UA6 & UA7 — FCS & CRT ROM ONLY

CONTAINS LISTING FOR —

211C – 2579	1. DISK HANDLER V. 6.78 (3 PHASE)	P.2
257A – 338A	2. F.C.S. V. 3.00	P.13
338B – 35A5	3. UTILITY SUB ROUTINE V. 4.0	P.57
35A5 – 368A	4. BLOCK STRUCTURE HANDLER V. 1.0	P.64
368B – 3FFF	5. CCII SOFTWARE V. 6.78	P.67

CONTAINS CRT HEADER 3766 – 25EB

CONTAINS BASIC HEADER 189C – 18EF

0000 – 0040 PNEM TO start of system software V *

0041 – 211B 16 BASIC ROM ([redacted] TAYLOR LIST) UA4 & UA5

SYSTEM SOFTWARE

for

COMPUCOLOR II V6.78

P.25 V3.H1 COLORCUE

MEM LOCATIONS 0001 – 0002 The # 376C(H) identifies V. 6.78

0048 – 211B
3E7A – 3FFF
} ?? [redacted]

DISK BASIC HEADER 189C – 18EF — MODIFIED TO SMALL CHAR & MY NAME 10/83

BASIC HEADER in UA5
CRT HEADER in UA6
ESCAPES in UA7
60HZ → 50HZ in UA7

999211

05/18/78

Intelligent Systems Corp.
Intecolor Drive
225 Technology Park/Atlanta
Norcross, Georgia, 30092
Telephone: 404/449-5961
TWX: 810/766-1581

```
FCS>ASM SYS678.SRC;01

                ;       SYSTEM SOFTWARE FOR COMPUCOLOR II
                ;       MAY 18, 1978 - VERSION 6.78
                ;       COPYRIGHT (C) 1978 BY COMPUCOLOR CORP.

      (211C)    SYSORG  EQU     211CH   ; ROM ORG FOR SYSTEM

      (1000)    DSBUFS  EQU     32*64*2 ; NOLIN*NOCHAR*2
      (6000)    DSBUF   EQU     6000H   ; START ADDRESS OF DISPLAY FAST RAM
      (7000)    X80     EQU     DSBUF+DSBUFS ; START OF DISPLAY SLOW RAM

      (8042)    STACK   EQU     8042H   ; STACK FROM SCREEN TO HERE
      (8046)    LINBF   EQU     8046H   ; BASIC LINE BUFFER
      (80DE)    TRAM    EQU     80DEH   ; TEMP RAM START IN BASIC RAM
      (80F0)    ORAM    EQU     80F0H   ; FCS RAM
      (81AF)    CRTRAM  EQU     81AFH   ; CRT RAM

A120  (6000)            ORG     6000H   ; BASIC ADDRESSES
6000  5334              DW      CMPDH
6002  5934              DW      SUBHD
6004  3B34              DW      MOVDH
6006  4434              DW      MOVHD
6008  4D34              DW      CMPHD
600A  1835              DW      ADHLA
600C  1D35              DW      ANHD
600E  2435              DW      NEGH
6010  2535              DW      NOTH
6012  2C35              DW      ORHD
6014  3335              DW      XORHD
6016  3A35              DW      SHLHD
6018  4435              DW      SHRHD
601A  6235              DW      MULHD
601C  8135              DW      DIVHD
601E  6034              DW      SPNOR
6020  2D26              DW      EMEGS
6022  4F28              DW      WF2
6024  8328              DW      RF2
6026  FC2E              DW      CHDLR
6028  7730              DW      PFSPC
602A  862C              DW      OPENX
602C  A526              DW      RESET
602E  F82E              DW      WR
6030  FB2E              DW      RD
6032  5928              DW      CLX

6034  (8042)            ORG     STACK
8042  (0001)    SBC:    DS      1       ; SECTOR BYTE COUNT FOR DISK
8043  (0001)    CRC1:   DS      1       ; 1ST CRC BYTE FOR DISK
8044  (0001)    CRC2:   DS      1       ; 2ND CRC BYTE FOR DISK

8045  (8046)            ORG     LINBF
```

```
8046 (0001)             DS      1       ; FOR BASIC'S ","
8047 (003B)   BUFP:     DS      59      ; FCS LINE BUFFER
8082 (0027)   ZRAM:     DS      39      ; FCS STUFF / BASIC INPUT BUFFER
80A9 (80A9)             ORG     $       ; *** SHOULD EQUAL "X20A" IN BASIC ***


              ; *** COMPUCOLOR DISK HANDLER ***
              ; *** VERSION FOR USE WITH COMPUCOLOR II ***
              ;
              ; COPYRIGHT (C) 1978
              ; BY INTELLIGENT SYSTEMS CORPORATION
              ;

              ; CONSTANTS :

    (000A)    NSEC    EQU     10      ; NUMBER OF SECTORS / TRACK
    (0029)    NTRK    EQU     41      ; NUMBER OF TRACKS
    (0014)    STPTIM  EQU     20      ; STEP TIME / 0.5 MS.
    (0640)    UPTIM   EQU     1600    ; MOTOR UP-TO-SPEED TIME / 0.5 MS.
    (00FF)    FILL    EQU     0FFH    ; "FILL" BYTE
    (0055)    IDM     EQU     55H     ; ID MARK
    (005A)    DATAM   EQU     5AH     ; DATA MARK

    (0003)    RGAPS   EQU     3       ; BYTE TIMES / GAP CONFIRMATION

    (0004)    WRTR    EQU     4       ; WRITE/VERIFY RETRY COUNT
    (0002)    VRTR    EQU     2       ; VERIFY RETRY COUNT
    (000A)    RRTR    EQU     10      ; READ RETRY COUNT
    (001E)    SRTR    EQU     3*NSEC  ; SEARCH TRY COUNT
    (001E)    HRTR    EQU     3*NSEC  ; HEADER RETRY COUNT

              ; "TEMP" RAM ALLOCATION :

80A9 (80DE)             ORG     TRAM

80DE (0002)   TEMPHL:   DS      2       ;TEMP REG FOR HL
80E0 (0001)   BRTRY:    DS      1       ; BLOCK RETRY COUNTER
80E1 (0001)   RFLG:     DS      1       ; "RESTORE" FLAG / COUNTER
80E2 (0001)   CRTRY:    DS      1       ; "CHUNK" RETRY COUNTER
80E3 (0002)   OBC:      DS      2       ; OLD BYTE COUNT

80E5 (0001)   TFCN:     DS      1       ; FUNCTION CODE
80E6 (0001)   TDRV:     DS      1       ; DRIVE NUMBER
80E7 (0002)   TBLK:     DS      2       ; BLOCK NUMBER
80E9 (0002)   TMEM:     DS      2       ; MEMORY BUFFER POINTER
80EB (0002)   TBC:      DS      2       ; BYTE COUNT

80ED (0001)   OSEC:     DS      1       ; OLD SECTOR NUMBER
80EE (0001)   SEC:      DS      1       ; SECTOR NUMBER
80EF (0001)   TRK:      DS      1       ; TRACK NUMBER
80F0 (80F0)             ORG     $       ; *** SHOULD NOT BE PAST "ORAM" ***


              ; COMPUCOLOR DISK HANDLER :
```

