LINK.
 0E9A   06          +                   .BYTE   ..0018-.-1
 0E9B   4F5554505554+                   .ASCII  "OUTPUT"
 0EA1               +..0018:]
 0EA1   E1                              POP     H
 0EA2   D1                              POP     D
 0EA3   C5                              PUSH    B
 0EA4   44                              MOV     B,H
 0EA5   4D                              MOV     C,L
 0EA6   ED59                            OUTP    E
 0EA8   C1                              POP     B
                                        NEXT[
 0EA9   FDE9        +                   PCIY]
                     ; *****
                     ; *
                     ; * INTERROGATE PIXEL ON SCREEN
                     ; *  x-coord y-coord PIXEL pixval .
                     ; *
                     ; *****
                                        VERB    "PIXEL"[
 0EAB   0E98        +                   .WORD   .LINK.
 0EAD     05           +          .BYTE   ..0019-.-1
 0EAE     504958454C   +          .ASCII  "PIXEL"
 0EB3                  +..0019:]
 0EB3     E1                      POP     H
 0EB4     D1                      POP     D
 0EB5     C5                      PUSH    B
 0EB6     55                      MOV     D,L
 0EB7     CD 0EC1                 CALL    GPIXEL
 0EBA     5F                      MOV     E,A
 0EBB     1600                    MVI     D,0
 0EBD     C1                      POP     B
 0EBE     D5                      PUSH    D
                                  NEXT[
 0EBF     FDE9         +          PCIY]
                        ; SUBROUTINE TO INTERROGATE PIXEL
 0EC1     CD 1476       GPIXEL:   CALL    R2ACLP   ; CONVERT TO ABSOLUTE
 0EC4     47                      MOV     B,A      ; B=SHIFT AMOUNT
 0EC5     3E04                    MVI     A,4      ; ASSUME OUTSIDE OF WIND
                        OW
 0EC7     F8                      RM               ; RETURN IF OUTSIDE
 0EC8     04                      INR     B        ; B=PIXEL SHIFT AMOUNT
 0EC9     7E                      MOV     A,M      ; A=DATA FROM SCREEN
 0ECA     07            ..SHFT:   RLC              ; ROTATE UNTIL
 0ECB     07                      RLC              ; WE GOT ADDRESSED PIXEL

 0ECC     10FC                    DJNZ    ..SHFT
 0ECE     E603                    ANI     3        ; IN BITS 0 AND 1
 0ED0     C9                      RET
 0ED1                   ..POX:
                        ; *****
                        ; *
                        ; * MODIFY POINT
                        ; *  x-coord y-coord mode POINT .
                        ; *
                        ; * VALUES FOR mode:
                        ; * 0-3           XOR WITH PIXEL VALUE 0,1,2, OR 3

                        ; * 4-7           PLOP WITH PIXEL VALUE N-4
                        ; * 8-11          OR WITH PIXEL VALUE N-8
                        ; * 12-15         PRIORITY WRITE WITH N-12
                        ; * ANY VALUE OUTSIDE THIS RANGE IS TAKEN MOD 16
                        .
                        ; *
                        ; *****
                                  VERB    "POINT"[
 0ED1     0EAB         +          .WORD   .LINK.
 0ED3     05           +          .BYTE   ..0020-.-1
 0ED4     504F494E54   +          .ASCII  "POINT"
 0ED9                  +..0020:]
 0ED9     E1                      POP     H
 0EDA     7D                      MOV     A,L
 0EDB     E1                      POP     H
 0EDC     D1                      POP     D
 0EDD     55                      MOV     D,L
 0EDE     C5                      PUSH    B
```

```
  0EDF    CD 0CF2                   CALL    INCLOK
  0EE2    CD 0EEB                   CALL    POINTR
  0EE5    CD 0CF9                   CALL    DECLOK
  0EE8    C1                        POP     B
                                    NEXT[
  0EE9    FDE9          +           PC[Y]
                          ; *****
                          ; *
                          ; * POINT SUBROUTINE
                          ; * A=MODE PARAMETER
                          ; * DE= Y,X COORDINATES
                          ; * HL=SMASHED
                          ; *
                          ; *****
  0EEB    CB57            POINTR:   BIT     2,A      ; OR OR XOR?
  0EED    201F                      JRNZ    PRPLOP   ; NO
                          ; OR OR XOR MODE - SET EXPANDER WITH PIXEL VALUE

  0EEF    07                        RLC
  0EF0    07                        RLC
  0EF1    E6FC                      ANI     0FCH
  0EF3    D319                      OUT     XPAND
  0EF5    CB6F                      BIT     5,A
  0EF7    2008                      JRNZ    ORPT
  0EF9    CD 1476         XORPT:    CALL    R2ACLP
  0EFC    F8                        RM
  0EFD    F628                      ORI     0101000B
  0EFF    1806                      JMPR    ORJOIN
  0F01    CD 1476         ORPT:     CALL    R2ACLP
  0F04    F8                        RM
  0F05    F618                      ORI     0011000B
  0F07    D30C            ORJOIN:   OUT     MAGIC
  0F09    CBB4                      RES     6,H
  0F0B    3680                      MVI     M,80H
  0F0D    C9                        RET
  0F0E    CB5F            PRPLOP:   BIT     3,A      ; IS IT PLOP?
  0F10    202A                      JRNZ    PRIOR
                          ; PLOP ROUTINE
  0F12    C5              PLOP:     PUSH    B
  0F13    D319            PLOP1:    OUT     XPAND
  0F15    3E08                      MVI     A,00001000B
  0F17    D30C                      OUT     MAGIC
  0F19    32 0FFF                   STA     URINAL
  0F1C    3A 4FFF                   LDA     CRAPPER
  0F1F    47                        MOV     B,A
  0F20    CD 1476                   CALL    R2ACLP
  0F23    FA 0F3A                   JM      PLOPNG
  0F26    F608                      ORI     00001000B
  0F28    D30C                      OUT     MAGIC
  0F2A    3E8C                      MVI     A,10001100B
  0F2C    D319                      OUT     XPAND
  0F2E    32 0FFF                   STA     URINAL
  0F31    3A 4FFF                   LDA     CRAPPER
  0F34    4F                        MOV     C,A
  0F35    78                        MOV     A,B
```

```
 0F36    AE                       XRA     M
 0F37    A1                       ANA     C
 0F38    AE                       XRA     M
 0F39    77                       MOV     M,A
 0F3A    C1             PLOPNG:   POP     B
 0F3B    C9                       RET
                        ; PRIORITY
 0F3C    C5             PRIOR:    PUSH    B
 0F3D    E603                     ANI     3
 0F3F    4F                       MOV     C,A
 0F40    CD 0EC1                  CALL    GPIXEL
 0F43    B9                       CMP     C         ; COMPARE TO WHATS THERE

 0F44    30F4                     JRNC    PLOPNG
 0F46    79                       MOV     A,C
 0F47    18CA                     JMPR    PLOP1
                        ; *****
                        ; *
                        ; * TERSE CIRCLE COMMAND
                        ; *  x-coord y-coord radius mode CIRCLE .
                        ; *
                        ; * THIS CIRCLE ROUTINE WAS ADAPTED FROM THE
                        ; * BRESENHAM CIRCLE ALGORITHM PUBLISHED IN
                        ; * CACM FEB 1977
                        ; *
                        ; *****
                                  VERB    "CIRCLE"[
 0F49    0ED1           +         .WORD   .LINK.
 0F4B    06             +         .BYTE   ..0021-.-1
 0F4C    434952434C45+            .ASCII  "CIRCLE"
 0F52                   +..0021:]
                                  FRAME[
 0F52    FDE5           +         PUSH    Y
 0F54    FD21 0000      +         LXI     Y,0
 0F58    FD39           +         DADY    SP]
 0F5A    C5                       PUSH    B
 0F5B    21 0001                  LXI     H,1               ; XA=1
 0F5E    CD 0CF2                  CALL    INCLOK
 0F61    22 8013                  SHLD    CIRXA
 0F64    2B                       DCX     H                 ; HL=0
 0F65    FD5E04                   MOV     E,FR.P2(Y)        ; DE=R
 0F68    FD5605                   MOV     D,FR.P2+1(Y)
 0F6B    D5                       PUSH    D
 0F6C    1B                       DCX     D
 0F6D    CD 1027                  CALL    CIRPNT            ; DRAW AT O,R
 0F70    EB                       XCHG
 0F71    CD 1058                  CALL    NEGHL
 0F74    EB                       XCHG
 0F75    CD 1027                  CALL    CIRPNT            ; AND AT O,-R
 0F78    E1                       POP     H                 ; R IS IN HL
 0F79    29                       DAD     H
 0F7A    2B                       DCX     H
 0F7B    22 8015                  SHLD    CIRYA
 0F7E    2B                       DCX     H                 ; DELTA=YA-1
 0F7F    22 8017                  SHLD    CIRDEL
```

```
  0F82                  CLOOP:
  0F82    2A 8017               LHLD    CIRDEL          ; D=DELTA*2
  0F85    29                    DAD     H
  0F86    CB7C                  BIT     7,H             ; IS D<0?
  0F88    201C                  JRNZ    DMINUS          ; YEP
  0F8A    ED5B 8015             LDED    CIRYA           ; NO, COMPUTE D=
                        D-YA
  0F8E    A7                    ANA     A
  0F8F    ED52                  DSBC    D
  0F91    CB7C                  BIT     7,H             ; NOW IS D<0?
  0F93    2029                  JRNZ    CBOTH           ; YEP - DO BOTH
                        MOVES
  0F95    2A 8013               LHLD    CIRXA           ; NO XA=XA+2
  0F98    23                    INX     H
  0F99    23                    INX     H
  0F9A    22 8013               SHLD    CIRXA
  0F9D    EB                    XCHG
  0F9E    2A 8017               LHLD    CIRDEL          ; DELTA=DELTA-XA

  0FA1    A7                    ANA     A
  0FA2    ED52                  DSBC    D
  0FA4    1831                  JMPR    DSTOR
                        ; D < 0 CASE
  0FA6                  DMINUS:
  0FA6    ED5B 8013             LDED    CIRXA           ; D=D+XA
  0FAA    19                    DAD     D
  0FAB    CB7C                  BIT     7,H             ; IF D<0
  0FAD    280F                  JRZ     CBOTH           ; BUMP BOTH
  0FAF    2A 8015               LHLD    CIRYA           ; ELSE YA=YA-2
  0FB2    2B                    DCX     H
  0FB3    2B                    DCX     H
  0FB4    22 8015               SHLD    CIRYA
  0FB7    EB                    XCHG
  0FB8    2A 8017               LHLD    CIRDEL          ; DELTA=DELTA+YA

  0FBB    19                    DAD     D
  0FBC    1819                  JMPR    DSTOR
                        ; INCREMENT BOTH
  0FBE    2A 8013       CBOTH:  LHLD    CIRXA           ; XA=XA+2
  0FC1    23                    INX     H
  0FC2    23                    INX     H
  0FC3    22 8013               SHLD    CIRXA
  0FC6    EB                    XCHG
  0FC7    2A 8015               LHLD    CIRYA           ; YA=YA-2
  0FCA    2B                    DCX     H
  0FCB    2B                    DCX     H
  0FCC    22 8015               SHLD    CIRYA
  0FCF    ED4B 8017             LBCD    CIRDEL          ; DELTA=DELTA+YA
                        -XA
  0FD3    09                    DAD     B
  0FD4    A7                    ANA     A
  0FD5    ED52                  DSBC    D
  0FD7    22 8017       DSTOR:  SHLD    CIRDEL
  0FDA    2A 8015               LHLD    CIRYA
  0FDD    2B                    DCX     H         ; IF NEG, 0 OR 1
  0FDE    2B                       DCX     H
  0FDF    CB7C                     BIT     7,H
  0FE1    281E                     JRZ     ..CYOK
                          ; CLOSE TO DONE - DRAW LAST TWO POINTS
  0FE3    11 0000                  LXI     D,0
  0FE6    FD6E04                   MOV     L,FR.P2(Y)       ; HL=R
  0FE9    FD6605                   MOV     H,FR.P2+1(Y)
  0FEC    CD 1027                  CALL    CIRPNT  ; AT R,0
  0FEF    CD 1058                  CALL    NEGHL
  0FF2    CD 1027                  CALL    CIRPNT  ; AND -R,0
                          ; WE ARE DONE
  0FF5    CD 0CF9                  CALL    DECLOK
  0FF8    C1                       POP     B                ; YES - GO HOME
                                   UNFRAME[
  0FF9    FDE1            +        POP     Y]
  0FFB    E1                       POP     H
  0FFC    E1                       POP     H
  0FFD    E1                       POP     H
  0FFE    E1                       POP     H
                                   NEXT[
  0FFF    FDE9            +        PCIY]
  1001                    ..CYOK:
                          ; SUBROUTINE TO DRAW 4 POINTS
                          ; AT X,Y -X,Y -X,-Y AND X,-Y
  1001    2A 8015         DRAW4:   LHLD    CIRYA
  1004    CD 1050                  CALL    DIV2HL
  1007    E5                       PUSH    H
  1008    EB                       XCHG
  1009    2A 8013                  LHLD    CIRXA
  100C    CD 1050                  CALL    DIV2HL
  100F    CD 1027                  CALL    CIRPNT  ; X,Y
  1012    EB                       XCHG
  1013    CD 1058                  CALL    NEGHL
  1016    EB                       XCHG
  1017    CD 1027                  CALL    CIRPNT
  101A    CD 1058                  CALL    NEGHL            ; -X
  101D    CD 1027                  CALL    CIRPNT           ; -X,-Y
  1020    D1                       POP     D                ; -X,Y
  1021    CD 1027                  CALL    CIRPNT
  1024    C3 0F82                  JMP     CLOOP
  1027    E5              CIRPNT:  PUSH    H
  1028    D5                       PUSH    D
  1029    FD4E08                   MOV     C,FR.P4(Y)       ; GET X TRANS FA
                          CTOR
  102C    FD4609                   MOV     B,FR.P4+1(Y)
  102F    09                       DAD     B
                          ; GROSS CLIP CHECK
  1030    7D                       MOV     A,L              ; SIGN OF L TO C
                          Y
  1031    07                       RLC
  1032    7C                       MOV     A,H
  1033    CE00                     ACI     0
  1035    2016                     JRNZ    ..CIRX
  1037    EB                       XCHG
  1038    FD4E06                   MOV     C,FR.P3(Y)       ; Y TRIP
  103B    FD4607                  MOV     B,FR.P3+1(Y)
  103E    09                      DAD     B
  103F    7D                      MOV     A,L
  1040    07                      RLC
  1041    7C                      MOV     A,H
  1042    CE00                    ACI     0
  1044    2007                    JRNZ    ..CIRX
  1046    55                      MOV     D,L
  1047    FD7E02                  MOV     A,FR.P1(Y)
  104A    CD 0EEB                 CALL    POINTR
  104D    D1              ..CIRX: POP     D
  104E    E1                      POP     H
  104F    C9                      RET
                          ; DIVIDE HL BY 2
  1050                    DIV2HL:
  1050    A7                      ANA     A
  1051    7C                      MOV     A,H
  1052    1F                      RAR
  1053    67                      MOV     H,A
  1054    7D                      MOV     A,L
  1055    1F                      RAR
  1056    6F                      MOV     L,A
  1057    C9                      RET
                          ; NEGATE CONTENTS OF HL
  1058                    NEGHL:
  1058    7C                      MOV     A,H
  1059    2F                      CMA
  105A    67                      MOV     H,A
  105B    7D                      MOV     A,L
  105C    2F                      CMA
  105D    6F                      MOV     L,A
  105E    23                      INX     H
  105F    C9                      RET
                          ; *****
                          ; *
                          ; * TERSE SCROLL COMMAND
                          ; *  x-coord y-coord x-size y-size scrollamount
                          SCROLL .
                          ; *
                          ; *****
                                  VERB    "SCROLL"[
  1060    0F49          +         .WORD   .LINK.
  1062    06            +         .BYTE   ..0022-.-1
  1063    5343524F4C4C+           .ASCII  "SCROLL"
  1069                  +..0022:]
  1069                    SCROLE:
                                  FRAME[
  1069    FDE5          +         PUSH    Y
  106B    FD21 0000     +         LXI     Y,0
  106F    FD39          +         DADY    SP]
  1071    C5                      PUSH    B
  1072    CD 12C5                 CALL    CLIP    ; CLIP BOX
  1075    384E                    JRC     ..NOSC
                          ; CONVERT X SIZE TO BYTES
  1077    FD7E06                  MOV     A,BX.XS(Y)
```