* UA6
STARTS AT
2000(H)

```
80F0 (211C)            ORG     SYSOP?

211C EB        CDHD:   XCHG
211D 2A9136            LHLD    CDNM    ; GET NAME
2120 EB                XCHG
2121 3A9336            LDA     CDNU    ; GET NUMBER OF UNITS
2124 CDC735            CALL    BSBO1   ; COPY PARAMETERS & SETUP RETURN
2127 F5                PUSH    PSW     ; SAVE FLAGS
2128 210000            LXI     H,0
212B 54                MOV     D,H
212C 3A9436            LDA     CDSEC   ; GET NUMBER OF SECTORS
212F 5F                MOV     E,A     ; E=NUMBER OF SECTORS / TRACK
2130 3E28              MVI     A,NTRK-1 ; NO. OF TRACKS USED FOR DATA.
2132 19        DX10:   DAD     D       ; ADD SECTORS AGAIN
2133 3D                DCR     A       ; ENOUGH ?
2134 C23221            JNZ     DX10    ; NO: LOOP!
2137 F1                POP     PSW     ; RESTORE FLAGS
2138 F29A21            JP      XFER    ; IT IS READ OF WRITE!
213B 22E780            SHLD    TBLK    ; SAVE TOTAL NUMBER OF BLOCKS
213E 3C                INR     A       ; "GET SIZE" FUNCTION ?
213F C8                RZ              ; YES: EXIT!
2140 3C                INR     A       ; "ADD USER" FUNCTION ?
2141 C24921            JNZ     DX11    ; NO!
2144 CD7025    ADDU:   CALL    GCUCNT  ; HL => CUCNTO OR CUCNT1
2147 34                INR     M       ; INCREMENT USER COUNT
2148 C9                RET             ; EXIT

2149 3C        DX11:   INR     A       ; "SUBTRACT USER" FUNCTION ?
214A C28B21            JNZ     CDRSET  ; NO: MUST BE "TURN OFF" FUNCTION!
214D CD7025    SUBU:   CALL    GCUCNT  ; HL => CUCNTO OR CUCNT1
2150 35                DCR     M       ; SUBTRACT A USER
2151 F25921            JP      DX12    ; O.K. !
2154 CD7025    TOFF:   CALL    GCUCNT  ; HL => CUCNTO OR CUCNT1
2157 3600              MVI     M,0     ; SET ZERO USERS
2159 C5        DX12:   PUSH    B
215A 7E                MOV     A,M     ; GET USER COUNT
215B A7                ANA     A       ; TEST IT
215C C26921            JNZ     DX13    ; NOT ZERO !
215F CDC321            CALL    DX14A   ; WAIT FOR BUFFERS TO EMPTY
2162 CDDD24            CALL    VTP     ; ARE TRACK & PHASE VALID ?
2165 AF                XRA     A
2166 CD2525            CALL    STEPS   ; MOVE HEAD TO HOME
2169 AF        DX13:   XRA     A       ; NO REASON
216A D307              OUT     EXTOT   ; DESELECT DRIVE
216C 3E10              MVI     A,16    ; WAIT ONE CHARACTER TIME ...
216E 3D        DX13A:  DCR     A
216F C26E21            JNZ     DX13A
2172 F3                DI
2173 3E10              MVI     A,10H
2175 D308              OUT     MASK    ; MASK SERIAL IN ONLY
2177 DB02              IN      RSTBF   ; "CLEAR" RECVR INTERUPT
2179 3A1700            LDA     RCOMD   ; PICK UP HI/LO RATE COMND
217C D304              OUT     COMND   ; ISSUE
217E 3AE281            LDA     CRATE   ; PICK UP CURRENT RATE
2181 D305              OUT     BAUD    ; ISSUE
2183 3AE081            LDA     CMASK
```

```
2186 D308              OUT     MASK    ; RESTORE 5501 MASK
2188 C1                POP     B
2189 FB                EI
218A C9                RET             ; EXIT !!

218B AF        CDRSET: XRA     A
218C 32E680            STA     TDRV    ; SET DRIVE ZERO
218F CD5421            CALL    TOFF    ; PARK DRIVE ZERO
2192 3E01              MVI     A,1
2194 32E680            STA     TDRV    ; SET DRIVE ONE
2197 C35421            JMP     TOFF    ; PARK DRIVE ONE AND EXIT

219A 7B        XFER:   MOV     A,E     ; NO. OF SECTORS
219B 2F                CMA
219C 3C                INR     A
219D 4F                MOV     C,A     ; C= - NO. OF SECTORS
219E EB                XCHG            ; D&E=TOTAL NUMBER OF BLOCKS
219F 2AE780            LHLD    TBLK    ; GET BLOCK NUMBER
21A2 CD4D34            CALL    CMPHD   ; IS IT VALID ?
21A5 0609              MVI     B,EBLK  ; SET ERROR CODE
21A7 D0                RNC             ; ERROR: INVALID BLOCK!
21A8 06FF              MVI     B,-1    ; SET HI BYTE OF B&C
21AA AF                XRA     A       ; CLEAR A
21AB 3C        DX14:   INR     A       ; COUNT TRACK
21AC 09                DAD     B       ; SUBTRACT SECTORS / TRACK
21AD DAAB21            JC      DX14    ; LOOP!
21B0 67                MOV     H,A     ; MOVE TRACK NUMBER TO H
21B1 7D                MOV     A,L     ; GET LO BYTE
21B2 91                SUB     C       ; ADJUST
21B3 6F                MOV     L,A     ; SET SECTOR NUMBER
21B4 22EE80            SHLD    SEC     ; SAVE TRACK & SECTOR
21B7 CD4421            CALL    ADDU    ; "ADD A USER"
21BA CDC321            CALL    DX14A   ; WAIT TIL CHAR FINISHED
21BD CD0122            CALL    DXX     ; PERFORM READ OR WRITE
21C0 C34D21            JMP     SUBU    ; SUBTRACT USER AND EXIT

21C3 3A2E00    DX14A:  LDA     CDMK
21C6 D308              OUT     MASK    ; SET 5501 MASK FOR EXTERNAL SENSOR ONLY
21C8 3AE281            LDA     CRATE   ; GET 5501 CURRENT RATE
21CB 07                RLC             ; GET RID OF STOP BIT BIT
21CC 210100            LXI     H,1     ; INITIALIZE DELAY NEEDED
21CF 0604              MVI     B,4     ; INITIALIZE COUNTER
21D1 07        DX15:   RLC             ; GET NEXT BIT
21D2 DAE621            JC      DX16    ; FOUND THE RATE !
21D5 29                DAD     H       ; DOUBLE THE DELAY NEEDED
21D6 05                DCR     B       ; AGAIN ?
21D7 C2D121            JNZ     DX15    ; YES !
21DA 29                DAD     H       ; DOUBLE THE DELAY NEEDED
21DB 07                RLC             ; GET NEXT BIT
21DC DAE621            JC      DX16    ; FOUND THE RATE !
21DF 29                DAD     H       ; DOUBLE THE DELAY NEEDED
21E0 07                RLC             ; GET NEXT BIT
21E1 DAE621            JC      DX16    ; FOUND THE RATE !
21E4 2E60              MVI     L,96    ; IT MUST BE 110 BAUD: SET DELAY NEEDED

21E6           DX16:                   ; WAIT FOR BUFFERS TO EMPTY :

21E6 3A1700    DX17:   LDA     RCOMD   ; GET RATE COMMAND
```

```
21E9 E610              ANI     10H       ; MASK OFF HI-RATE BIT
21EB CAF021            JZ      DX18      ; LO-RATE: USE ZERO COUNTER !
21EE 3E20              MVI     A,32      ; HI-RATE: USE PROPER COUNTER
21F0 3D          DX18: DCR     A         ; ENOUGH ?
21F1 C2F021            JNZ     DX18      ; NO: LOOP !
21F4 2D                DCR     L         ; ENOUGH ?
21F5 C2E621            JNZ     DX17      ; NO: LOOP !

21F8 3E18              MVI     A,18H
21FA D304              OUT     COMND     ; SET HIGH SPEED
21FC 3EC0              MVI     A,0C0H
21FE D305              OUT     BAUD      ; SET RATE = 76.8 KBAUD
2200 C9                RET

2201 CDDD24      DXX:  CALL    VTP       ; POUND HOME IF NEEDED
2204 CD6A25            CALL    GCTRK     ; HL => CTRK0 OR CTRK1
2207 66                MOV     H,M       ; GET CURRENT TRACK NUMBER
2208 3A9536            LDA     CDK       ; GET UPTIM/STPTIM
220B 6F                MOV     L,A       ; SAVE IT
220C 3AEF80            LDA     TRK       ; GET DESIRED TRACK NUMBER
220F 94                SUB     H         ; COMPUTE POSITIVE DIFFERENCE ...
2210 F21522            JP      CD03
2213 2F                CMA
2214 3C                INR     A
2215 95          CD03: SUB     L         ; COMPUTE -(ADDITIONAL STEP DELAYS NEEDED)
2216 F22222            JP      SEEK      ; NO ADDITIONAL DELAY NEEDED !
2219 F5          CD04: PUSH    PSW       ; SAVE COUNT
221A CD5E25            CALL    STPWAT    ; DELAY ONE STEP TIME
221D F1                POP     PSW       ; RESTORE COUNT
221E 3C                INR     A         ; ENOUGH ?
221F C21922            JNZ     CD04      ; NO: WAIT AGAIN !

2222 FB          SEEK: EI                ; *** ENABLE INTERRUPTS ***
2223 3AEF80            LDA     TRK       ; GET DESIRED TRACK NUMBER
2226 CD2525            CALL    STEPS     ; MOVE HEAD TO DESIRED TRACK

2229 3AEE80            LDA     SEC       ; GET CURRENT SECTOR NUMBER
222C 32ED80            STA     OSEC      ; SAVE IT FOR VERIFY PASS
222F 2AEB80            LHLD    TBC       ; GET CURRENT BYTE COUNT
2232 22E380            SHLD    OBC       ; SAVE IT FOR VERIFY PASS

2235 FB          ISEC: EI                ; *** ENABLE INTERRUPTS ***
2236 21E180            LXI     H,RFLG    ; HL => RESTORE FLAG / RETRY COUNTER
2239 7E                MOV     A,M       ; GET FLAG
223A A7                ANA     A         ; HAVE WE DONE A RESTORE ?
223B CA4022            JZ      IS1       ; NO: LEAVE FLAG ALONE !
223E 3603              MVI     M,3       ; INITIALIZE WIGGLE COUNTER
2240             IS1:

2240 2AEB80            LHLD    TBC       ; GET REMAINING BYTE COUNT
2243 3E80              MVI     A,80H     ; DETERMINE SECTOR BYTE COUNT ...
2245 A5                ANA     L
2246 B4                ORA     H
2247 3E80              MVI     A,80H
2249 C24D22            JNZ     IS2       ; FULL SECTOR (128.) REQUIRED !
224C 7D                MOV     A,L       ; PARTIAL SECTOR BYTE COUNT
224D 324280      IS2:  STA     SBC       ; SAVE SECTOR BYTE COUNT
```

```
2250 3AE580            LDA     TFCN    ; GET FUNCTION CODE
2253 3D                DCR     A       ; TEST IT
2254 3E0A              MVI     A,RRTR  ; SET RETRY COUNT
2256 FA6722            JM      IS4     ; FUNCTION IS READ !

2259 CD9F24            CALL    CRCX    ; COMPUTE CRC FOR SECTOR
225C EB                XCHG
225D 224380            SHLD    CRC1    ; SAVE CRC BYTES
2260 3E04              MVI     A,WRTR  ; SET WRITE RETRY COUNT
2262 32E280            STA     CRTRY
2265 3E02      IS3:    MVI     A,VRTR  ; SET VERIFY RETRY COUNT ...
2267 32E080    IS4:    STA     BRTRY   ; INITIALIZE RETRY COUNT

226A 0E1E      DSEC:   MVI     C,SRTR  ; INITIALIZE SEARCH TRY COUNT
226C 061E      FG0:    MVI     B,HRTR  ; INITIALIZE HEADER RETRY COUNT
226E DB00      FG1:    IN      RXBUF   ; CLEAR RECEIVER BUFFER
2270 2E04              MVI     L,RGAPS+1 ; SET NO. BYTE TIMES / GAP
2272 DB03      FG2:    IN      STAT5   ; GET 5501 STATUS
2274 E608              ANI     8       ; HAS A BYTE BEEN RECEIVED ?
2276 C26E22            JNZ     FG1     ; YES: RESET & TRY AGAIN !
2279 2D                DCR     L       ; HAS GAP BEEN CONFIRMED ?
227A CA8622            JZ      FG4     ; YES !
227D 3E10              MVI     A,16    ; WAIT ONE BYTE TIME ....
227F 3D        FG3:    DCR     A
2280 C27F22            JNZ     FG3
2283 C37222            JMP     FG2     ; GO CHECK AGAIN !
2286           FG4:                    ; GAP HAS BEEN FOUND

2286 F3                DI              ; *** DISABLE INTERRUPTS ***
2287 11FFFF            LXI     D,0FFFFH ; INITIALIZE CRC RESIDUE
228A CD7424            CALL    RBYTEC  ; READ ID MARK
228D FE55              CPI     IDM     ; IS IT CORRECT ?
228F C2A522            JNZ     HER1    ; NO !
2292 CD7424            CALL    RBYTEC  ; READ TRACK NUMBER
2295 67                MOV     H,A     ; SAVE IT
2296 CD7424            CALL    RBYTEC  ; READ SECTOR NUMBER
2299 6F                MOV     L,A     ; SAVE IT
229A CD7424            CALL    RBYTEC  ; READ 1ST CRC BYTE
229D CD7424            CALL    RBYTEC  ; READ 2ND CRC BYTE
22A0 7A                MOV     A,D
22A1 B3                ORA     E       ; IS CRC CORRECT ?