```
 107A    C603                   ADI     3
 107C    0F                     RRC
 107D    0F                     RRC
 107E    E63F                   ANI     3FH
 1080    4F                     MOV     C,A
                        ; WHICH DIRECTION?
 1081    FD7E02                 MOV     A,BX.MOD(Y)
 1084    A7                     ANA     A
 1085    283E                   JRZ     ..NOSC
 1087    11 FFD8                LXI     D,-40   ; ASSUME MINUS
 108A    FD4604                 MOV     B,BX.YS(Y)
 108D    FD7E08                 MOV     A,BX.Y(Y)
 1090    FA 1098                JM      ..MINU
                        ; POSITIVE CASE
 1093    11 0028                LXI     D,40
 1096    80                     ADD     B
 1097    3D                     DCR     A
 1098    05             ..MINU: DCR     B       ; FUDGE Y SIZE
 1099    282A                   JRZ     ..NOSC  ; SKIP IF WAS ONLY 1
 109B    D5                     PUSH    D
 109C    57                     MOV     D,A
 109D    FD5E0A                 MOV     E,BX.X(Y)
 10A0    CD 1491                CALL    R2A
 10A3    D1                     POP     D
 10A4    FD7E02                 MOV     A,BX.MOD(Y)     ; A=SCROLL AMOUN
                        T
 10A7    CD 1471                CALL    ABS
                        ; A=SCROLLAMOUNT, DE=LINE DISPLACEMENT
                        ; C=HORIZONTAL BYTES, B=VERTICAL LINES
                        ; REPEAT SCROLLAMOUNT TIMES ...
 10AA    C5             ..SCR1: PUSH    B
 10AB    E5                     PUSH    H
                        ; REPEAT Y SIZE TIMES ...
 10AC    C5             ..SCR2: PUSH    B
 10AD    D5                     PUSH    D
 10AE    0600                   MVI     B,0
 10B0    EB                     XCHG
 10B1    19                     DAD     D
 10B2    E5                     PUSH    H
 10B3    EDB0                   LDIR            ; TRANSFER ONE HORIZONTA
                        L LINE
 10B5    E1                     POP     H
 10B6    D1                     POP     D
 10B7    C1                     POP     B
 10B8    10F2                   DJNZ    ..SCR2
                        ; CLEAR AWAY LAST LINE SO WE DON'T GET STREAKIES

 10BA    3600           ..KILL: MVI     M,0
 10BC    23                     INX     H
 10BD    0D                     DCR     C
 10BE    20FA                   JRNZ    ..KILL
 10C0    E1                     POP     H
 10C1    C1                     POP     B
 10C2    3D                     DCR     A
 10C3    20E5                   JRNZ    ..SCR1
```

```
  10C5                  ..NOSC:
  10C5    C1                    POP     B
                                UNFRAME[
  10C6    FDE1         +        POP     Y]
  10C8    E1                    POP     H
  10C9    E1                    POP     H
  10CA    E1                    POP     H
  10CB    E1                    POP     H
  10CC    E1                    POP     H
                                NEXT[
  10CD    FDE9         +        PCIY]
                        ; *****
                        ; *
                        ; * SNAP
                        ; * x-coord y-coord x-size y-size arrayaddress S
                        NAP .
                        ; *
                        ; * THIS VERB 'TAKES A PICTURE' OF A RECTANGULAR
                         AREA
                        ; * OF THE SCREEN.  THIS IMAGE IS STORED OFFSCRE
                        EN IN
                        ; * AN ARRAY.  THE FIRST TWO WORDS OF THIS ARRAY
                         CONTAIN
                        ; * THE X AND Y SIZE OF THE SNAPPED AREA.
                        ; *
                        ; *****
                                VERB    "SNAP"[
  10CF    1060         +        .WORD   .LINK.
  10D1    04           +        .BYTE   ..0023-.-1
  10D2    534E4150     +        .ASCII  "SNAP"
  10D6                 +..0023:]
                                FRAME[
  10D6    FDE5         +        PUSH    Y
  10D8    FD21 0000    +        LXI     Y,0
  10DC    FD39         +        DADY    SP]
  10DE    C5                    PUSH    B
  10DF    CD 12C5               CALL    CLIP
  10E2    387C                  JRC     ..NOSN
  10E4    CD 0CF2               CALL    INCLOK
  10E7    DDE5                  PUSH    X
  10E9    FD7E06                MOV     A,BX.XS(Y)       ; TRANSFER SIZE
  10EC    FD6E02                MOV     L,BX.MOD(Y)      ; HL=POINTER
  10EF    FD6603                MOV     H,BX.MOD+1(Y)
  10F2    77                    MOV     M,A              ; STUFF X SIZE O
                        F PAT
  10F3    23                    INX     H
  10F4    3600                  MVI     M,0
  10F6    23                    INX     H
  10F7    C603                  ADI     3       ; X SIZE TO BYTES
  10F9    0F                    RRC
  10FA    0F                    RRC
  10FB    E63F                  ANI     3FH
  10FD    5F                    MOV     E,A     ; REMEMBER FOR TRANSFER
                        CODE
  10FE    FD5604                MOV     D,BX.YS(Y)       ; MOVE OVER Y SI
```

```
                         ZE TOO
  1101    72                     MOV     M,D
  1102    23                     INX     H
  1103    3600                   MVI     M,0
  1105    23                     INX     H
  1106    E5                     PUSH    H
  1107    DDE1                   POP     X         ; IX=DEST ARRAY PTR
  1109    D5                     PUSH    D
  110A    FD5E0A                 MOV     E,BX.X(Y)         ; E=X
  110D    FD7E08                 MOV     A,BX.Y(Y)         ; FUDGE Y TO UPP
                         ER
  1110    82                     ADD     D
  1111    3D                     DCR     A
  1112    57                     MOV     D,A       ; TO D
  1113    CD 1491                CALL    R2A
  1116    E5                     PUSH    H                 ; IY=SOURCE POIN
                         TER
  1117    FDE1                   POP     Y
  1119    4F                     MOV     C,A               ; C=SHIFT AMOUNT

  111A    D1                     POP     D                 ; DE=SIZES
  111B    2827                   JRZ     ..EASY            ; JUMP ON EASY C
                         ASE
                         ; CASE OF FIRST PIXEL TO SNAP NOT BEING ON BYTE
                         BOUNDARY
  111D    D5             ..HARD: PUSH    D
  111E    FDE5                   PUSH    Y
  1120    FD6600                 MOV     H,0(Y)            ; H=FIRST BYTE T
                         O START
  1123    FD23           ..HDBL: INX     Y                 ; BUMP SOURCE
  1125    FD5600                 MOV     D,0(Y)            ; D=NEXT GUY
  1128    6A                     MOV     L,D               ; INTO L AS WELL

  1129    41                     MOV     B,C               ; REINIT SHIFT A
                         MT
  112A    29             ..HDSL: DAD     H                 ; SHIFT OVER SA
                         PIXELS
  112B    29                     DAD     H
  112C    10FC                   DJNZ    ..HDSL
  112E    DD7400                 MOV     0(X),H            ; STUFF TO DEST
  1131    62                     MOV     H,D               ; SETUP FOR NEXT
                          ITERATION
  1132    DD23                   INX     X
  1134    1D                     DCR     E
  1135    20EC                   JRNZ    ..HDBL
  1137    FDE1                   POP     Y                 ; TO NEXT LINE
  1139    11 0028                LXI     D,40
  113C    FD19                   DADY    D
  113E    D1                     POP     D
  113F    15                     DCR     D
  1140    20DB                   JRNZ    ..HARD
  1142    181A                   JMPR    ..SNPD
                         ; FASTER LOOP FOR ZERO SHIFT AMOUNT CASE
  1144    01 0028        ..EASY: LXI     B,40
  1147    7B             ..EZLL: MOV     A,E       ; A=TEMP BYTE COUNTER
```

```
1148    FDE5                        PUSH     Y
114A    FD6E00          ..EZBL: MOV      L,0(Y)
114D    DD7500                  MOV      0(X),L
1150    DD23                    INX      X
1152    FD23                    INX      Y
1154    3D                      DCR      A
1155    20F3                    JRNZ     ..EZBL
1157    FDE1                    POP      Y
1159    FD09                    DADY     B
115B    15                      DCR      D          ; D=OUTER LOOP CTR
115C    20E9                    JRNZ     ..EZLL
                        ;
115E                    ..SNPD:
115E    DDE1                    POP      X
1160    C1              ..NOSN: POP      B
1161    CD 0CF9                 CALL     DECLOK
                                UNFRAME[
1164    FDE1            +       POP      Y]
1166    E1                      POP      H
1167    E1                      POP      H
1168    E1                      POP      H
1169    E1                      POP      H
116A    E1                      POP      H
                                NEXT[
116B    FDE9            +       PCIY]
                        ; *****
                        ; *
                        ; * TERSE SHOW COMMAND
                        ; * x-coord y-coord magic-reg-val arrayaddress S
                        HOW .
                        ; *
                        ; * THIS COMMAND DISPLAYS A PREVIOUSLY SNAPPED O
                        BJECT
                        ; * ON THE SCREEN AT THE SPECIFIED PLACE.  THE P
                        ARAMETER
                        ; * magic-reg-val HAS THE FOLLOWING FORMAT:
                        ; *
                        ; * :-------:-------:-------:-------:-------:-------:-
                        ------:-------:
                        ; * :7      :6      :5      :4      :3      :2      :1
                              :0      :
                        ; * : XPAND ON COL:XPAND OFF COL: FLOP :   XOR :
                         OR   : XPAND:
                        ; * :               :               :       :       :
                              :       :
                        ; * :-------:-------:-------:-------:-------:-------:-
                        ------:-------:
                        ; *
                        ; *****
                                VERB     "SHOW"[
116D    10CF            +       .WORD    .LINK.
116F    04              +       .BYTE    ..0024-.-1
1170    53484F57        +       .ASCII   "SHOW"
1174                    +..0024:]
1174    E1                      POP      H
```

```
 1175    D1                      POP     D
 1176    7B                      MOV     A,E       ; SAVE MODE
 1177    1600                    MVI     D,0
                        ; MAKE A FAKE 'BOX' STACK FRAME FOR CLIPPING PUR
                        POSES
 1179    5E                      MOV     E,M       ; PUT SIZES ON STACK
 117A    D5                      PUSH    D
 117B    23                      INX     H
 117C    23                      INX     H
 117D    5E                      MOV     E,M
 117E    D5                      PUSH    D
 117F    C5                      PUSH    B         ; SAVE B TOO
                                 FRAME[
 1180    FDE5           +        PUSH    Y
 1182    FD21 0000      +        LXI     Y,0
 1186    FD39           +        DADY    SP]
 1188    E5                      PUSH    H
 1189    F5                      PUSH    PSW
 118A    E5                      PUSH    H
 118B    CD 12C5                 CALL    CLIP
 118E    E1                      POP     H
 118F    3866                    JRC     ..NOSH
 1191    FD7E04                  MOV     A,BX.YS(Y)
                        ; CHECK FOR SIZE SHRINKO - IF SO DO NOT DRAW
 1194    BE                      CMP     M         ; DID Y SIZE SHRINKO?
 1195    2060                    JRNZ    ..NOSH    ; YES
 1197    47                      MOV     B,A       ; NO
 1198    2B                      DCX     H         ; TO X SIZE
 1199    2B                      DCX     H
 119A    FD7E06                  MOV     A,BX.XS(Y) ; LOOKAT X SIZE
 119D    BE                      CMP     M
 119E    2057                    JRNZ    ..NOSH
 11A0    C603                    ADI     3         ; COMPUTE X SIZE IN BYTE
                        S
 11A2    0F                      RRC
 11A3    0F                      RRC
 11A4    E63F                    ANI     3FH
 11A6    4F                      MOV     C,A
                        ; GET AND FIX COORDINATES
 11A7    FD7E08                  MOV     A,BX.Y(Y)
 11AA    80                      ADD     B         ; FUDGE TO TOP OF BOX
 11AB    57                      MOV     D,A
 11AC    FD5E0A                  MOV     E,BX.X(Y)
 11AF    CD 1491                 CALL    R2A
 11B2    57                      MOV     D,A
 11B3    F1                      POP     PSW       ; RESTORE MODE
 11B4    07                      RLC               ; LINEUP XPAND STUFF
 11B5    07                      RLC
 11B6    07                      RLC
 11B7    07                      RLC
 11B8    CD 0CF2                 CALL    INCLOK
 11BB    D319                    OUT     XPAND     ; SET EXPAND COLORS
 11BD    0F                      RRC
 11BE    E678                    ANI     78H
 11C0    B2                      ORA     D
```

```
  11C1    D30C                  OUT     MAGIC
  11C3    CBB4                  RES     6,H       ; MAKE ADDRESS MAGIC
  11C5    D1                    POP     D
  11C6    13                    INX     D         ; MOVE PAST Y SIZE
  11C7    13                    INX     D
  11C8    EB                    XCHG              ; HL=SOURCE, DE=DEST
  11C9    CB5F                  BIT     MRXPND,A          ; EXPAND WANTED?