22A2 CADB22            JZ      GH1     ; YES: GOT A GOOD HEADER !
22A5 FB        HER1:   EI              ; *** ENABLE INTERRUPTS ***
22A6 05                DCR     B       ; HEADER RETRIES EXHAUSTED ?
22A7 C26E22            JNZ     FG1     ; NO: TRY READING HEADER AGAIN !

22AA FB        SEEKF:  EI              ; *** ENABLE INTERRUPTS ***
22AB 060C              MVI     B,ESKF  ; SET ERROR CODE
22AD 3AE180            LDA     RFLG    ; GET "RESTORE" FLAG
22B0 A7                ANA     A       ; HAVE WE ALREADY DONE A RESTORE ?
22B1 C2BA22            JNZ     WIG1    ; YES: WIGGLE OR GIVE UP !
22B4 CD1125            CALL    POUND   ; POUND TO HOME
22B7 C32222            JMP     SEEK    ; NOW SEEK DESIRED TRACK !

22BA 21E180    WIG1:   LXI     H,RFLG  ; HL => RESTORE FLAG / WIGGLE COUNTER
22BD 35                DCR     M       ; UPDATE & TEST FLAG
22BE C8                RZ              ; ERROR EXIT !
22BF F2C422            JP      WIG2    ; COUNTER IS O.K. !
```

```
22C2 3602           MVI     M,2       ; SET COUNTER CORRECTLY
22C4 7E       WIG2: MOV     A,M       ; GET COUNTER
22C5 3D             DCR     A         ; TEST IT
22C6 CAD222         JZ      WIG3      ; 2ND WIGGLE: IN-OUT
22C9 CD4D25         CALL    STPOUT    ; STEP OUT
22CC CD3D25         CALL    STPIN     ; STEP IN
22CF C34022         JMP     IS1       ; GO TRY AGAIN !
22D2 CD3D25   WIG3: CALL    STPIN     ; STEP IN
22D5 CD4D25         CALL    STPOUT    ; STEP OUT
22D8 C34022         JMP     IS1       ; GO TRY AGAIN !

22DB 3AEF80   GH1:  LDA     TRK       ; GET DESIRED TRACK
22DE BC             CMP     H         ; CORRECT TRACK ?
22DF C2AA22         JNZ     SEEKF     ; NO !
22E2 3AEE80   GH2:  LDA     SEC       ; GET DESIRED SECTOR
22E5 BD             CMP     L         ; CORRECT SECTOR ?
22E6 CAF322         JZ      GH4       ; YES: DO IT !
22E9 FB       GH3:  EI                ; *** ENABLE INTERRUPTS ***
22EA 0D             DCR     C         ; SEARCH TRIES EXHAUSTED ?
22EB C26C22         JNZ     FGO       ; NO: TRY AGAIN !
22EE 060F           MVI     B,ECFB    ; SET POSSIBLE ERROR CODE
22F0 C3BA22         JMP     WIG1      ; GO WIGGLE OR GIVE UP !

22F3 3A4280   GH4:  LDA     SBC       ; GET SECTOR BYTE COUNT
22F6 47             MOV     B,A
22F7 2AE980         LHLD    TMEM      ; GET BUFFER POINTER
22FA 3AE580         LDA     TFCN      ; GET FUNCTION CODE
22FD 3D             DCR     A         ; TEST FUNCTION CODE
22FE FA1124         JM      RDOO      ; COMMAND IS READ !
2301 3D             DCR     A
2302 C2A623         JNZ     WROO      ; COMMAND IS WRITE !

2305 CDC824   VEOO: CALL    GDATAM    ; CONFIRM DATA MARK
2308 CA2A23         JZ      VEO2      ; GOT IT !
230B FB       VERR: EI                ; *** ENABLE INTERRUPTS ***
230C 21E080         LXI     H,BRTRY   ; POINT TO VERIFY RETRY COUNTER
230F 35             DCR     M         ; VERIFY RETRIES EXHAUSTED ?
2310 C26A22         JNZ     DSEC      ; NO: TRY AGAIN !
2313 3E03           MVI     A,3
2315 32E580         STA     TFCN      ; SET REWRITE FUNCTION CODE
2318 21E280         LXI     H,CRTRY   ; POINT TO WRITE RETRY COUNTER
231B 35             DCR     M         ; WRITE RETRIES EXHAUSTED ?
231C C26522         JNZ     IS3       ; NO: GO WRITE SECTOR AGAIN !
231F 061B           MVI     B,EVFY    ; SET ERROR CODE
2321 C9             RET               ; ERROR: VERIFY FAILURE !

2322 CD6A24   VEO1: CALL    RBYTE     ; READ NEXT DATA BYTE
2325 AE             XRA     M         ; "XOR" WITH MEMORY BYTE
2326 C20B23         JNZ     VERR      ; VERIFY ERROR !
2329 23             INX     H         ; BUMP MEMORY PNTR
232A 05       VEO2: DCR     B         ; ANY DATA LEFT ?
232B F22223         JP      VEO1      ; YES: LOOP !
232E 3A4280         LDA     SBC       ; GET SECTOR BYTE COUNT
2331 3D             DCR     A         ; ADJUST
2332 47             MOV     B,A       ; B=TRAILER COUNTER (^ TO 80H)
2333 C33D23         JMP     VEO4      ; GO CHECK TRAILERS IF ANY !

2336 CD6A24   VEO3: CALL    RBYTE     ; READ NEXT TRAILER BYTE
```

```
2339 A7                 ANA     A       ; IT SHOULD BE ZERO
233A C20B23             JNZ     VERR    ; VERIFY ERROR !
233D 04         VE04:   INR     B       ; ANY TRAILERS LEFT ?
233E F23623             JP      VE03    ; YES: LOOP !

2341 E5                 PUSH    H       ; SAVE MEMORY POINTER
2342 CD6A24             CALL    RBYTE   ; READ 1ST CRC BYTE
2345 5F                 MOV     E,A
2346 CD6A24             CALL    RBYTE   ; READ 2ND CRC BYTE
2349 57                 MOV     D,A
234A 2A4380             LHLD    CRC1    ; GET CORRECT CRC BYTES
234D CD4D34             CALL    CMPHD   ; COMPARE CRC BYTES
2350 E1                 POP     H       ; RESTORE MEMORY POINTER
2351 C20B23             JNZ     VERR    ; VERIFY ERROR !

2354 22E980     ESEC:   SHLD    TMEM    ; SAVE MEMORY POINTER
2357 2AE780             LHLD    TBLK    ; GET BLOCK NUMBER
235A 23                 INX     H       ; INCREMENT IT
235B 22E780             SHLD    TBLK    ; SAVE UPDATED BLOCK NUMBER
235E 2AEB80             LHLD    TBC     ; GET BYTE COUNT
2361 1180FF             LXI     D,-128
2364 19                 DAD     D       ; SUBTRACT ONE SECTOR
2365 DA6B23             JC      BS13    ; LAST SECTOR WAS NOT PARTIAL !
2368 210000             LXI     H,0     ; SET BYTE COUNT = 0
236B 22EB80     BS13:   SHLD    TBC     ; SAVE UPDATED BYTE COUNT
236E 7C                 MOV     A,H     ; TEST NEW BYTE COUNT ...
236F B5                 ORA     L
2370 C27E23             JNZ     BS10    ; MORE TO DO!