  11CB    2012                  JRNZ    ..MWX
                        ; NORMAL? WRITE
  11CD    AF                    XRA     A
  11CE                  ..NWRT:
  11CE    C5                    PUSH    B
  11CF    D5                    PUSH    D
  11D0    47                    MOV     B,A
  11D1    EDB0                  LDIR
  11D3    12                    STAX    D
  11D4    D1                    POP     D
  11D5    EB                    XCHG
  11D6    0E28                  MVI     C,BYTEPL
  11D8    09                    DAD     B
  11D9    EB                    XCHG
  11DA    C1                    POP     B
  11DB    10F1                  DJNZ    ..NWRT
  11DD    181A                  JMPR    ..OK
                        ; WRITE EXPANDED
  11DF    EB            ..MWX:  XCHG
  11E0    C5            ..MWX1: PUSH    B
  11E1    E5                    PUSH    H
  11E2    41                    MOV     B,C
  11E3    1A            ..MWX2: LDAX    D
  11E4    13                    INX     D
  11E5    77                    MOV     M,A
  11E6    23                    INX     H
  11E7    77                    MOV     M,A
  11E8    23                    INX     H
  11E9    10F8                  DJNZ    ..MWX2
  11EB    70                    MOV     M,B
  11EC    23                    INX     H
  11ED    70                    MOV     M,B
  11EE    E1                    POP     H
  11EF    0E28                  MVI     C,BYTEPL
  11F1    09                    DAD     B
  11F2    C1                    POP     B
  11F3    10EB                  DJNZ    ..MWX1
  11F5    1802                  JMPR    ..OK
  11F7    F1            ..NOSH: POP     PSW
  11F8    F1                    POP     PSW
  11F9                  ..OK:   UNFRAME[
  11F9    FDE1        +         POP     Y]
  11FB    CD 0CF9               CALL    DECLOK
  11FE    C1                    POP     B
  11FF    E1                    POP     H
  1200    E1                    POP     H
  1201    E1                    POP     H
```

```
1202    E1                      POP     H
                                NEXT[
1203    FDE9            +       PCIY]
                        ; *****
                        ; *
                        ; * TERSE BOX COMMAND
                        ; * x-coord y-coord x-size y-size mode BOX .
                        ; *
                        ; * THIS ROUTINE PAINTS VERTICAL STRIPES
                        ; * FIRST IT PAINTS A RIGHT JUSTIFIED STRIPE
                        ; * SO WE MOVE OVER TO A BYTE BOUNDARY
                        ; * THEN WE PAINT AS MANY SOLID BYTES AS WE
                        ; * CAN.  FINALLY WE PAINT A LEFT JUSTIFIED
                        ; * STRIPE TO FINISH OFF THE BOX.
                        ; *
                        ; *****
                                VERB    "BOX"[
1205    116D            +       .WORD   .LINK.
1207    03              +       .BYTE   ..0025-.-1
1208    424F58          +       .ASCII  "BOX"
120B                    +..0025:]
120B                    DOBOX:
                                FRAME[
120B    FDE5            +       PUSH    Y
120D    FD21 0000       +       LXI     Y,0
1211    FD39            +       DADY    SP]
1213    C5                      PUSH    B
1214    CD 0CF2                 CALL    INCLOK
1217    CD 12C5                 CALL    CLIP
121A    386F            ..SKPL: JRC     ..SKIP          ; ABORT IF TOTAL
                         OFFSCREEN
                        ; WE NOW HAVE REASONABLE STUFF ON OUR STACK
                        ; LETS DEAL WITH MODE STUFF NOW
121C    FD4E02                  MOV     C,BX.MOD(Y)
121F    79                      MOV     A,C             ; IS MODE ZERO?
1220    A7                      ANA     A
1221    2868                    JRZ     ..SKIP          ; YEP - IGNORE
1223    E604                    ANI     4               ; ISOLATE WRITE
                        MODE
1225    32 8012                 STA     WRMODE
1228    79                      MOV     A,C             ; ISOLATE PIXEL
                        NUMBER
1229    E603                    ANI     3
122B    4F                      MOV     C,A
122C    0600                    MVI     B,0             ; LOOKUP BYTE OF
                         THOSE GUYS
122E    21 13F2                 LXI     H,MSKTBL
1231    09                      DAD     B
1232    7E                      MOV     A,M
1233    32 8011                 STA     PIXVAL
                        ; NOW THE EXCITING BOX PAINTING STARTS
1236    FD5E06                  MOV     E,BX.XS(Y)
1239    7B              ..BOXP: MOV     A,E
123A    A7                      ANA     A
123B    284E                    JRZ     ..SKIP  ; SKIP IF ALL DONE OR NO
```

```
                        NE TO DO
                        ; IS MOD(X,4)=1?
                        ; IF SO WE IS ON A BYTE BOUNDARY
 123D    FD7E0A                 MOV     A,BX.X(Y)
 1240    E603                   ANI     3
 1242    FE01                   CPI     1
 1244    200E                   JRNZ    ..MNZ
                        ; YES - IS XS>4?
 1246    7B                     MOV     A,E
 1247    FE04                   CPI     4
 1249    3834                   JRC     ..XSL4
                        ; YS IS >4 - SO PAINT A FULL STRIPE
 124B    0EFF                   MVI     C,0FFH          ; DO WHOLE KIT A
                        ND KABOODLE
 124D    CD 1298                CALL    ..STRC
 1250    1604                   MVI     D,4             ; A=X ADDR SUBTR
                        ACTOR
 1252    181F                   JMPR    ..XSTF
                        ; COME UP WITH A BITSTRING TO PUT US ON A BYTE B
                        OUNDARY
                        ; IF SIZE IS'NT LARGE ENOUGH COME UP WITH THE ON
                        E AND ONLY
                        ; MASK WE WILL NEED.
 1254    3D             ..MNZ:  DCR     A
 1255    E603                   ANI     3
 1257    4F                     MOV     C,A
 1258    3E04                   MVI     A,4
 125A    91                     SUB     C
 125B    BB                     CMP     E               ; COMPARE TO XS
 125C    3801                   JRC     ..XSBG
 125E    7B                     MOV     A,E             ; MOD IS BIGGER
 125F    47             ..XSBG: MOV     B,A             ; B=MIN
 1260    57                     MOV     D,A
 1261    AF                     XRA     A               ; FORM BIT MASK
 1262    0F             ..BITF: RRC
 1263    0F                     RRC
 1264    F6C0                   ORI     11000000B
 1266    10FA                   DJNZ    ..BITF
 1268    41                     MOV     B,C
                        ; APPLY SHIFT AMOUNT
 1269    0F             ..DOSF: RRC
 126A    0F                     RRC
 126B    E63F                   ANI     3FH
 126D    10FA                   DJNZ    ..DOSF
 126F    4F                     MOV     C,A             ; REMEMBERIZE
 1270    CD 1298                CALL    ..STRC
 1273    7A             ..XSTF: MOV     A,D
 1274    FD860A                 ADD     BX.X(Y)         ; UPDATE X COORD
                        INATE
 1277    FD770A                 MOV     BX.X(Y),A
 127A    7B                     MOV     A,E             ; AND PIXELS LEF
                        T (XS)
 127B    92                     SUB     D
 127C    5F                     MOV     E,A
 127D    18BA                   JMPR    ..BOXP          ; LOOP BACK FOR
```

```
                         MORE
                         ; PAINT A FINAL STRIPE
                         ; THIS MASK IS ALWAYS LEFT JUSTIFIED
  127F   47              ..XSL4: MOV    B,A
  1280   AF                      XRA    A
  1281   0F              ..XSLA: RRC
  1282   0F                      RRC
  1283   F6C0                    ORI    11000000B
  1285   10FA                    DJNZ   ..XSLA
  1287   4F                      MOV    C,A
  1288   CD 1298                 CALL   ..STRC
  128B                   ..SKIP:
  128B   CD 0CF9                 CALL   DECLOK
  128E   C1                      POP    B
                                 UNFRAME[
  128F   FDE1           +        POP    Y]
  1291   E1                      POP    H
  1292   E1                      POP    H
  1293   E1                      POP    H
  1294   E1                      POP    H
  1295   E1                      POP    H
                                 NEXT[
  1296   FDE9           +        PCIY]
                         ; LOOP TO PAINT A VERTICAL BOX STRIPE
                         ; MASK TO USE PASSED IN C
  1298   D5              ..STRC: PUSH   D
  1299   FD5608                  MOV    D,BX.Y(Y)
  129C   FD5E0A                  MOV    E,BX.X(Y)
  129F   CD 1491                 CALL   R2A             ; CONVERT COORDI
                         NATES
  12A2   11 FFD8                 LXI    D,-40           ; NEGATIVE SCREE
                         N INCREMENT
  12A5   FD4604                  MOV    B,BX.YS(Y)      ; B=Y SIZE
  12A8   3A 8012                 LDA    WRMODE          ; WHICH TIGHT LO
                         OP TO USE?
  12AB   A7                      ANA    A
  12AC   200B                    JRNZ   ..PLOP
                         ; WRITE MAGIC (XOR FOR NOW)
  12AE   3A 8011         ..XORL: LDA    PIXVAL
  12B1   A1                      ANA    C
  12B2   AE                      XRA    M
  12B3   77                      MOV    M,A
  12B4   19                      DAD    D
  12B5   10F7                    DJNZ   ..XORL
  12B7   D1                      POP    D
  12B8   C9                      RET
                         ; PLOP WRITE LOOP
  12B9   3A 8011         ..PLOP: LDA    PIXVAL
  12BC   AE                      XRA    M
  12BD   A1                      ANA    C
  12BE   AE                      XRA    M
  12BF   77                      MOV    M,A
  12C0   19                      DAD    D
  12C1   10F6                    DJNZ   ..PLOP
  12C3   D1                      POP    D
  12C4    C9                      RET
                        ; CLIP BOTH COORDINATES ROUTINE
                        ; THIS ROUTINE EXPECTS PARAMETERS TO BE ON THE S
                        TACK FRAME
                        ; AS IN THE BOX COMMAND
  12C5                  CLIP:
  12C5    2A 8000                 LHLD    WINPTR
  12C8    CD 12DF                 CALL    CLIPPER
  12CB    D8                      RC
  12CC    FDE5                    PUSH    Y
  12CE    FD2B                    DCX     Y
  12D0    FD2B                    DCX     Y
  12D2    2A 8000                 LHLD    WINPTR
  12D5    01 0004                 LXI     B,4
  12D8    09                      DAD     B
  12D9    CD 12DF                 CALL    CLIPPER
  12DC    FDE1                    POP     Y
  12DE    C9                      RET
                        ; CLIP COORDINATE ROUTINE
                        ; HL = PARM AREA START IN WINDOW TABLE
                        ; IY POINTS TO STACK FRAME SUCH THAT
                        ; SIZE IS 6 BYTES DOWN, COORDINATE 10 BYTES DOWN

  12DF                  CLIPPER:
  12DF    5E                      MOV     E,M     ; GET UPPER LIMIT
  12E0    23                      INX     H
  12E1    56                      MOV     D,M
  12E2    23                      INX     H
  12E3    4E                      MOV     C,M     ; GET LOWER LIMIT
  12E4    23                      INX     H
  12E5    46                      MOV     B,M
  12E6    C5                      PUSH    B
  12E7    FD6E0A                  MOV     L,CLP.C(Y)              ; HL=COO
                        RDINATE
  12EA    FD660B                  MOV     H,CLP.C+1(Y)
  12ED    FD4E06                  MOV     C,CLP.S(Y)              ; BC=SIZ
                        E
  12F0    FD4607                  MOV     B,CLP.S+1(Y)
  12F3    CD 136E                 CALL    ..TSTB          ; BARF IF <= 0
  12F6    2873                    JRZ     ..NODR
  12F8    0B                      DCX     B
  12F9    CD 137E                 CALL    ..DVBC          ; BC=BC DIVIDE 2

  12FC    A7                      ANA     A
  12FD    ED42                    DSBC    B               ; HL=LOWER
  12FF    FD750A                  MOV     CLP.C(Y),L              ; STUFF
                        BACK STUFF
  1302    FD740B                  MOV     CLP.C+1(Y),H
  1305    EB                      XCHG                    ; TO DE
  1306    CD 1386                 CALL    CPHLDE          ; IS LOWER>UL?
  1309    3860                    JRC     ..NODR          ; DONT DRAW
  130B    E3                      XTHL
  130C    CD 1386                 CALL    CPHLDE          ; IS LOWER < LOW
                        ER LIMIT?
  130F    381F                    JRC     ..LOK
```

```
1311    E5                  PUSH    H               ; SAVE LOWER LIM
                    IT
1312    EB                  XCHG                    ; HL=LOWER,DE=LL

1313    A7                  ANA     A
1314    ED52                DSBC    D               ; HL=LOWER - LIM
                    IT
1316    EB                  XCHG
1317    FD6E06              MOV     L,CLP.S(Y)              ; HL=SIZ
                    E
131A    FD6607              MOV     H,CLP.S+1(Y)
131D    19                  DAD     D
131E    FD7506              MOV     CLP.S(Y),L              ; STORE
                    BACK
1321    FD7407              MOV     CLP.S+1(Y),H
1324    CD 1377             CALL    ..TSTH          ; IF H<= 0 ABORT

1327    2841                JRZ     ..NOD1
1329    E1                  POP     H               ; SET COORDINATE
                     AT
132A    FD750A              MOV     CLP.C(Y),L              ; LOWER
                    LIMIT
132D    FD740B              MOV     CLP.C+1(Y),H
                    ; DEAL WITH OTHER END
1330    EB          ..LOK:  XCHG            ; DE=LOWER LIMIT
1331    FD6E0A              MOV     L,CLP.C(Y)              ; HL=COO
                    RDINATE
1334    FD660B              MOV     H,CLP.C+1(Y)
1337    FD4E06              MOV     C,CLP.S(Y)              ; BC=SIZ
                    E
133A    FD4607              MOV     B,CLP.S+1(Y)
133D    CD 136E             CALL    ..TSTB
1340    2829                JRZ     ..NODR
1342    0B                  DCX     B
1343    09                  DAD     B               ; ADD TO LOWER E
                    DGE
1344    EB                  XCHG                    ; UPPER TO DE
1345    CD 1386             CALL    CPHLDE          ; CAN WE DRAW?
1348    E1                  POP     H               ; H=UL
1349    281D                JRZ     ..UOK           ; JUMP IF ON EDG
                    E
134B    301F                JRNC    ..NOD2          ; IF UPPER < LOW
                    ER LIMIT DONT
134D    CD 1386             CALL    CPHLDE          ; IS UPPER > UL?

1350    3016                JRNC    ..UOK           ; NO PROB
1352    A7                  ANA     A               ; COMPUTE SIZE F
                    UDGE
1353    ED52                DSBC    D
1355    EB                  XCHG                    ; TO DE
1356    FD6E06              MOV     L,CLP.S(Y)
1359    FD6607              MOV     H,CLP.S+1(Y)
135C    19                  DAD     D               ; HL=NEW SIZE
135D    FD7506              MOV     CLP.S(Y),L
1360    FD7407              MOV     CLP.S+1(Y),H
```

```
    1363     CD 1377                    CALL     ..TSTH  ; IF HL<=0 ABORT
    1366     2804                       JRZ      ..NOD2
    1368     A7                ..UOK:   ANA      A                  ; RETURN CARRY C
                               LEAR
    1369     C9                         RET                         ; FOR GOOD GUYS
    136A     E1                ..NOD1:  POP      H                  ; BAD GUY - CLEA
                               N UP STACK
    136B     E1                ..NODR:  POP      H
    136C     37                ..NOD2:  STC                         ; CY FOR DONT DR
                               AW
    136D     C9                         RET
                               ; TEST FOR BC BEING <= 0
    136E     78                ..TSTB:  MOV      A,B
    136F     A7                         ANA      A
    1370     FA 1375                    JM       ..LESZ
    1373     B1                         ORA      C
    1374     C9                         RET
    1375     AF                ..LESZ:  XRA      A
    1376     C9                         RET
                               ; SIMILAR ROUTINE FOR HL
    1377     7C                ..TSTH:  MOV      A,H
    1378     A7                         ANA      A
    1379     FA 1375                    JM       ..LESZ
    137C     B5                         ORA      L
    137D     C9                         RET
                               ; DIVIDE BC BY 2
    137E     A7                ..DVBC:  ANA      A
    137F     78                         MOV      A,B
    1380     1F                         RAR
    1381     47                         MOV      B,A
    1382     79                         MOV      A,C
    1383     1F                         RAR
    1384     4F                         MOV      C,A
    1385     C9                         RET
                               ; *****
                               ; *
                               ; * ROUTINE TO COMPARE HL TO DE
                               ; * RETURNS CY SET FOR HL<DE (OR DE>HL)
                               ; * CY CLEAR, ZERO SET IF HL=DE
                               ; * CY CLEAR, ZERO CLEAR IF HL>DE (OR DE<HL)
                               ; *
                               ; *****
    1386     7C                CPHLDE:  MOV      A,H
    1387     AA                         XRA      D                  ; ARE SIGNS DIFF
                               ?
    1388     F2 1391                    JP       ..CK1              ; NO
    138B     EB                         XCHG                        ; YES - REVERSE
                               ARGS
    138C     CD 1391                    CALL     ..CK1              ; DO CHECK
    138F     EB                         XCHG                        ; BACK TO NORMAL

    1390     C9                         RET
    1391     7C                ..CK1:   MOV      A,H
    1392     BA                         CMP      D
    1393     C0                         RNZ
```

```
 1394   7D                      MOV     A,L
 1395   BB                      CMP     E
 1396   C9                      RET
                        ; *****
                        ; *
                        ; * CLEAR THE SCREEN, THE WHOLE SCREEN, AND NOTH
                        ING BUT THE SCREEN
                        ; *
                        ; *****
                                VERB    "CLEAR"[
 1397   1205           +        .WORD   .LINK.
 1399   05             +        .BYTE   ..0026-.-1
 139A   434C454152     +        .ASCII  "CLEAR"
 139F                  +..0026:]
 139F                   CLEARE:
 139F   C5                      PUSH    B
 13A0   21 4000                 LXI     H,4000H
 13A3   75                      MOV     M,L
 13A4   11 4001                 LXI     D,4001H
 13A7   01 0FFF                 LXI     B,0FFFH
 13AA   EDB0                    LDIR
                        ; RESET CX, CY TO ULHC OF SCREEN
 13AC   21 FFB1                 LXI     H,-79
 13AF   22 8002                 SHLD    CDXCEL
 13B2   21 0033                 LXI     H,51
 13B5   22 8004                 SHLD    CDYCEL
 13B8   C1                      POP     B
                                NEXT[
 13B9   FDE9           +        PCIY]
                        ; *****
                        ; *
                        ; * LINE DRAWER
                        ; * x-coord1 y-coord1 x-coord2 y-coord2 mode DRA
                        W x-coord2 y-coord2
                        ; *
                        ; *****
                                VERB    "DRAW"[
 13BB   1397           +        .WORD   .LINK.
 13BD   04             +        .BYTE   ..0027-.-1
 13BE   44524157       +        .ASCII  "DRAW"
 13C2                  +..0027:]
                                FRAME[
 13C2   FDE5           +        PUSH    Y
 13C4   FD21 0000      +        LXI     Y,0
 13C8   FD39           +        DADY    SP]
 13CA   FD5604                  MOV     D,FR.P2(Y)
 13CD   FD5E06                  MOV     E,FR.P3(Y)
 13D0   FD6608                  MOV     H,FR.P4(Y)
 13D3   FD6E0A                  MOV     L,FR.P5(Y)
 13D6   FD7E02                  MOV     A,FR.P1(Y)
 13D9   A7                      ANA     A
 13DA   280B                    JRZ     ..NODR
 13DC   C5                      PUSH    B
 13DD   CD 0CF2                 CALL    INCLOK
 13E0   CD 13F6                 CALL    DVECT
```

```
13E3    CD 0CF9                 CALL    DECLOK
13E6    C1                      POP     B
13E7                    ..NODR: UNFRAME[
13E7    FDE1          +         POP     Y]
13E9    E1                      POP     H
13EA    E1                      POP     H
13EB    D1                      POP     D
13EC    F1                      POP     PSW
13ED    F1                      POP     PSW
13EE    D5                      PUSH    D
13EF    E5                      PUSH    H
                                NEXT[
13F0    FDE9          +         PCIY]
                        ; STRANGE TABLE ...
13F2    0055AAFF        MSKTBL: .BYTE   0,055H,0AAH,0FFH
                        ; *****
                        ; *
                        ; * THIS ROUTINE IMPLEMENTS LARRY LIVERMORE'S VE
                        CTOR
                        ; * DRAWING ALGORITHM.
                        ; * INPUT:      L = X1 COORDINATE
                        ; *     H = Y1 COORDINATE
                        ; *     E = X2 COORDINATE
                        ; *     D = Y2 COORDINATE
                        ; * OUTPUT:     A = 0, DE = X2,Y2, BC, HL CLOBBE
                        RED
                        ; * RAM USE:
                        ; * INCRO               2 BYTES HOLDS X,Y INCREM
                        ENTS
                        ; * MNMX                2 BYTES HOLDS MIN, MAX D
                        ELTAS ,
                        ; * PIXVAL      1 BYTE HOLDS LEFT JUSTIFIED PIXE
                        L FOR XOR WRITE
                        ; * WRMODE      1 BYTE HOLDS PLOP-XOR FLAG
                        ; *
                        ; *****
                        ; COMPUTE DELTAS AND ABS(DELTAS)
13F6    D5              DVECT:  PUSH    D
13F7    45                      MOV     B,L     ; COMPUTE Y STUFF
13F8    4B                      MOV     C,E
13F9    CD 1448                 CALL    CDELTA
13FC    58                      MOV     E,B
13FD    69                      MOV     L,C
13FE    44                      MOV     B,H
13FF    4A                      MOV     C,D     ; AND X STUFF
1400    CD 1448                 CALL    CDELTA
1403    61                      MOV     H,C
1404    50                      MOV     D,B
1405    22 800D                 SHLD    INCRO
                        ; DECIDE WHICH IS BIGGER - CALL BIGGER MX, SMALL
                        ER MN
1408    0E00                    MVI     C,0
140A    7A                      MOV     A,D
140B    BB                      CMP     E
140C    3803                    JRC     ..DV1
```

```
 140E    53                     MOV     D,E
 140F    5F                     MOV     E,A
 1410    0C                     INR     C
 1411    7A            ..DV1:   MOV     A,D
 1412    CB3F                   SRLR    A
 1414    47                     MOV     B,A
 1415    EB                     XCHG
 1416    22 800F                SHLD    MNMX
 1419    D1                     POP     D
 141A    7D                     MOV     A,L
 141B    3C                     INR     A
 141C    F5            VECT2:   PUSH    PSW
 141D    FD7E02                 MOV     A,FR.P1(Y)
 1420    CD 0EEB                CALL    POINTR  ; DRAW THE POINT!
                       ; NOW INCREMENT COORDINATES
 1423    2A 800F       VECT2A:  LHLD    MNMX
 1426    78                     MOV     A,B
 1427    84                     ADD     H
 1428    BD                     CMP     L
 1429    380D                   JRC     VECT4   ; JUMP IF NOT
                       ; M+MN IS >=MX, SET M=MOD(M+MN,MX)
 142B    95                     SUB     L
 142C    47                     MOV     B,A
                       ; INCREMENT BOTH DIRECTIONS
 142D    2A 800D                LHLD    INCR0
 1430    7A                     MOV     A,D     ; CONFUSE Y
 1431    84                     ADD     H
 1432    57                     MOV     D,A
 1433    7B            VECT3:   MOV     A,E     ; THEN X
 1434    85                     ADD     L
 1435    5F                     MOV     E,A
 1436    180B                   JMPR    VECT5
                       ; M + MN IS < MX, SET M = M + MN
 1438    47            VECT4:   MOV     B,A
                       ; INCREMENT ONLY MAX DIMENSION
 1439    2A 800D                LHLD    INCR0
 143C    79                     MOV     A,C     ; C = DIRECTION FLAG
 143D    0F                     RRC
 143E    30F3                   JRNC    VECT3   ; 0=>X, SO GO DO IT
 1440    7A                     MOV     A,D     ; Y CASE
 1441    84                     ADD     H
 1442    57                     MOV     D,A
                       ; END OF LOOP
 1443    F1            VECT5:   POP     PSW
 1444    3D                     DCR     A
 1445    20D5                   JRNZ    VECT2
 1447    C9                     RET
                       ; SUBROUTINE TO COMPUTE DELTA AND INCREMENT FOR
                       TWO COORDINATES
 1448    E5            CDELTA:  PUSH    H
 1449    D5                     PUSH    D
 144A    69                     MOV     L,C
 144B    CD 146A                CALL    SGNEXT
 144E    EB                     XCHG
 144F    68                     MOV     L,B
```

```
1450    CD 146A                 CALL    SGNEXT
1453    AF                      XRA     A
1454    ED52                    DSBC    D
1456    B4                      ORA     H
1457    2807                    JRZ     ..CD1
1459    4F                      MOV     C,A
145A    7D                      MOV     A,L
145B    2F                      CMA
145C    3C                      INR     A
145D    47                      MOV     B,A
145E    1807                    JMPR    ..CD3
1460    B5              ..CD1:  ORA     L
1461    2802                    JRZ     ..CD2
1463    3E01                    MVI     A,1
1465    45              ..CD2:  MOV     B,L
1466    4F                      MOV     C,A
1467    D1              ..CD3:  POP     D
1468    E1                      POP     H
1469    C9                      RET
                        ; SIGN EXTENSION SUBROUTINE
146A    2600            SGNEXT: MVI     H,0
146C    7D                      MOV     A,L
146D    A7                      ANA     A
146E    F0                      RP
146F    25                      DCR     H
1470    C9                      RET
                        ; ABSOLUTE VALUE ROUTINE
                        ; THIS ROUTINE COMPUTES THE ABSOLUTE VALUE OF TH
                        E ARGUMENT
                        ; PASSED IN A.  THE RESULT IS RETURNED IN A.
1471    A7              ABS:    ANA     A
1472    F0                      RP
1473    2F                      CMA
1474    3C                      INR     A
1475    C9                      RET
                        ; *****
                        ; *
                        ; * RELATIVE TO ABSOLUTE CONVERSION ROUTINE
                        ; * WITH CLIPPING AGAINST BOUNDARYS OF CURRENT W
                        INDOW
                        ; *
                        ; * D=Y COORDINATE E=X COORDINATE
                        ; * HL=ABSOLUTE ADDRESS (NOT MAGIC)
                        ; * A=SHIFT AMOUNT
                        ; * MINUS SET IF COORDINATE OUTSIDE OF WINDOW
                        ; *
                        ; *****
1476                    R2ACLP:
1476    2A 8000                 LHLD    WINPTR
1479    7B                      MOV     A,E             ; CHECK X UPPER
147A    BE                      CMP     M
147B    23                      INX     H
147C    23                      INX     H
147D    FA 1484                 JM      ..OKX
1480    2802                    JRZ     ..OKX
```

```
1482    BE                      CMP     M
1483    F8                      RM
1484    23              ..OKX:  INX     H
1485    23                      INX     H
1486    7A                      MOV     A,D
1487    BE                      CMP     M
1488    FA 1491                 JM      R2A
148B    2804                    JRZ     R2A
148D    23                      INX     H
148E    23                      INX     H
148F    BE                      CMP     M
1490    F8                      RM
                        ; *****
                        ; *
                        ; * NONCLIPPING ENTRY POINT
                        ; *
                        ; *****
1491    C5              R2A:    PUSH    B
1492    7A                      MOV     A,D
1493    2F                      CMA
1494    C634                    ADI     52
1496    6F                      MOV     L,A
1497    2600                    MVI     H,0
1499    29                      DAD     H
149A    29                      DAD     H
149B    29                      DAD     H
149C    44                      MOV     B,H
149D    4D                      MOV     C,L
149E    29                      DAD     H
149F    29                      DAD     H
14A0    09                      DAD     B
14A1    7B                      MOV     A,E
14A2    C64F                    ADI     79
14A4    0F                      RRC
14A5    0F                      RRC
14A6    E63F                    ANI     3FH
14A8    4F                      MOV     C,A
14A9    0600                    MVI     B,0
14AB    09                      DAD     B
14AC    CBF4                    SET     6,H
14AE    7B                      MOV     A,E
14AF    3D                      DCR     A
14B0    E603                    ANI     3
14B2    C1                      POP     B
14B3    C9                      RET
14B4                    CMPM:
14B4    AE                      XRA     M       ; DO SIGNS DIFFER?
14B5    FA 14BB                 JM      ..REVR  ; JUMP IF SO
14B8    AE                      XRA     M       ; ELSE FIX
14B9    BE                      CMP     M       ; AND COMPARE
14BA    C9                      RET
14BB    AE              ..REVR: XRA     M       ; SAME FIX
14BC    BE                      CMP     M
14BD    3F                      CMC             ; REVERSE SENSE OF CY
14BE    C9                      RET
```