2373 3AE580             LDA     TFCN    ; GET FUNCTION CODE
2376 3D                 DCR     A       ; "WRITE" FUNCTION ?
2377 CA8923             JZ      BS11    ; YES: GO VERIFY!
237A 0600               MVI     B,0     ; SET "GOOD" STATUS CODE
237C FB                 EI              ; *** ENABLE INTERRUPTS ***
237D C9                 RET             ; FINISHED!

237E 21EE80     BS10:   LXI     H,SEC   ; POINT TO CURRENT SECTOR
2381 34                 INR     M       ; INCREMENT TO NEXT SECTOR
2382 3A9436             LDA     CDSEC   ; GET NO. OF SECTORS
2385 BE                 CMP     M       ; OVERFLOW TO NEXT TRACK ?
2386 C23522             JNZ     ISEC    ; NO: GO DO NEXT SECTOR!
2389 CD5236     BS11:   CALL    BSB08   ; SETUP
238C DA9D23             JC      BS12    ; RESET SECTOR!
238F 3600               MVI     M,0     ; SET SECTOR = ZERO
2391 23                 INX     H       ; POINT TO TRACK
2392 34                 INR     M       ; INCREMENT TRACK
2393 3E29               MVI     A,NTRK  ; GET HIGHEST TRACK NUMBER PLUS ONE
2395 BE                 CMP     M       ; IS IT VALID ?
2396 C22222             JNZ     SEEK    ; YES: GO START NEXT TRACK !
2399 0609               MVI     B,EBLK  ; SET ERROR CODE
239B FB                 EI              ; *** ENABLE INTERRUPTS ***
239C C9                 RET             ; ERROR EXIT: INVALID BLOCK NUMBER !

239D 3AED80     BS12:   LDA     OSEC    ; GET OLD SECTOR
23A0 32EE80             STA     SEC     ; RESET SECTOR
23A3 C33522             JMP     ISEC    ; GO DO NEXT SECTOR!

23A6 E5         WR00:   PUSH    H
23A7 CD6425             CALL    GXOUT   ; HL => XOUT0 OR XOUT1
```

```
23AA 7E              MOV     A,M       ; GET PHASE
23AB F608            ORI     08H       ; SET WRITE BIT
23AD 5F              MOV     E,A       ; COPY IT
23AE CD0225          CALL    SXO       ; SEND TO DRIVE
23B1 E1              POP     H
23B2 3EFF            MVI     A,FILL
23B4 CD5E24          CALL    WBYTE     ; WRITE A FILL BYTE
23B7 3E5A            MVI     A,DATAM
23B9 CD5E24          CALL    WBYTE     ; WRITE DATA MARK
23BC C3C423          JMP     WR02      ; GO WRITE DATA IF ANY !

23BF 7E        WR01: MOV     A,M       ; GET NEXT BYTE FROM MEMORY
23C0 23              INX     H         ; BUMP BUFFER POINTER
23C1 CD5E24          CALL    WBYTE     ; WRITE DATA BYTE
23C4 05        WR02: DCR     B         ; ANY DATA LEFT ?
23C5 F2BF23          JP      WR01      ; YES: LOOP!
23C8 3A4280          LDA     SBC       ; GET SECTOR BYTE COUNT
23CB 3D              DCR     A         ; ADJUST
23CC 47              MOV     B,A       ; B=TRAILER COUNTER (^ TO 80H)
23CD C3D423          JMP     WR04      ; GO WRITE TRAILERS IF ANY !

23D0 AF        WR03: XRA     A         ; CLEAR A REGISTER
23D1 CD5E24          CALL    WBYTE     ; WRITE A ZERO TRAILER BYTE
23D4 04        WR04: INR     B         ; ANY TRAILERS LEFT TO DO ?
23D5 F2D023          JP      WR03      ; YES: LOOP !

23D8 3A4380          LDA     CRC1
23DB CD5E24          CALL    WBYTE     ; WRITE 1ST CRC BYTE
23DE 3A4480          LDA     CRC2
23E1 CD5E24          CALL    WBYTE     ; WRITE 2ND CRC BYTE

23E4 3E20            MVI     A,32      ; WAIT FOR BUFFERS TO EMPTY (2 BYTE TIMES) ....
23E6 3D        WR05: DCR     A
23E7 C2E623          JNZ     WR05
23EA 3E1A            MVI     A,1AH     ; INTERUPT ACKNOWLEDGE BIT SET !!!!!
23EC D304            OUT     COMND     ; SET TX LINE TO "SPACING"
23EE 3E02            MVI     A,2       ; DELAY A FEW BIT TIMES ....
23F0 3D        WR06: DCR     A
23F1 C2F023          JNZ     WR06
23F4 E5              PUSH    H
23F5 CD6425          CALL    GXOUT     ; HL => XOUT0 OR XOUT1
23F8 5E              MOV     E,M       ; GET PHASE
23F9 CD0225          CALL    SXO       ; DISABLE WRITE CIRCUITRY
23FC E1              POP     H
23FD 3E18            MVI     A,18H     ; INTERUPT ACKNOWLEDGE BIT SET !!!!!
23FF D304            OUT     COMND     ; SET TX LINE TO "MARKING"

2401 3AE580          LDA     TFCN      ; GET FUNCTION CODE
2404 3D              DCR     A         ; TEST IT
2405 CA5423          JZ      ESEC      ; NORMAL WRITE: WRAP-UP !
2408 3E02            MVI     A,2
240A 32E580          STA     TFCN      ; SET FUNCTION BACK TO VERIFY
240D FB              EI                ; *** ENABLE INTERRUPTS ***
240E C36A22          JMP     DSEC      ; NOW GO VERIFY AGAIN !

2411 CDC824    RD00: CALL    GDATAM    ; CONFIRM DATA MARK
2414 CA2924          JZ      RD02      ; GOT IT !
2417 0612            MVI     B,EDSY    ; SET POSSIBLE ERROR CODE
```

```
2417 C35324            JMP     RERR

241C CD6A24   RD01:    CALL    RBYTE   ; READ NEXT DATA BYTE
241F 77                MOV     M,A     ; STORE THE BYTE
2420 AE                XRA     M       ; "XOR" BACK FROM MEMORY
2421 23                INX     H       ; BUMP MEMORY PNTR
2422 CA2924            JZ      RD02    ; O.K. !
2425 0618              MVI     B,EMEM  ; SET ERROR CODE
2427 FB                EI              ; *** ENABLE INTERRUPTS ***
2428 C9                RET             ; MEMORY ERROR !
2429 05       RD02:    DCR     B       ; ANY DATA LEFT ?
242A F21C24            JP      RD01    ; YES: LOOP !
242D 3A4280            LDA     SBC     ; GET SECTOR BYTE COUNT
2430 3D                DCR     A       ; ADJUST
2431 47                MOV     B,A     ; B=TRAILER COUNT (^ TO 80H)
2432 C33824            JMP     RD04    ; GO CHECK TRAILERS IF ANY !

2435 CD6A24   RD03:    CALL    RBYTE   ; READ NEXT TRAILER BYTE
2438 04       RD04:    INR     B       ; ANY TRAILERS LEFT ?
2439 F23524            JP      RD03    ; YES: LOOP !