```
                         ; LOAD ADDR OF SMALL FONT
                                 CONSTANT[SMALL,SMLFNT][
  14BF    13BB          +        .WORD   .LINK.
  14C1    05            +        .BYTE   ..0028-.-1
  14C2    534D414C4C    +        .ASCII  "SMALL"
  14C7    CD 0CEB       +:]      CALL    CONST
  14CA    14CC          +        .WORD   SMLFNT]
  14CC    20             SMLFNT: .BYTE   20H
  14CD    03050406               .BYTE   3,5,4,6
  14D1    1B3C                   .WORD   FNT35
  14D3    01                     .BYTE   1
                                 CONSTANT[LARGE,LRGFNT][
  14D4    14BF          +        .WORD   .LINK.
  14D6    05            +        .BYTE   ..0029-.-1
  14D7    4C41524745    +        .ASCII  "LARGE"
  14DC    CD 0CEB       +:]      CALL    CONST
  14DF    14E1          +        .WORD   LRGFNT]
  14E1                   LRGFNT:
  14E1    20                     .BYTE   20H
  14E2    05070608               .BYTE   5,7,6,8
  14E6    196E                   .WORD   FNT57
  14E8    01                     .BYTE   1
                         ; *****
                         ; *
                         ; * VERB TO TYPE CHARACTER ON 'BUILT IN' SCREEN
                         WINDOW
                         ; * character CDOUT .
                         ; *
                         ; *****
                                 VERB    "CDOUT"[
  14E9    14D4          +        .WORD   .LINK.
  14EB    05            +        .BYTE   ..0030-.-1
  14EC    43444F5554    +        .ASCII  "CDOUT"
  14F1                  +..0030:]
  14F1    D1                     POP     D       ; DE=CHAR
  14F2    FDE5                   PUSH    Y       ; WE WILL BE NEEDING THI
                         S
  14F4    2A 8000                LHLD    WINPTR  ; REMEMBER PREVIOUS WIND
                         OW POINTER VALUE
  14F7    E5                     PUSH    H
  14F8    2A 8002                LHLD    CDXCEL
  14FB    E5                     PUSH    H
  14FC    2A 8004                LHLD    CDYCEL
  14FF    E5                     PUSH    H
  1500    2A 8006                LHLD    CDCCEL
  1503    E5                     PUSH    H
  1504    D5                     PUSH    D               ; PUT CHAR ON
  1505    2A 8008                LHLD    CDFCEL          ; FONT
  1508    E5                     PUSH    H
  1509    CD 0CF2                CALL    INCLOK
  150C    2A 800A                LHLD    CDWCEL
  150F    22 8000                SHLD    WINPTR
  1512    FD21 1518              LXI     Y,..CUMB
  1516    181A                   JMPR    CHARO
  1518                   ..CUMB:
```

```
1518    E1                      POP     H
1519    22 8004                 SHLD    CDYCEL
151C    E1                      POP     H
151D    22 8002                 SHLD    CDXCEL
1520    E1                      POP     H
1521    22 8000                 SHLD    WINPTR
1524    FDE1                    POP     Y
1526    CD 0CF9                 CALL    DECLOK
                                NEXT[
1529    FDE9            +       PCIY]
                        ; *****
                        ; *
                        ; * CHARACTER DISPLAY VERB
                        ; * x-coord y-coord color character fontaddress
                        CHAR x-coord y-coord .
                        ; *
                        ; *****
                        ; PARMS ON STACK FRAME
000C                    CF.X=FR.P6
000A                    CF.Y=FR.P5
0008                    CF.M=FR.P4
0006                    CF.C=FR.P3
0004                    CF.F=FR.P2
                        ; FIELDS IN FONT DESCRIPTOR
0000                    FD.BASE=0
0001                    FD.XCS=1
0002                    FD.YCS=2
0003                    FD.XF=3
0004                    FD.YF=4
0005                    FD.AD=5
0007                    FD.FLG=7
                        ; BITS IN STATUS BYTE OF FONT DESCRIPTOR
0000                    FDF.XL=0        ; TRANSLATE LOWER CASE TO UPPER
                                VERB    "CHAR"[
152B    14E9            +       .WORD   .LINK.
152D    04              +       .BYTE   ..0031-.-1
152E    43484152        +       .ASCII  "CHAR"
1532                    +..0031:]
1532                    CHAR0:
1532    CD 0CF2                 CALL    INCLOK
1535    DDE3                    XTIX
1537    C5                      PUSH    B
                                FRAME[
1538    FDE5            +       PUSH    Y
153A    FD21 0000       +       LXI     Y,0
153E    FD39            +       DADY    SP]
1540    FD7E06                  MOV     A,CF.C(Y)       ; CHECK FOR WIER
                        D CHARS
1543    FE0A                    CPI     LF              ; LIKE LINEFEED
1545    2837                    JRZ     ..DONE          ; IGNORE THAT GU
                        Y
1547    FE0D                    CPI     NL
1549    200B                    JRNZ    ..NONL
                        ; NEW LINE CASE
154B    CD 1675                 CALL    RESCX           ; RESET CX
```

CC
CT CL
CB CR

LC

FT

USE STACK

```
  154E    CD 168B               CALL    CYSCROLL        ; SCROLL IF CY I
                        S OFFSCREEN
  1551    CD 174C               CALL    BUMPCY          ; ADVANCE CY
  1554    1828                  JMPR    ..DONE
  1556                  ..NONL:
  1556    FE08                  CPI     RUBOUT          ; HOW ABOUT RUBO
                        UT?
  1558    2010                  JRNZ    ..NORB
  155A    CD 1633               CALL    LEFTX           ; MOVE CX LEFT
  155D    CD 170C               CALL    XCHECK
  1560    301C                  JRNC    ..DONE          ; JUMP IF OK
  1562    CD 1648               CALL    UPY             ; ELSE BACK UP Y

  1565    CD 165B               CALL    FINDLAST ; FIND LAST CHAR POS ON
                         PREV LINE
  1568    1814                  JMPR    ..DONE
                        ; NORMAL
  156A                  ..NORB:
  156A    CD 170C               CALL    XCHECK          ; IS X OFFSCREEN

  156D    3006                  JRNC    ..XON
  156F    CD 1675               CALL    RESCX           ; RESET CX
  1572    CD 174C               CALL    BUMPCY          ; ADVANCE CY
  1575                  ..XON:
  1575    CD 168B               CALL    CYSCROLL        ; SCROLL IF CY N
                        EEDS IT
  1578    CD 158A               CALL    DCHAR           ; BUG CHARACTER
                        DISPLAYER
  157B    CD 1739               CALL    BUMPCX          ; ADVANCE CX
  157E                  ..DONE:
  157E    CD 0CF9               CALL    DECLOK
                                UNFRAME[
  1581    FDE1          +       POP     Y]
  1583    C1                    POP     B
  1584    DDE1                  POP     X               ; FIX RETURN STA
                        CK
  1586    E1                    POP     H               ; EAT CHARACTER
  1587    E1                    POP     H               ; AND MODE
                                NEXT                    ; GOOD-BYE[
  1588    FDE9          +       PCIY]
                        ; SUBROUTINE TO DISPLAY CHARACTER ON THE SCREEN
  158A                  DCHAR:
                        ; FIRST DRAW BOX TO ERASE
  158A    FDE5                  PUSH    Y
  158C    FD5E0C                MOV     E,CF.X(Y)
  158F    FD560D                MOV     D,CF.X+1(Y)
  1592    FD6E0A                MOV     L,CF.Y(Y)
  1595    FD660B                MOV     H,CF.Y+1(Y)
  1598    D5                    PUSH    D
  1599    E5                    PUSH    H
  159A    DD5E03                MOV     E,FD.XF(X)
  159D    1600                  MVI     D,0
  159F    DD6E04                MOV     L,FD.YF(X)
  15A2    62                    MOV     H,D
  15A3    D5                    PUSH    D
```

check
interface

```
 15A4     E5                  PUSH    H
 15A5     2E04                MVI     L,4
 15A7     E5                  PUSH    H
 15A8     FD21 15AF           LXI     Y,..CUMA
 15AC     C3 120B             JMP     DOBOX
 15AF                 ..CUMA:
 15AF     FDE1                POP     Y
                      ; PERFORM PATTERN LOOKUP
 15B1     FD5608              MOV     D,CF.M(Y)
 15B4     FD7E06              MOV     A,CF.C(Y)         ; A=CHAR, D=MODE

                      ; DO WE WANT LOWER TO UPPER TRANSLATION?
 15B7     DDCB0746            BIT     FDF.XL,FD.FLG(X)
 15BB     280B                JRZ     ..NOTR
                      ; YES - ARE WE IN RANGE FOR IT?
 15BD     FE61                CPI     'a'      ; lower case a
 15BF     3807                JRC     ..NOTR
 15C1     FE7B                CPI     'z'+1
 15C3     3003                JRNC    ..NOTR
 15C5     14                  INR     D                 ; BUMP COLOR
 15C6     D620                SUI     20H      ; FUDGE TO CORRECT
 15C8     FE20        ..NOTR: CPI     20H      ; CONTROL CHARACTER?
 15CA     3004                JRNC    ..NOTC   ; NOPE
 15CC     C640                ADI     40H
 15CE     14                  INR     D
 15CF     14                  INR     D
 15D0     DD9600      ..NOTC: SUB     FD.BASE(X)
 15D3     5F                  MOV     E,A
 15D4     7A                  MOV     A,D
 15D5     E603                ANI     3
 15D7     2001                JRNZ    ..COK
 15D9     3C                  INR     A
 15DA     07          ..COK:  RLC
 15DB     07                  RLC
 15DC     D319                OUT     XPAND
 15DE     1600                MVI     D,0
 15E0     DD7E01              MOV     A,FD.XCS(X)       ; CONVERT X BITS
                       INTO BYTES
 15E3     C607                ADI     7
 15E5     0F                  RRC
 15E6     0F                  RRC
 15E7     0F                  RRC
 15E8     E61F                ANI     1FH
 15EA     47                  MOV     B,A
 15EB     DD4E02              MOV     C,FD.YCS(X)
 15EE     DD6E05              MOV     L,FD.AD(X)
 15F1     DD6606              MOV     H,FD.AD+1(X)
 15F4     C5                  PUSH    B
 15F5     C5          ..MPY1: PUSH    B
 15F6     19          ..MPY2: DAD     D
 15F7     10FD                DJNZ    ..MPY2
 15F9     C1                  POP     B
 15FA     0D                  DCR     C
 15FB     20F8                JRNZ    ..MPY1
 15FD     E5                  PUSH    H
```

```
  15FE    DD7E01                  MOV     A,FD.XCS(X)       ; FUDGE COORDINA
                          TE TO ULHC
  1601    CD 1771                 CALL    COMLV
  1604    FD860C                  ADD     CF.X(Y)
  1607    5F                      MOV     E,A
  1608    DD7E02                  MOV     A,FD.YCS(X)
  160B    CD 1778                 CALL    COMUV
  160E    FD860A                  ADD     CF.Y(Y)
  1611    57                      MOV     D,A
  1612    CD 1491                 CALL    R2A
  1615    F628                    ORI     XORWMR+XPWMR
  1617    D30C                    OUT     MAGIC
  1619    CBB4                    RES     6,H               ; MAKE ADDR MAGI
                          C
  161B    D1                      POP     D
  161C    C1                      POP     B
  161D    C5              ..WX1:  PUSH    B
  161E    E5                      PUSH    H
  161F    1A              ..WX2:  LDAX    D
  1620    13                      INX     D
  1621    77                      MOV     M,A
  1622    23                      INX     H
  1623    77                      MOV     M,A
  1624    23                      INX     H
  1625    10F8                    DJNZ    ..WX2
  1627    70                      MOV     M,B
  1628    23                      INX     H
  1629    70                      MOV     M,B
  162A    E1                      POP     H
  162B    0E28                    MVI     C,BYTEPL
  162D    09                      DAD     B
  162E    C1                      POP     B
  162F    0D                      DCR     C
  1630    20EB                    JRNZ    ..WX1
  1632    C9                      RET
                          ; MOVE X TO THE LEFT
  1633                    LEFTX:
  1633    FD6E0C                  MOV     L,CF.X(Y)
  1636    FD660D                  MOV     H,CF.X+1(Y)
  1639    DD5E03                  MOV     E,FD.XF(X)
  163C    1600                    MVI     D,0
  163E    A7                      ANA     A
  163F    ED52                    DSBC    D
  1641    FD750C                  MOV     CF.X(Y),L
  1644    FD740D                  MOV     CF.X+1(Y),H
  1647    C9                      RET
                          ; MOVE CY UP
  1648                    UPY:
  1648    FD6E0A                  MOV     L,CF.Y(Y)
  164B    FD660B                  MOV     H,CF.Y+1(Y)
  164E    DD5E04                  MOV     E,FD.YF(X)
  1651    1600                    MVI     D,0
  1653    19                      DAD     D
  1654    FD750A                  MOV     CF.Y(Y),L
  1657    FD740B                  MOV     CF.Y+1(Y),H
```

```
 165A     C9                          RET
                             ; SET CX TO LAST POSITION ON LINE
 165B                        FINDLAST:
 165B     CD 1675                     CALL     RESCX
 165E     FD6E0C             ..MORE:  MOV      L,CF.X(Y)
 1661     FD660D                      MOV      H,CF.X+1(Y)
 1664     E5                          PUSH     H
 1665     CD 1739                     CALL     BUMPCX
 1668     CD 170C                     CALL     XCHECK
 166B     E1                          POP      H
 166C     30F0                        JRNC     ..MORE
 166E     FD750C                      MOV      CF.X(Y),L
 1671     FD740D                      MOV      CF.X+1(Y),H
 1674     C9                          RET
                             ; RESET CX TO LHS OF WINDOW
 1675     DD7E01             RESCX:   MOV      A,FD.XCS(X)
 1678     3D                          DCR      A
 1679     CD 1778                     CALL     COMUV
 167C     6F                          MOV      L,A
 167D     2600                        MVI      H,0
 167F     CD 1761                     CALL     DEPARM
 1682     02                          .BYTE    WXL
 1683     19                          DAD      D
 1684     FD750C                      MOV      CF.X(Y),L
 1687     FD740D                      MOV      CF.X+1(Y),H
 168A     C9                          RET
                             ; SCROLL IF CY OFFSCREEN AT BOTTOM
 168B                        CYSCROLL:
                             ; CHECK FOR Y ABOVE UPPER LIMIT
 168B     DD7E02                      MOV      A,FD.YCS(X)
 168E     CD 1778                     CALL     COMUV
 1691     5F                          MOV      E,A
 1692     1600                        MVI      D,0
 1694     FD6E0A                      MOV      L,CF.Y(Y)
 1697     FD660B                      MOV      H,CF.Y+1(Y)
 169A     19                          DAD      D
 169B     CD 1761                     CALL     DEPARM
 169E     04                          .BYTE    WYU
 169F     CD 1386                     CALL     CPHLDE
                             ; IF OFFSCREEN AT TOP RESET TO TOP OF SCREEN
 16A2     D4 16DC                     CNC      RESCY
 16A5     CD 16F4                     CALL     YCHECK   ; IS Y OFFSCREEN AT BOTT
                             OM
 16A8     D0                          RNC               ; NO
                             ; SET CY TO BOTTOM MOST LINE THAT IS OK MEASURED
                              FROM TOP
 16A9     CD 16DC                     CALL     RESCY
 16AC     FD6E0A             ..LOPR:  MOV      L,CF.Y(Y)
 16AF     FD660B                      MOV      H,CF.Y+1(Y)
 16B2     E5                          PUSH     H
 16B3     CD 174C                     CALL     BUMPCY
 16B6     CD 16F4                     CALL     YCHECK
 16B9     E1                          POP      H
 16BA     30F0                        JRNC     ..LOPR
 16BC     FD750A                      MOV      CF.Y(Y),L
```