243C E5                PUSH    H       ; SAVE MEMORY POINTER
243D CD6A24            CALL    RBYTE   ; READ 1ST CRC BYTE
2440 6F                MOV     L,A
2441 CD6A24            CALL    RBYTE   ; READ 2ND CRC BYTE
2444 67                MOV     H,A
2445 E5                PUSH    H       ; SAVE CRC BYTES
2446 CD9F24            CALL    CRCX    ; COMPUTE CRC FOR SECTOR
2449 E1                POP     H       ; RESTORE CRC BYTES
244A CD4D34            CALL    CMPHD   ; COMPARE CRC BYTES
244D E1                POP     H       ; RESTORE MEMORY POINTER
244E CA5423            JZ      ESEC    ; NO ERRORS !
2451 0615              MVI     B,EDCS  ; SET POSSIBLE ERROR CODE
2453 FB       RERR:    EI              ; *** ENABLE INTERRUPTS ***
2454 21E080            LXI     H,BRTRY ; POINT TO READ RETRY COUNTER
2457 35                DCR     M       ; RETRIES EXHAUSTED ?
2458 C26A22            JNZ     DSEC    ; NO: TRY AGAIN !
245B C3BA22            JMP     WIG1    ; GO WIGGLE MAYBE !


245E 4F       WBYTE:   MOV     C,A
245F DB03     WB1:     IN      STAT5   ; GET 5501 STATUS
2461 E610              ANI     10H     ; READY ?
2463 CA5F24            JZ      WB1     ; NO: WAIT !
2466 79                MOV     A,C
2467 D306              OUT     TXBUF   ; SEND THE BYTE
2469 C9                RET             ; EXIT !

246A DB03     RBYTE:   IN      STAT5   ; GET 5501 STATUS
246C E608              ANI     08H     ; GOT A BYTE ?
246E CA6A24            JZ      RBYTE   ; NO: WAIT !
2471 DB00              IN      RXBUF   ; READ THE BYTE
2473 C9                RET             ; EXIT !

2474 DB03     RBYTEC:  IN      STAT5   ; GET 5501 STATUS
2476 E608              ANI     08H     ; GOT A BYTE ?
2478 CA7424            JZ      RBYTEC  ; NO: WAIT !
247B DB00              IN      RXBUF   ; READ THE BYTE
```

```
247D F5        CRC:    PUSH    PSW
247E E5                PUSH    H
247F AB                XRA     E
2480 67                MOV     H,A
2481 OF                RRC
2482 OF                RRC
2483 OF                RRC
2484 OF                RRC
2485 E60F              ANI     OFH
2487 AC                XRA     H
2488 6F                MOV     L,A
2489 OF                RRC
248A OF                RRC
248B OF                RRC
248C 67                MOV     H,A
248D E61F              ANI     1FH
248F AA                XRA     D
2490 5F                MOV     E,A
2491 7C                MOV     A,H
2492 E6E0              ANI     OEOH
2494 AD                XRA     L
2495 57                MOV     D,A
2496 7C                MOV     A,H
2497 OF                RRC
2498 E6F0              ANI     OFOH
249A AB                XRA     E
249B 5F                MOV     E,A
249C E1                POP     H
249D F1                POP     PSW
249E C9                RET

249F 3A4280    CRCX:   LDA     SBC     ; GET SECTOR BYTE COUNT
24A2 47                MOV     B,A     ; B=DATA BYTE COUNTER
24A3 3D                DCR     A
24A4 4F                MOV     C,A     ; C="TRAILER" COUNTER (^ TO 80H)
24A5 11FFFF            LXI     D,OFFFFH ; INITIALIZE CRC RESIDUE
24A8 2AE980            LHLD    TMEM    ; GET MEMORY POINTER
24AB 3E5A              MVI     A,DATAM ; DATA MARK
24AD CD7D24            CALL    CRC     ; UPDATE CRC RESIDUE
24B0 C3B824            JMP     CR2
24B3 7E        CR1:    MOV     A,M     ; GET NEXT BYTE
24B4 23                INX     H       ; BUMP MEMORY POINTER
24B5 CD7D24            CALL    CRC     ; UPDATE CRC RESIDUE
24B8 05        CR2:    DCR     B       ; ANY MORE DATA ?
24B9 F2B324            JP      CR1     ; YES: LOOP !
24BC C3C324            JMP     CR4
24BF AF        CR3:    XRA     A
24C0 CD7D24            CALL    CRC     ; UPDATE CRC RESIDUE
24C3 OC        CR4:    INR     C       ; ANY TRAILERS LEFT ?
24C4 F2BF24            JP      CR3     ; YES: LOOP !
24C7 C9                RET     ; DE = CRC !

24C8 CD6A24    GDATAM: CALL    RBYTE   ; READ GARBAGE BYTE
24CB CD6A24            CALL    RBYTE   ; READ NEXT BYTE
24CE FE5A              CPI     DATAM   ; DATA MARK ?
24D0 C8                RZ              ; YES !
```

```
24D1 CD6A24          CALL    RBYTE   ; READ NEXT BYTE
24D4 FE5A            CPI     DATAM   ; DATA MARK ?
24D6 C8              RZ              ; YES !
24D7 CD6A24          CALL    RBYTE   ; READ NEXT BYTE
24DA FE5A            CPI     DATAM   ; DATA MARK ?
24DC C9              RET             ; <Z>: O.K., <NZ>: ERROR !

24DD AF      VTP:    XRA     A       ; NO COMMENT
24DE 32E180          STA     RFLG    ; CLEAR RESTORE FLAG
24E1 CD6A25          CALL    GCTRK   ; HL => CTRK0 OR CRTK1
24E4 7E              MOV     A,M     ; GET SUPPOSED CURRENT TRACK
24E5 FE29            CPI     NTRK    ; IS IT VALID ?
24E7 D21125          JNC     POUND   ; NO: GO POUND !
24EA C602            ADI     2       ; ADJUST TRACK NUMBER
24EC D603    VTP1:   SUI     3       ; "MOD" TRACK NUMBER ...
24EE F2EC24          JP      VTP1    ; ... BY 3
24F1 5F              MOV     E,A
24F2 3E08            MVI     A,8
24F4 0F      VTP2:   RRC
24F5 1C              INR     E
24F6 C2F424          JNZ     VTP2
24F9 5F              MOV     E,A     ; E=CORRECT PHASE FOR SUPPOSED TRACK
24FA CD6425          CALL    GXOUT   ; HL => XOUT0 OR XOUT1
24FD 7E              MOV     A,M     ; GET SUPPOSED PHASE
24FE BB              CMP     E       ; CHECK IF IT'S CORRECT
24FF C21125          JNZ     POUND   ; NOT CORRECT: GO POUND !
2502 3AE680  SX0:    LDA     TDRV    ; GET DRIVE NUMBER
2505 A7              ANA     A       ; TEST DRIVE NUMBER
2506 3E10            MVI     A,10H   ; SELECT BIT FOR DRIVE ZERO
2508 CA0C25          JZ      SX01    ; SELECT BIT IS CORRECT !
250B 07              RLC             ; CHANGE TO SELECT BIT FOR DRIVE ONE
250C B3      SX01:   ORA     E       ; MERGE
250D D307            OUT     EXTOT   ; SELECT DRIVE AND MOTOR ON
250F 37              STC             ; SET CARRY
2510 C9              RET             ; RETURN <C>: NO POUND !

2511 CD6425  POUND:  CALL    GXOUT   ; HL => XOUT0 OR XOUT1
2514 1E04            MVI     E,4     ; SET PHASE
2516 73              MOV     M,E     ; STORE IT
2517 CD0225          CALL    SX0     ; SEND TO DRIVE
251A 3E03            MVI     A,3
251C 32E180          STA     RFLG    ; SET WIGGLE COUNTER
251F CD6A25          CALL    GCTRK   ; HL => CTRK0 OR CTRK1
2522 362A            MVI     M,42    ; SET CURRENT TRACK NUMBER
2524 AF              XRA     A       ; SET TRACK ZERO

2525 47      STEPS:  MOV     B,A     ; B=DESIRED TRACK
2526 CD6A25          CALL    GCTRK   ; HL => CTRK0 OR CTRK1
2529 78      STP1:   MOV     A,B     ; GET DESIRED TRACK
252A BE              CMP     M       ; COMPARE WITH CURRENT TRACK
252B C8              RZ              ; WE ARE THERE: EXIT !
252C DA3625          JC      STP2    ; NEED TO STEP OUT !
252F 34              INR     M       ; UPDATE CURRENT TRACK
2530 CD3D25          CALL    STPIN   ; STEP IN ONE TRACK
2533 C32925          JMP     STP1    ; SEE IF WE'RE THERE NOW !
2536 35      STP2:   DCR     M       ; UPDATE CURRENT TRACK
2537 CD4D25          CALL    STPOUT  ; STEP OUT ONE TRACK
```

```
253A C32925             JMP     STP1    ; SEE IF WE'RE THERE NOW !

253D E5       STPIN:    PUSH    H
253E CD6425             CALL    GXOUT   ; HL => XOUTO OR XOUT1
2541 7E                 MOV     A,M     ; GET PHASE
2542 07                 RLC             ; SHIFT PHASE LEFT ONE
2543 FE08               CPI     8       ; SHIFT OUT ?
2545 DA5825             JC      STP3    ; NO: GO STEP !
2548 3E01               MVI     A,1     ; RESET TO PHASE ZERO
254A C35825             JMP     STP3    ; GO STEP !

254D E5       STPOUT:   PUSH    H
254E CD6425             CALL    GXOUT   ; HL => XOUTO OR XOUT1
2551 7E                 MOV     A,M     ; GET PHASE
2552 0F                 RRC             ; SHIFT PHASE RIGHT ONE
2553 D25825             JNC     STP3    ; GO STEP !
2556 3E04               MVI     A,4     ; RESET TO PHASE TWO
2558 77       STP3:     MOV     M,A     ; STORE PHASE BYTE
2559 5F                 MOV     E,A     ; COPY PHASE BYTE
255A CD0225             CALL    SXO     ; SEND TO DRIVE
255D E1                 POP     H

255E 3A2700   STPWAT:   LDA     STEPCD  ;STEP RATE FOR CHEAP DISK
2561 C31C34             JMP     WATS    ; DELAY ONE STEP TIME & RETURN

2564 21AF81   GXOUT:    LXI     H,XOUTO
2567 C37325             JMP     WXYZ

256A 21B181   GCTRK:    LXI     H,CTRKO
256D C37325             JMP     WXYZ

2570 21B581   GCUCNT:   LXI     H,CUCNTO
2573 3AE680   WXYZ:     LDA     TDRV
2576 A7                 ANA     A
2577 C8                 RZ
2578 23                 INX     H
2579 C9                 RET

              ; **************************************************
              [redacted]                ; FILE CONTROL SYSTEM - VERSION 3.0

              ;COPYRIGHT (C) 1977,1978
              ; BY INTELLIGENT SYSTEMS CORPORATION
              ;


              ; COMMAND TABLE:

257A 494E49   CMTAB:    DB      'INI'
257D 2127               DW      INIOO
257F 444952             DB      'DIR'
2582 9127               DW      DIROO
2584 534156             DB      'SAV'
2587 3328               DW      SAVOO
2589 4C4F41             DB      'LOA'
```

```
258C 6928              DW      LOA00
258E 52554E            DB      'RUN'
2591 5629              DW      RUN00
2593 44454C            DB      'DEL'
2596 B529              DW      DEL00
2598 52454E            DB      'REN'
259B C12A              DW      REN00
259D 434F50            DB      'COP'
25A0 202B              DW      COP00
25A2 445550            DB      'DUP'
25A5 932B              DW      DUP00
25A7 444556            DB      'DEV'
25AA 9629              DW      DEV00
25AC 524541            DB      'REA'
25AF F126              DW      REA00
25B1 575249            DB      'WRI'
25B4 EE26              DW      WRI00
25B6 00                DB      0       ;TERMINATOR !
     (25B7)            FCSORG  EQU     $       ;SET FOR CONTINUATION

                ; OPEN TYPE CODE BIT DEFINITIONS :

     (0001)     FNEW    EQU     01H     ;0: OLD FILE, 1: NEW FILE

                ; DIRECTORY ENTRY TYPE CODE BIT DEFINITIONS :

     (0001)     TFREE   EQU     01H     ;"FREE SPACE" ENTRY - BYTE VALUE

     (0001)     TPROT   EQU     01H     ;PROTECTED FILE
     (0002)     TFILE   EQU     02H     ;PERMANENT FILE ENTRY

                ; RAM ALLOCATION :

25B7 (8082)             ORG     ZRAM

8082 (0001)     ZFPB:   DS      1
8083 (0001)     ZFATR:  DS      1
8084 (0006)     ZFNAM:  DS      6
808A (0003)     ZFTYP:  DS      3
808D (0001)     ZFVER:  DS      1
808E (0002)     ZFSBK:  DS      2
8090 (0002)     ZFSIZ:  DS      2
8092 (0001)     ZFLBC:  DS      1
8093 (0002)     ZFLAD:  DS      2
8095 (0002)     ZFSAD:  DS      2
8097 (0001)             DS      1
8098 (0001)     ZFDBK:  DS      1
8099 (0001)     ZFDEN:  DS      1
809A (0002)     ZFAUX:  DS      2
809C (0002)     ZFHAN:  DS      2
809E (0001)     ZFFCN:  DS      1
809F (0001)     ZFDRV:  DS      1
80A0 (0002)     ZFBLK:  DS      2
80A2 (0002)     ZFBUF:  DS      2
80A4 (0002)     ZFXBC:  DS      2