```
    16BF     FD740B                    MOV     CF.Y+1(Y),H
                              ; BUILD CALL TO SCROLLER
    16C2     DD4E04                    MOV     C,FD.YF(X)        ; BC=SCROLL AMOU
                              NT
    16C5     0600                      MVI     B,0
    16C7     FDE5                      PUSH    Y
    16C9     21 0000                   LXI     H,0
    16CC     E5                        PUSH    H
    16CD     E5                        PUSH    H
    16CE     24                        INR     H          ; SET HL=256
    16CF     E5                        PUSH    H          ; AND LET CLIPPER FIX IT

    16D0     E5                        PUSH    H
    16D1     C5                        PUSH    B
    16D2     FD21 16D9                 LXI     Y,..CUMA
    16D6     C3 1069                   JMP     SCROLE
    16D9                      ..CUMA:
    16D9     FDE1                      POP     Y
    16DB     C9                        RET
                              ; RESET CY TO SCREEN TOP
    16DC                      RESCY:
    16DC     CD 1761                   CALL    DEPARM
    16DF     04                        .BYTE   WYU
    16E0     EB                        XCHG
    16E1     DD7E02                    MOV     A,FD.YCS(X)
    16E4     CD 1778                   CALL    COMUV
    16E7     5F                        MOV     E,A
    16E8     1600                      MVI     D,0
    16EA     A7                        ANA     A
    16EB     ED52                      DSBC    D
    16ED     FD750A                    MOV     CF.Y(Y),L
    16F0     FD740B                    MOV     CF.Y+1(Y),H
    16F3     C9                        RET
                              ; CHECK FOR CY ONSCREEN AT BOTTOM
                              ; CY SET IF SCROLL NEEDED
    16F4                      YCHECK:
    16F4     DD7E02                    MOV     A,FD.YCS(X)
    16F7     CD 1771                   CALL    COMLV
    16FA     5F                        MOV     E,A
    16FB     16FF                      MVI     D,0FFH
    16FD     FD6E0A                    MOV     L,CF.Y(Y)
    1700     FD660B                    MOV     H,CF.Y+1(Y)
    1703     19                        DAD     D
    1704     CD 1761                   CALL    DEPARM
    1707     06                        .BYTE   WYL
    1708     CD 1386                   CALL    CPHLDE
    170B     C9                        RET
                              ; ROUTINE TO CHECK CX FOR BEING ONSCREEN
                              ; RETURNS CY SET IF OFFSCREEN
    170C                      XCHECK:
    170C     FD6E0C                    MOV     L,CF.X(Y)
    170F     FD660D                    MOV     H,CF.X+1(Y)
    1712     DD7E01                    MOV     A,FD.XCS(X)
    1715     CD 1778                   CALL    COMUV
    1718     5F                        MOV     E,A
```

check interface

```
1719    1600                    MVI     D,0
171B    E5                      PUSH    H
171C    19                      DAD     D
171D    CD 1761                 CALL    DEPARM
1720    00                      .BYTE   WXR
1721    EB                      XCHG                ; HL=LMT, DE=EXTENT
1722    CD 1386                 CALL    CPHLDE
1725    E1                      POP     H
1726    D8                      RC
1727    DD7E01                  MOV     A,FD.XCS(X)
172A    CD 1771                 CALL    COMLV
172D    5F                      MOV     E,A
172E    16FF                    MVI     D,0FFH
1730    19                      DAD     D
1731    CD 1761                 CALL    DEPARM
1734    02                      .BYTE   WXL
1735    CD 1386                 CALL    CPHLDE
1738    C9                      RET
                        ; ROUTINE TO BUMP CX
1739    DD5E03          BUMPCX: MOV     E,FD.XF(X)
173C    1600                    MVI     D,0
173E    FD6E0C                  MOV     L,CF.X(Y)
1741    FD660D                  MOV     H,CF.X+1(Y)
1744    19                      DAD     D
1745    FD750C                  MOV     CF.X(Y),L
1748    FD740D                  MOV     CF.X+1(Y),H
174B    C9                      RET
                        ; SUBTRACT YF FROM CY
174C    DD5E04          BUMPCY: MOV     E,FD.YF(X)
174F    1600                    MVI     D,0
1751    FD6E0A                  MOV     L,CF.Y(Y)
1754    FD660B                  MOV     H,CF.Y+1(Y)
1757    A7                      ANA     A
1758    ED52                    DSBC    D
175A    FD750A                  MOV     CF.Y(Y),L
175D    FD740B                  MOV     CF.Y+1(Y),H
1760    C9                      RET
                        ; ROUTINE TO GET WINDOW PARM INTO DE
1761                    DEPARM:
1761    E3                      XTHL
1762    5E                      MOV     E,M
1763    23                      INX     H
1764    E3                      XTHL
1765    1600                    MVI     D,0
1767    E5                      PUSH    H
1768    2A 8000                 LHLD    WINPTR
176B    19                      DAD     D
176C    5E                      MOV     E,M
176D    23                      INX     H
176E    56                      MOV     D,M
176F    E1                      POP     H
1770    C9                      RET
1771                    COMLV:
1771    3D                      DCR     A
1772    0F                      RRC
```

```
1773    E67F                    ANI     7FH
1775    2F                      CMA
1776    3C                      INR     A
1777    C9                      RET
1778                    COMUV:
1778    0F                      RRC
1779    E67F                    ANI     7FH
177B    C9                      RET
                        ; *****
                        ; *
                        ; * EASY ENTRY KEYPAD SCANNER
                        ; *  EZKP keycode .
                        ; *
                        ; *****
                                VERB    "EZKP"[
177C    152B            +       .WORD   .LINK.
177E    04              +       .BYTE   ..0032-.-1
177F    455A4B50        +       .ASCII  "EZKP"
1783                    +..0032:]
1783    C5                      PUSH    B
1784    CD 178E                 CALL    KEYPSN
1787    6F                      MOV     L,A
1788    2600                    MVI     H,0
178A    C1                      POP     B
178B    E5                      PUSH    H
                                NEXT[
178C    FDE9            +       PCIY]
                        ; EASY ENTRY KEYPAD SCAN SUBROUTINE
                        ;
178E    01 0414         KEYPSN: LXI     B,0414H         ; BC=COL #/PORT
                        #
1791    11 8027                 LXI     D,KEYPTK
1794    ED78            ..SCN1: INP     A               ; GET A COL
1796    E63F                    ANI     3FH             ; ISOLATE RELEVA
                        NT
1798    2006                    JRNZ    ..SCN2          ; JUMP IF SOMETH
                        ING THERE
179A    0C                      INR     C               ; BUMP PORT
179B    10F7                    DJNZ    ..SCN1
179D    AF                      XRA     A
179E    12                      STAX    D               ; SAY NOTHIN FOU
                        ND
179F    C9                      RET
                        ; GOT SOMETHANG - LOOK FOR THE HOT BIT
17A0    05              ..SCN2: DCR     B               ; FUDGE COL TO 0
                        -3
17A1    0E00                    MVI     C,0             ; C=BIT COUNTER
17A3    0F              ..SCN4: RRC                     ; COUNT UNTIL BI
                        T SHOWS UP
17A4    3803                    JRC     ..SCN3
17A6    0C                      INR     C
17A7    18FA                    JMPR    ..SCN4
17A9    79              ..SCN3: MOV     A,C             ; COMBINE WITH C
                        OL #
17AA    07                      RLC
```

```
17AB    07                      RLC
17AC    B0                      ORA     B
17AD    3C                      INR     A               ; GIVING KEYCODE
                         1-24
17AE    47                      MOV     B,A
17AF    1A                      LDAX    D               ; DIFERENT FROM
                        LAST?
17B0    A8                      XRA     B
17B1    C8                      RZ                      ; NO
17B2    78                      MOV     A,B
17B3    12                      STAX    D
17B4    C9                      RET
                        ; *****
                        ; *
                        ; * CHROMERICS KEYBOARD SCAN ROUTINE
                        ; *
                        ; *****
                        ; EQUATES:
0080                    SLBITM=80H      ; SHIFT LOCK BIT IN KEYFLG AND L
                        EDS
0040                    MDBITM=40H      ; MODE BIT SAME STORY
0003                    CK1RAM=3        ; LOCATION OF CONTROL KEYS
0003                    CK1BIT=3
0007                    CK2RAM=7
0004                    CK2BIT=4
0000                    SK1RAM=0        ; LOCATION OF SHIFT KEYS
0000                    SK1BIT=0
0007                    SK2RAM=7
0002                    SK2BIT=2
0000                    TOKRAM=0        ; LOCATION OF TOKEN KEY
0002                    TOKBIT=2
0001                    SHYLOK=1        ; KEYCODE NUMBER FOR SHIFT LOCK
                        KEY
0004                    ESCKEY=4        ; KEYCODE NUMBER OF ESCAPE KEY
17B5                    KEYSCN:
17B5    06FE                    MVI     B,0FEH          ; B=COL SET BIT
17B7    11 0000                 LXI     D,0             ; DE=COLUMN #
17BA    78              ..SCAN: MOV     A,B
17BB    CD 186B                 CALL    OUT98
17BE    CD 1873                 CALL    IN98
17C1    2F                      CMA
17C2    21 801D                 LXI     H,KEYTRK
17C5    19                      DAD     D
17C6    77                      MOV     M,A
17C7    21 187B                 LXI     H,KEYMES
17CA    19                      DAD     D
17CB    A6                      ANA     M
17CC    200A                    JRNZ    ..LIVE
17CE    1C                      INR     E
17CF    CB00                    RLCR    B               ; SHOVE OVER THE
                         MASK
17D1    38E7                    JRC     ..SCAN          ; LOOP TILL FALL
                        S OFF END
17D3    AF                      XRA     A               ; FAILURE-NAIL O
                        LDKEY MEM
```

```
 17D4    32 8026              STA     OLDKEY
 17D7    C9                   RET                       ; HOME TO MAMMA
                      ; A KEY IS DOWN - CONVERT TO INTERMEDIATE KEYCOD
                      E
                      ; FIRST CONVERT TO BIT NUMBER
 17D8    0F           ..LIVE: RRC
 17D9    3803                 JRC     ..BITF
 17DB    14                   INR     D
 17DC    18FA                 JMPR    ..LIVE
 17DE    7B           ..BITF: MOV     A,E      ; COLUMN #
 17DF    A7                   ANA     A
 17E0    07                   RLC                       ; * 8
 17E1    07                   RLC
 17E2    07                   RLC
 17E3    82                   ADD     D                 ; + BIT #
 17E4    4F                   MOV     C,A               ; KEYCODE TO C
                      ; IS THE KEYCODE THE SAME AS THAT FOUND ON PREVI
                      OUS SCAN??
 17E5    21 8026              LXI     H,OLDKEY
 17E8    BE                   CMP     M
 17E9    C8                   RZ                        ; QUIT IF SO
 17EA    77                   MOV     M,A               ; ELSE UPDATE TH
                      ANGS
                      ;
 17EB    21 8025              LXI     H,KEYFLG          ; POINT AT SHIFT
                      LOCK/MODE FLAGS
                      ; CHECK FOR SHIFT LOCK KEYPRESS
 17EE    FE01                 CPI     SHYLOK
                      ;       JMPR    ..NOSL
 17F0    2005                 JRNZ    ..NOSL
 17F2    7E                   MOV     A,M               ; YEP - TOGGLE S
                      HIFT LOCK BIT
 17F3    EE80                 XRI     SLBITM
 17F5    1816                 JMPR    ..ULED            ; JUMP TO UPDATE

                      ; CHECK FOR CONTROL KEY
 17F7    3A 8020      ..NOSL: LDA     KEYTRK+CK1RAM
 17FA    CB5F                 BIT     CK1BIT,A
 17FC    2007                 JRNZ    ..CKDN
 17FE    3A 8024              LDA     KEYTRK+CK2RAM
 1801    CB67                 BIT     CK2BIT,A
 1803    280F                 JRZ     ..NOCK
                      ; WE GOT A CONTROL KEY - DO WE HAVE ESCAPE AS WE
                      LL?
 1805    79           ..CKDN: MOV     A,C
 1806    FE04                 CPI     ESCKEY
 1808    2005                 JRNZ    ..NOES
 180A    7E                   MOV     A,M               ; YEP - TOGGLE M
                      ODE BIT
 180B    EE40                 XRI     MDBITM
 180D    77           ..ULED: MOV     M,A               ; SHIFTLOCK JOIN
                      S HERE TOO
 180E    C9                   RET
                      ; NO ESCAPE - NORMAL CONTROL KEY
 180F    21 1903      ..NOES: LXI     H,CKTBL
```

```
  1812    181B                 JMPR    ..LOOK
                    ; HOW ABOUT SHIFT KEY?
  1814    3A 801D   ..NOCK: LDA     KEYTRK+SK1RAM
  1817    CB47              BIT     SK1BIT,A
  1819    200C              JRNZ    ..SKDN
  181B    3A 8024           LDA     KEYTRK+SK2RAM
  181E    CB57              BIT     SK2BIT,A
  1820    2005              JRNZ    ..SKDN
  1822    7E                MOV     A,M                 ; IS SHIFT LOCKE
                    D?
  1823    E680              ANI     SLBITM
  1825    2805              JRZ     ..NOSK
                    ; YEP - USE SHIFT LOOKUP TABLE
  1827    21 18C3   ..SKDN: LXI     H,SKTBL
  182A    1803              JMPR    ..LOOK
  182C    21 1883   ..NOSK: LXI     H,NORTBL            ; ASSUME NOT
  182F    0600      ..LOOK: MVI     B,0                 ; DO TABLE LOOKU
                    P
  1831    09                DAD     B
  1832    7E                MOV     A,M                 ; GET ASCII
  1833    4F                MOV     C,A                 ; SAVE CHARACTER

  1834    A7                ANA     A                   ; VALID KEY?
  1835    C8                RZ                          ; ZERO MEANS NOT
                     SO
                    ; IS UPPER/LOWER ALPHA REVERSE WANTED?
  1836    3A 8025           LDA     KEYFLG
  1839    E640              ANI     MDBITM
  183B    79                MOV     A,C                 ; STAGE CHAR FOR
                     WHATEVER
  183C    2812              JRZ     ..NORM
                    ; REVERSE MODE IS SET/IS CHARACTER IN RANGE FOR
                    CONFUSION
  183E    FE41              CPI     'A'
  1840    380E              JRC     ..NORM              ; SKIP IF < UPPE
                    R A
  1842    FE7B              CPI     7BH                 ; OR IF ABOVE LO
                    WER Z
  1844    300A              JRNC    ..NORM
  1846    FE61              CPI     61H                 ; COOL IF >= LOW
                    ER A
  1848    3004              JRNC    ..BIZR
  184A    FE5B              CPI     5BH                 ; BAD IF ABOVE U
                    PPER Z
  184C    3002              JRNC    ..NORM
  184E    EE20      ..BIZR: XRI     20H        ; DO REVERSAL
  1850              ..NORM:
  1850    5F                MOV     E,A
                    ; PLACE CHAR INTO TYPEAHEAD BUFFER
  1851    2A 8028           LHLD    CONPRO              ; GET POINTERS
  1854    7D                MOV     A,L
  1855    CD 1865           CALL    BUMPTR              ; P=P+1
  1858    BC                CMP     H                   ; =C?
  1859    C8                RZ
  185A    32 8028           STA     PROPTR              ; NO - UPDATE P
```

```
 185D    2600                   MVI     H,0              ; STORE AT OLD P

 185F    01 802A                LXI     B,KEYBUF
 1862    09                     DAD     B
 1863    73                     MOV     M,E
 1864    C9                     RET
                        ; BUMP POINTER TO CIRCULAR BUFFER
 1865                   BUMPTR:
 1865    3C                     INR     A
 1866    FE20                   CPI     KEYBSZ
 1868    C0                     RNZ
 1869    AF                     XRA     A
 186A    C9                     RET
                        ; OUTPUT TO PORT 98 KEYBOARD COL SELECT MASK
 186B    C5             OUT98:  PUSH    B
 186C    01 0098                LXI     B,98H
 186F    ED79                   OUTP    A
 1871    C1                     POP     B
 1872    C9                     RET
                        ; INPUT FROM PORT 98 KEYBOARD DATA BITSTRING
 1873                   IN98:
 1873    C5                     PUSH    B
 1874    01 0098                LXI     B,98H
 1877    ED78                   INP     A
 1879    C1                     POP     B
 187A    C9                     RET
                        ; *****
                        ; *
                        ; * KEYBOARD SCANNER TABLES
                        ; *
                        ; *****
                        ; TABLE OF LIVE KEYS - ORDERED AS THE SCAN MATRI
                        X IS
 187B                   KEYMES:
 187B    FA                     .BYTE   11111010B
 187C    DF                     .BYTE   11011111B
 187D    FF                     .BYTE   11111111B
 187E    F7                     .BYTE   11110111B
 187F    FF                     .BYTE   11111111B
 1880    FF                     .BYTE   11111111B
 1881    FF                     .BYTE   11111111B
 1882    E0                     .BYTE   11100000B
                        ; CHARACTER LOOKUP TABLES
                        ;
                        ; UNSHIFTED CHARACTERS
                        ; ORDERED BY ROW, THEN COLUMN STARTING WITH BIT
                        ZERO
 1883                   NORTBL:
 1883    000000091B51           .BYTE   0,0,0,09H,1BH,'Q','[','1'
 188B    204247465900           .BYTE   ' ','B','G','F','Y',0,'6','7'
 1893    564344535254           .BYTE   'V','C','D','S','R','T','4','5'
 189B    585A41005745           .BYTE   'X','Z','A',0,'W','E','2','3'
 18A3    4E4D4A484955           .BYTE   'N','M','J','H','I','U','8','9'
 18AB    2C2E4C4B504F           .BYTE   ',','.','L','K','P','O','0','-'
 18B3    2F007C3B003A           .BYTE   '/',0,7CH,';',0,':',']','*'
```

```
18BB    00000000005C            .BYTE   0,0,0,0,0,5CH,RUBKEY,0DH
                        ; SHIFTED CHARACTERS
18C3                    SKTBL:
18C3    000000091B71            .BYTE   0,0,0,09H,1BH,71H,'[','!'
18CB    206267667900            .BYTE   ' ',62H,67H,66H,79H,0,'&',27H
18D3    766364737274            .BYTE   76H,63H,64H,73H,72H,74H,'$','%'
18DB    787A61007765            .BYTE   78H,7AH,61H,0,77H,65H,22H,'#'
18E3    6E6D6A686975            .BYTE   6EH,6DH,6AH,68H,69H,75H,'(',')'
18EB    3C3E6C6B706F            .BYTE   '<','>',6CH,6BH,70H,6FH,0,'='
18F3    3F005F2B0040            .BYTE   '?',0,5FH,'+',0,'@',']',85H
18FB    000000000000            .BYTE   0,0,0,0,0,0,LINKIL,0DH
                        ; CONTROL CHARACTERS
1903                    CKTBL:
1903    000000091B11            .BYTE   0,0,0,09H,1BH,11H,0,0
190B    200207061900            .BYTE   ' ',2,7,6,19H,0,0,0
1913    160304131214            .BYTE   16H,3,4,13H,12H,14H,0,0
191B    181A01001705            .BYTE   18H,1AH,1,0,17H,5,0,0
1923    0E0D0A080915            .BYTE   0EH,0DH,0AH,8,9,15H,0,0
192B    00000C0B100F            .BYTE   0,0,0CH,0BH,10H,0FH,0,0
1933    008182008000            .BYTE   0,81H,82H,0,80H,0,1DH,84H
193B    000000000083            .BYTE   0,0,0,0,0,83H,86H,0DH
                        ; *****
                        ; *
                        ; * VERB TO GRAB A CHARACTER FROM KEYBOARD
                        ; *  KIN character .
                        ; *
                        ; *****
                                VERB    "KIN"[
1943    177C           +        .WORD   .LINK.
1945    03             +        .BYTE   ..0033-.-1
1946    4B494E         +        .ASCII  "KIN"
1949                   +..0033:]
1949                    GETKEY:
1949    CD 1954         ..WAIT: CALL    KEYCHK
194C    28FB                    JRZ     ..WAIT
194E    6F                      MOV     L,A
194F    2600                    MVI     H,0
1951    E5                      PUSH    H
                                NEXT[
1952    FDE9           +        PCIY]
                        ; *****
                        ; *
                        ; * SUBROUTINE TO GET A CHARACTER FROM KEYBOARD
                        ; * RETURNS CHAR IN A AND NZ STATUS
                        ; * ELSE RETURNS Z SET MEANING NO CHAR READY
                        ; *
                        ; *****
1954    2A 8028         KEYCHK: LHLD    CONPRO
1957    7C                      MOV     A,H
1958    BD                      CMP     L       ; ARE WE ALL CAUGHT UP?
1959    C8                      RZ
195A    5C                      MOV     E,H
195B    1600                    MVI     D,0
195D    21 802A                 LXI     H,KEYBUF
1960    19                      DAD     D
```

```
1961    56                      MOV     D,M
1962    7B                      MOV     A,E
1963    CD 1865                 CALL    BUMPTR
1966    32 8029                 STA     CONPTR
1969    3E01                    MVI     A,1
196B    A7                      ANA     A
196C    7A                      MOV     A,D
196D    C9                      RET
                        ; *****
                        ; *
                        ; * 5 X 7 CHAR FONT
                        ; *
                        ; *****
196E                    FNT57:
196E    000000000000    .BYTE 000H,000H,000H,000H,000H,000H,000H ;
1975    202020202000    .BYTE 020H,020H,020H,020H,020H,000H,020H ; !
197C    505050000000    .BYTE 050H,050H,050H,000H,000H,000H,000H ; "
1983    4848FC48FC48    .BYTE 048H,048H,0FCH,048H,0FCH,048H,048H ; #
198A    2078807008F0    .BYTE 020H,078H,080H,070H,008H,0F0H,020H ; $
1991    C0C810204098    .BYTE 0C0H,0C8H,010H,020H,040H,098H,018H ; %
1998    6090A040A890    .BYTE 060H,090H,0A0H,040H,0A8H,090H,068H ; &
199F    606060000000    .BYTE 060H,060H,060H,000H,000H,000H,000H ; '
19A6    102020202020    .BYTE 010H,020H,020H,020H,020H,020H,010H ; (
19AD    402020202020    .BYTE 040H,020H,020H,020H,020H,020H,040H ; )
19B4    00A870D870A8    .BYTE 000H,0A8H,070H,0D8H,070H,0A8H,000H ; *
19BB    002020F82020    .BYTE 000H,020H,020H,0F8H,020H,020H,000H ; +
19C2    000000606020    .BYTE 000H,000H,000H,060H,060H,020H,040H ; ,
19C9    000000F80000    .BYTE 000H,000H,000H,0F8H,000H,000H,000H ; -
19D0    000000000060    .BYTE 000H,000H,000H,000H,000H,060H,060H ; .
19D7    000810204080    .BYTE 000H,008H,010H,020H,040H,080H,000H ; /
19DE    708888888888    .BYTE 070H,088H,088H,088H,088H,088H,070H ; 0
19E5    206020202020    .BYTE 020H,060H,020H,020H,020H,020H,070H ; 1
19EC    708808708080    .BYTE 070H,088H,008H,070H,080H,080H,0F8H ; 2
19F3    708808300888    .BYTE 070H,088H,008H,030H,008H,088H,070H ; 3
19FA    10305090F810    .BYTE 010H,030H,050H,090H,0F8H,010H,010H ; 4
1A01    F880F0080888    .BYTE 0F8H,080H,0F0H,008H,008H,088H,070H ; 5
1A08    304080F08888    .BYTE 030H,040H,080H,0F0H,088H,088H,070H ; 6
1A0F    F80810204040    .BYTE 0F8H,008H,010H,020H,040H,040H,040H ; 7
1A16    708888708888    .BYTE 070H,088H,088H,070H,088H,088H,070H ; 8
1A1D    708888780810    .BYTE 070H,088H,088H,078H,008H,010H,060H ; 9
1A24    006060006060    .BYTE 000H,060H,060H,000H,060H,060H,000H ; :
1A2B    606000606020    .BYTE 060H,060H,000H,060H,060H,020H,040H ; ;
1A32    102040804020    .BYTE 010H,020H,040H,080H,040H,020H,010H ; <
1A39    0000F800F800    .BYTE 000H,000H,0F8H,000H,0F8H,000H,000H ; =
1A40    402010081020    .BYTE 040H,020H,010H,008H,010H,020H,040H ; >
1A47    708808102000    .BYTE 070H,088H,008H,010H,020H,000H,020H ; ?
1A4E    7088B8A8B880    .BYTE 070H,088H,0B8H,0A8H,0B8H,080H,078H ; @
1A55    708888F88888    .BYTE 070H,088H,088H,0F8H,088H,088H,088H ; A
1A5C    F08888F08888    .BYTE 0F0H,088H,088H,0F0H,088H,088H,0F0H ; B
1A63    708880808088    .BYTE 070H,088H,080H,080H,080H,088H,070H ; C
1A6A    F08888888888    .BYTE 0F0H,088H,088H,088H,088H,088H,0F0H ; D
1A71    F88080E08080    .BYTE 0F8H,080H,080H,0E0H,080H,080H,0F8H ; E
1A78    F88080E08080    .BYTE 0F8H,080H,080H,0E0H,080H,080H,080H ; F
1A7F    708880809888    .BYTE 070H,088H,080H,080H,098H,088H,078H ; G
1A86    888888F88888    .BYTE 088H,088H,088H,0F8H,088H,088H,088H ; H
```

```
  1A8D      702020202020  .BYTE 070H,020H,020H,020H,020H,020H,070H ; I
  1A94      080808080888  .BYTE 008H,008H,008H,008H,008H,088H,070H ; J
  1A9B      8890A0C0A090  .BYTE 088H,090H,0A0H,0C0H,0A0H,090H,088H ; K
  1AA2      808080808080  .BYTE 080H,080H,080H,080H,080H,080H,0F8H ; L
  1AA9      88D8A8A88888  .BYTE 088H,0D8H,0A8H,0A8H,088H,088H,088H ; M
  1AB0      88C8A8988888  .BYTE 088H,0C8H,0A8H,098H,088H,088H,088H ; N
  1AB7      F88888888888  .BYTE 0F8H,088H,088H,088H,088H,088H,0F8H ; O
  1ABE      F08888F08080  .BYTE 0F0H,088H,088H,0F0H,080H,080H,080H ; P
  1AC5      70888888A890  .BYTE 070H,088H,088H,088H,0A8H,090H,068H ; Q
  1ACC      F08888F0A090  .BYTE 0F0H,088H,088H,0F0H,0A0H,090H,088H ; R
  1AD3      708880700888  .BYTE 070H,088H,080H,070H,008H,088H,070H ; S
  1ADA      F82020202020  .BYTE 0F8H,020H,020H,020H,020H,020H,020H ; T
  1AE1      888888888888  .BYTE 088H,088H,088H,088H,088H,088H,070H ; U
  1AE8      888888505020  .BYTE 088H,088H,088H,050H,050H,020H,020H ; V
  1AEF      888888A8A8D8  .BYTE 088H,088H,088H,0A8H,0A8H,0D8H,088H ; W
  1AF6      888850205088  .BYTE 088H,088H,050H,020H,050H,088H,088H ; X
  1AFD      888850202020  .BYTE 088H,088H,050H,020H,020H,020H,020H ; Y
  1B04      F80810204080  .BYTE 0F8H,008H,010H,020H,040H,080H,0F8H ; Z
  1B0B      704040404040  .BYTE 070H,040H,040H,040H,040H,040H,070H ;
  1B12      008040201008  .BYTE 000H,080H,040H,020H,010H,008H,000H ; \
  1B19      701010101010  .BYTE 070H,010H,010H,010H,010H,010H,070H ;
  1B20      2070A8202020  .BYTE 020H,070H,0A8H,020H,020H,020H,020H ; UP
  1B27      002040F84020  .BYTE 000H,020H,040H,0F8H,040H,020H,000H ; LEFT

  1B2E      20202020A870  .BYTE 020H,020H,020H,020H,0A8H,070H,020H ; DOWN

  1B35      002010F81020  .BYTE 000H,020H,010H,0F8H,010H,020H,000H ; RIGH
                          T
                          ; *****
                          ; *
                          ; * 3 X 5 CHAR FONT
                          ; *
                          ; *****
  1B3C                    FNT35:
  1B3C      000000000040  .BYTE 000H,000H,000H,000H,000H,040H,040H,040H,0
                          00H,040H,0A0H,0A0H
  1B48      000000A0E0A0  .BYTE 000H,000H,000H,0A0H,0E0H,0A0H,0E0H,0A0H,0
                          40H,0E0H,080H,0E0H
  1B54      408020408020  .BYTE 040H,080H,020H,040H,080H,020H,000H,000H,0
                          40H,0A0H,0A0H,040H
  1B60      400000004080  .BYTE 040H,000H,000H,000H,040H,080H,080H,080H,0
                          40H,040H,020H,020H
  1B6C      204000A040A0  .BYTE 020H,040H,000H,0A0H,040H,0A0H,000H,000H,0
                          40H,0E0H,040H,000H
  1B78      000000408000  .BYTE 000H,000H,000H,040H,080H,000H,000H,0E0H,0
                          00H,000H,000H,000H
  1B84      000040002040  .BYTE 000H,000H,040H,000H,020H,040H,080H,000H,0
                          40H,0A0H,0A0H,0A0H
  1B90      404040404040  .BYTE 040H,040H,040H,040H,040H,040H,0E0H,020H,0
                          E0H,080H,0E0H,0E0H
  1B9C      206020E0A0A0  .BYTE 020H,060H,020H,0E0H,0A0H,0A0H,0E0H,020H,0
                          20H,0E0H,080H,0C0H
  1BA8      20C0E080E0A0  .BYTE 020H,0C0H,0E0H,080H,0E0H,0A0H,0E0H,0E0H,0
                          20H,040H,040H,040H
  1BB4      E0A0E0A0E0E0  .BYTE 0E0H,0A0H,0E0H,0A0H,0E0H,0E0H,0A0H,0E0H,0
```

```
                         20H,0E0H,000H,040H
  1BC0    004000004000  .BYTE 000H,040H,000H,000H,040H,000H,040H,080H,0
                         20H,040H,080H,040H
  1BCC    2000E000E000  .BYTE 020H,000H,0E0H,000H,0E0H,000H,080H,040H,0
                         20H,040H,080H,0E0H
  1BD8    20400040E0A0  .BYTE 020H,040H,000H,040H,0E0H,0A0H,0E0H,080H,0
                         C0H,0E0H,0A0H,0E0H
  1BE4    A0A0E0A0C0A0  .BYTE 0A0H,0A0H,0E0H,0A0H,0C0H,0A0H,0E0H,0E0H,0
                         80H,080H,080H,0E0H
  1BF0    C0A0A0A0C0E0  .BYTE 0C0H,0A0H,0A0H,0A0H,0C0H,0E0H,080H,0C0H,0
                         80H,0E0H,0E0H,080H
  1BFC    C08080E080A0  .BYTE 0C0H,080H,080H,0E0H,080H,0A0H,0A0H,0E0H,0
                         A0H,0A0H,0E0H,0A0H
  1C08    A0E0404040E0  .BYTE 0A0H,0E0H,040H,040H,040H,0E0H,020H,020H,0
                         20H,0A0H,0E0H,0A0H
  1C14    A0C0A0A08080  .BYTE 0A0H,0C0H,0A0H,0A0H,080H,080H,080H,080H,0
                         E0H,0A0H,0E0H,0E0H
  1C20    A0A0C0A0A0A0  .BYTE 0A0H,0A0H,0C0H,0A0H,0A0H,0A0H,0A0H,0E0H,0
                         A0H,0A0H,0A0H,0E0H
  1C2C    E0A0E08080E0  .BYTE 0E0H,0A0H,0E0H,080H,080H,0E0H,0A0H,0A0H,0
                         E0H,020H,0C0H,0A0H
  1C38    C0A0A0E080E0  .BYTE 0C0H,0A0H,0A0H,0E0H,080H,0E0H,020H,0E0H,0
                         E0H,040H,040H,040H
  1C44    40A0A0A0A0E0  .BYTE 040H,0A0H,0A0H,0A0H,0A0H,0E0H,0A0H,0A0H,0
                         A0H,0A0H,040H,0A0H
  1C50    A0E0E0A0A0A0  .BYTE 0A0H,0E0H,0E0H,0A0H,0A0H,0A0H,040H,0A0H,0
                         A0H,0A0H,0A0H,040H
  1C5C    4040E0204080  .BYTE 040H,040H,0E0H,020H,040H,080H,0E0H,0C0H,0
                         80H,080H,080H,0C0H
  1C68    008040200060  .BYTE 000H,080H,040H,020H,000H,060H,020H,020H,0
                         20H,060H,040H,0E0H
  1C74    4040402040E0  .BYTE 040H,040H,040H,020H,040H,0E0H,040H,020H
                         ; DEFAULT WINDOW DESCRIPTOR
  1C7C                   DEFWIN:
  1C7C    0050                   .WORD   80
  1C7E    FFB1                   .WORD   -79
  1C80    0033                   .WORD   51
  1C82    FFCE                   .WORD   -50
                         ; VERBS FOR "I","J", AND "K"
                                 VERB    "I"[
  1C84    1943         +         .WORD   .LINK.
  1C86    01           +         .BYTE   ..0034-.-1
  1C87    49           +         .ASCII  "I"
  1C88                 +..0034:]
  1C88    DD6E00                 MOV     L,0(X)
  1C8B    DD6601                 MOV     H,1(X)
  1C8E    E5                     PUSH    H
                                 NEXT[
  1C8F    FDE9         +         PCIY]
                                 VERB    "J"[
  1C91    1C84         +         .WORD   .LINK.
  1C93    01           +         .BYTE   ..0035-.-1
  1C94    4A           +         .ASCII  "J"
  1C95                 +..0035:]
  1C95    DD6E06                 MOV     L,6(X)
```

```
    1C98    DD6607                  MOV     H,7(X)
    1C9B    E5                      PUSH    H
                                    NEXT[
    1C9C    FDE9          +         PCIY]
                                    VERB    "K"[
    1C9E    1C91          +         .WORD   .LINK.
    1CA0    01            +         .BYTE   ..0036-.-1
    1CA1    4B            +         .ASCII  "K"
    1CA2                  +..0036:]
    1CA2    DD6E0C                  MOV     L,12(X)
    1CA5    DD660D                  MOV     H,13(X)
    1CA8    E5                      PUSH    H
                                    NEXT[
    1CA9    FDE9          +         PCIY]
                           ; CHARACTER WINDOW VARIABLE NAMES
                                    CONSTANT[WINDOW,WINPTR][
    1CAB    1C9E          +         .WORD   .LINK.
    1CAD    06            +         .BYTE   ..0037-.-1
    1CAE    57494E444F57+           .ASCII  "WINDOW"
    1CB4    CD 0CEB       +:]       CALL    CONST
    1CB7    8000          +         .WORD   WINPTR]
                                    CONSTANT[CDX,CDXCEL][
    1CB9    1CAB          +         .WORD   .LINK.
    1CBB    03            +         .BYTE   ..0038-.-1
    1CBC    434458        +         .ASCII  "CDX"
    1CBF    CD 0CEB       +:]       CALL    CONST
    1CC2    8002          +         .WORD   CDXCEL]
                                    CONSTANT[CDY,CDYCEL][
    1CC4    1CB9          +         .WORD   .LINK.
    1CC6    03            +         .BYTE   ..0039-.-1
    1CC7    434459        +         .ASCII  "CDY"
    1CCA    CD 0CEB       +:]       CALL    CONST
    1CCD    8004          +         .WORD   CDYCEL]
                                    CONSTANT[CDC,CDCCEL][
    1CCF    1CC4          +         .WORD   .LINK.
    1CD1    03            +         .BYTE   ..0040-.-1
    1CD2    434443        +         .ASCII  "CDC"
    1CD5    CD 0CEB       +:]       CALL    CONST
    1CD8    8006          +         .WORD   CDCCEL]
                                    CONSTANT[CDFONT,CDFCEL][
    1CDA    1CCF          +         .WORD   .LINK.
    1CDC    06            +         .BYTE   ..0041-.-1
    1CDD    4344464F4E54+           .ASCII  "CDFONT"
    1CE3    CD 0CEB       +:]       CALL    CONST
    1CE6    8008          +         .WORD   CDFCEL]
                                    CONSTANT[CDWIND,CDWCEL][
    1CE8    1CDA          +         .WORD   .LINK.
    1CEA    06            +         .BYTE   ..0042-.-1
    1CEB    434457494E44+           .ASCII  "CDWIND"
    1CF1    CD 0CEB       +:]       CALL    CONST
    1CF4    800A          +         .WORD   CDWCEL]
                           ; TABLE OF INITIAL VALUES FOR INITIALIZED AREA
    1CF6                   INIVAL:
    1CF6    1C7C                    .WORD   DEFWIN  ; WINPTR TO DEFAULT WIND
                           OW
```

```
 1CF8   FFB1                    .WORD   -79     ; CX,CY TO SCREEN ULHC
 1CFA   0033                    .WORD   51
 1CFC   0001                    .WORD   1       ; COLOR OF CHARS TYPED
 1CFE   14CC                    .WORD   SMLFNT  ; FONT TO TYPE CHARS WIT
                        H
 1D00   1C7C                    .WORD   DEFWIN  ; WINDOW TO TYPE CHARS I
                        NTO
 1D02                   ENDGR:
                        ; *****
                        ; *
                        ; * RAM STUFF FOLLOWS!
                        ; *
                        ; *****
 8000                           .LOC    RAMBEG
                        ; THIS AREA INITIALIZED ON POWERUP TO REASONABLE
                         STUFF
 8000                   WINPTR: .BLKB   2
 8002                   CDXCEL: .BLKB   2
 8004                   CDYCEL: .BLKB   2
 8006                   CDCCEL: .BLKB   2
 8008                   CDFCEL: .BLKB   2
 800A                   CDWCEL: .BLKB   2
                        ; END OF INITIALIZED AREA
 800C                   ZEROUT:
 800C                   INTLOK: .BLKB   1       ; INTERRUPT LOCKOUT FLAG

 800D                   INCRO:  .BLKB   2       ; GRAPHICS SCRATCH MEMOR
                        Y
 800F                   MNMX:   .BLKB   2
 8011                   PIXVAL: .BLKB   1
 8012                   WRMODE: .BLKB   1
                        ; RAM FOR CIRCLE COMMAND
 8013                   CIRXA:  .BLKB   2
 8015                   CIRYA:  .BLKB   2
 8017                   CIRDEL: .BLKB   2
 8019                   INTVRB: .BLKB   2       ; INTERRUPT VERB TO DO
 801B                   IYVALU: .BLKB   2 ; IY VALUE TO USE FOR INTERRUP
                        T ( ** TEMP **)
                        ; RAM FOR KEYBOARD SCANNER
 801D                   KEYTRK: .BLKB   8       ; KEYBOARD TRACKING MEM
 8025                   KEYFLG: .BLKB   1       ; KEYBOARD SHIFT MODE FL
                        AG
 8026                   OLDKEY: .BLKB   1       ; LAST KEYCODE TYPED
 8027                   KEYPTK: .BLKB   1       ; EASY ENTRY KEYPAD TRAC
                        KER
 8028                   CONPRO:
 8028                   PROPTR: .BLKB   1       ; PRODUCER POINTER
 8029                   CONPTR: .BLKB   1       ; CONSUMER POINTER
 802A                   KEYBUF: .BLKB   KEYBSZ  ; KEYBOARD BUFFER
 003E                   ZERSIZ=.-ZEROUT
 1CE8                   LSTLNK=.LINK.
                                .END
```

```
TDL Z80 CP/M DISK ASSEMBLER VERSION 2.21                            PAGE 56

+++++ SYMBOL TABLE +++++


ABS    1471          BREAK  00FF          BUMPCX 1739
BUMPCY 174C          BUMPTR 1865          BX.MOD 0002
BX.X   000A          BX.XS  0006          BX.Y   0008
BX.YS  0004          BYTEPL 0028          CBOTH  0FBE
CDC    1CD5          CDCCEL 8006          CDELTA 1448
CDFCEL 8008          CDFONT 1CE3          CDWCEL 800A
CDWIND 1CF1          CDX    1CBF          CDXCEL 8002
CDY    1CCA          CDYCEL 8004          CF.C   0006
CF.F   0004          CF.M   0008          CF.X   000C
CF.Y   000A          CHARO  1532          CIRDEL 8017
CIRPNT 1027          CIRXA  8013          CIRYA  8015
CK1BIT 0003          CK1RAM 0003          CK2BIT 0004
CK2RAM 0007          CKTBL  1903          CLEARE 139F
CLIP   12C5          CLIPPE 12DF          CLOOP  0F82
CLP.C  000A          CLP.S  0006          CMPM   14B4
CNTMSK 0030          COMLV  1771          COMUV  1778
CONPRO 8028          CONPTR 8029          CONST  0CEB
CPHLDE 1386          CRAPPE 4FFF          CYSCRO 168B
DCHAR  158A          DECLOK 0CF9          DEFWIN 1C7C
DEPARM 1761          DIV2HL 1050          DMINUS 0FA6
DOBOX  120B          DOWNA  0000          DPADDR 8060
DPVAL  8080          DQUOTE 0022          DRAW4  1001
DSTOR  0FD7          DVECT  13F6          ENDGR  1D02
ESC    001B          ESCKEY 0004          FDF.XL 0000
FD.AD  0005          FD.BAS 0000          FD.FLG 0007
FD.XCS 0001          FD.XF  0003          FD.YCS 0002
FD.YF  0004          FINDLA 165B          FLIP   00FE
FNT35  1B3C          FNT57  196E          FORWA  0000
FR.P1  0002          FR.P2  0004          FR.P3  0006
FR.P4  0008          FR.P5  000A          FR.P6  000C
FR.P7  000E          FR.P8  0010          GETKEY 1949
GOBACK 0D47          GOBAKV 0D45          GPIXEL 0EC1
GRADDR 0CA0          IN98   1873          INBIT  0DCD
INCHAR 0DB9          INCLOK 0CF2          INCRO  800D
INFBK  000D          INICOL 0CE0          INIVAL 1CF6
INLIN  000F          INMOD  000E          INTLOK 800C
INTNOG 0D52          INTVEC 0D18          INTVRB 8019
IYVALU 801B          KEYBSZ 0020          KEYBUF 802A
KEYCHK 1954          KEYFLG 8025          KEYMES 187B
KEYPSN 178E          KEYPTK 8027          KEYSCN 17B5
KEYTRK 801D          KEYTRV 0003          LARGE  14DC
LASTAD 8064          LEADER 0E5A          LEFTA  005F
LEFTX  1633          LF     000A          LINKIL 0018
LOCK   00FD          LRGFNT 14E1          LSTLNK 1CE8
MAGIC  000C          MDBITM 0040          MNMX   800F
MRXPND 0003          MSKTBL 13F2          NEGHL  1058
NL     000D          NORTBL 1883          OLDKEY 8026
ORJOIN 0F07          ORPT   0F01          OUT98  186B
OUTBYT 0E2F          PIXVAL 8011          PLOP   0F12
PLOP1  0F13          PLOPNG 0F3A          POINTR 0EEB
POWERU 0CA0          PRIOR  0F3C          PROPTR 8028
PRPLOP 0F0E          R2A    1491          R2ACLP 1476
RAMBEG 8000          RESCX  1675          RESCY  16DC
RUBKEY 005F          RUBOUT 0008          SAVEE  0E10
SCRINT 0D1A          SCROLE 1069          SGNEXT 146A
```

```
+++++ SYMBOL TABLE +++++


SHKMSK 0044         SHYLOK 0001         SK1BIT 0000
SK1RAM 0000         SK2BIT 0002         SK2RAM 0007
SKTBL  18C3         SLBITM 0080         SMALL  14C7
SMLFNT 14CC         SQUOTE 0027         TAB    0009
TAPEIO 0099         TAPGET 0D97         TOKBIT 0002
TOKRAM 0000         UPA    005E         UPY    1648
URINAL 0FFF         VECT2  141C         VECT2A 1423
VECT3  1433         VECT4  1438         VECT5  1443
WINDOW 1CB4         WINPTR 8000         WRBLOC 0E24
WRMODE 8012         WRONE  0E70         WRZERO 0E7A
WXL    0002         WXR    0000         WYL    0006
WYU    0004         XCHECK 170C         XORCHR 0000
XORPT  0EF9         XORWMR 0020         XPAND  0019
XPWMR  0008         YCHECK 16F4         ZEROUT 800C
ZERSIZ 003E         .LINK. 1CE8
```