80A6 (0002)     ZFPTR:  DS      2
```

```
80A8                ZFPBE:  ; END OF AUX. FPB, END OF BASIC INPUT BUFFER

80A8 (80F0)                 ORG     ORAM

80F0 (0002)         DFDV:   DS      2       ;DEFAULT DEVICE (ASCII)
80F2 (0001)         DFUN:   DS      1       ;DEFAULT UNIT (ASCII)

80F3 (0002)         FPBP:   DS      2       ;FILE PARAMETER BLOCK POINTER
80F5 (0001)         OCODE:  DS      1       ;OPEN TYPE CODE
80F6 (0001)         OVERS:  DS      1       ;ORIGINAL VERSION

                    ; SYSTEM FILE PARAMETER BLOCK ALLOCATION :

80F7 (0001)         FPB:    DS      1       ;OPEN TYPE CODE
80F8 (0001)         FATR:   DS      1       ;ATTRIBUTE BYTE
80F9 (0006)         FNAM:   DS      6       ;FILE NAME
80FF (0003)         FTYP:   DS      3       ;FILE TYPE
8102 (0001)         FVER:   DS      1       ;FILE VERSION NUMBER
8103 (0002)         FSBK:   DS      2       ;STARTING BLOCK NUMBER
8105 (0002)         FSIZ:   DS      2       ;NUMBER OF BLOCKS
8107 (0001)         FLBC:   DS      1       ;BYTE COUNT OF LAST BLOCK
8108 (0002)         FLAD:   DS      2       ;LOAD ADR. FOR "IMAGE" FILE
810A (0002)         FSAD:   DS      2       ;START ADR. FOR "IMAGE" FILE
810C (0001)                 DS      1       ;SPARE
810D (0001)         FDBK:   DS      1       ;DIRECTORY BLOCK NUMBER
810E (0001)         FDEN:   DS      1       ;DIRECTORY ENTRY NUMBER
810F (0002)         FAUX:   DS      2       ;NEW FILE CLOSING SIZE, OR ...
                                            ;... AUX. BYTE COUNT FOR SEQUENTIAL ROUTINES
8111 (0002)         FHAN:   DS      2       ;HANDLER ADDRESS
8113 (0001)         FFCN:   DS      1       ;HANDLER FUNCTION CODE
8114 (0001)         FDRV:   DS      1       ;DRIVE NUMBER
8115 (0002)         FBLK:   DS      2       ;BLOCK NO. FOR TRANSFER
8117 (0002)         FBUF:   DS      2       ;BUFFER POINTER FOR TRANSFER
8119 (0002)         FXBC:   DS      2       ;BYTE COUNT FOR TRANSFER
811B (0002)         FPTR:   DS      2       ;BBUF PNTR FOR SEQUENTIAL ROUTINES
811D                FPBE:                   ;END OF SYSTEM FPB


811D                DBF:                    ;DIR BLOCK BUFFER
811D (0001)         DBLK:   DS      1       ;"THIS" DIR BLOCK NUMBER
811E (0001)         MDBLK:  DS      1       ;MAX. DIR BLOCK NUMBER
811F (007E)                 DS      126     ;REMAINDER OF 128. BYTE DIR BLOCK BUFFER
819D                DBFE:                   ;END OF DIR BLOCK BUFFER

                    ; THE FOLLOWING 18. BYTES ARE THE DIR BLOCK BUFFER EXTENSION
                    ; USED BY CLOSE WHEN THE "FREE" ENTRY MOVES TO THE NEXT BLOCK:

819D (0002)         XFHAN:  DS      2       ;AUX. HANDLER ADDRESS
819F (0001)         XFFCN:  DS      1       ;AUX. HANDLER FUNCTION CODE
81A0 (0001)         XFDRV:  DS      1       ;AUX. DRIVE NUMBER
81A1 (0002)         XFBLK:  DS      2       ;AUX. BLOCK NUMBER
81A3 (0002)         XFBUF:  DS      2       ;AUX. BUFFER POINTER
81A5 (0002)         XFXBC:  DS      2       ;AUX. BYTE COUNT
81A7 (0004)                 DS      4       ; *** USED BY COPY ***
81AB (0002)         TMP1:   DS      2       ; USED BY COPY & MAYBE OTHERS ?
81AD (0002)                 DS      2
81AF (81AF)                 ORG     $       ; *** SHOULD NOT BE PAST "CRTRAM" ***
```

```
                81AF (25B7)             ORG     FCSORG

25B7 21C625     CRTMO:  LXI     H,CRTMSG
25BA CDF433             CALL    OSTR
25BD 2A3600             LHLD    ACRTSP
25C0 F9                 SPHL
25C1 213800             LXI     H,BEGEX
25C4 E5                 PUSH    H
25C5 E9                 PCHL

25C6 06020B1D   CRTMSG: DB      6,2,11,29,'COMPUCOLOR II ',17,'CRT ',22,'MODE V6.78'
25CA 434F4D50
25CE 55434F4C
25D2 4F522049
25D6 49201143
25DA 52542016
25DE 4D4F4445
25E2 2056362E
25E6 3738
25E8 120A0BEF           DB      18,10,11,239

25EC 117A25     FCS:    LXI     D,CMTAB ;POINT TO COMMAND TABLE
25EF E5         F1:     PUSH    H       ;SAVE COMMAND LINE PNTR
25F0 0603               MVI     B,3     ;SET COUNTER
25F2 1A         F2:     LDAX    D       ;GET NEXT COMMAND CHARACTER
25F3 13                 INX     D
25F4 BE                 CMP     M       ;MATCH ?
25F5 23                 INX     H
25F6 CA0726             JZ      F4      ;YES!
25F9 13         F3:     INX     D       ;SKIP TO ...
25FA 05                 DCR     B       ;... NEXT ...
25FB F2F925             JP      F3      ;... COMMAND
25FE E1                 POP     H       ;GET COMMAND LINE PNTR
25FF 1A                 LDAX    D       ;GET BYTE
2600 A7                 ANA     A       ;END OF COMMAND TABLE ?
2601 C2EF25             JNZ     F1      ;NO: TRY NEXT COMMAND!
2604 0603       IVC:    MVI     B,EIVC  ;SET ERROR CODE
2606 C9                 RET             ;ERROR EXIT: INVALID COMMAND!
2607 05         F4:     DCR     B       ;ENOUGH ?
2608 C2F225             JNZ     F2      ;NO: CHECK NEXT CHARACTER!
260B E3                 XTHL
260C 212226             LXI     H,FCSEX ;POINT TO EXIT ROUTINE
260F E3                 XTHL
2610 EB                 XCHG
2611 4E                 MOV     C,M     ;GET LOBYTE OF ROUTINE ADDRESS
2612 23                 INX     H
2613 46                 MOV     B,M     ;GET HIBYTE OF ROUTINE ADDRESS
2614 C5                 PUSH    B       ;PUT ROUTINE ADDRESS ON STACK
2615 21F780             LXI     H,FPB   ;POINT TO FPB
2618 22F380             SHLD    FPBP    ;SAVE POINTER
261B EB                 XCHG
261C CD7E34             CALL    LTNOR   ;SKIP TRAILING LETTERS
261F C36034             JMP     SPNOR   ;SKIP SPACES & GO TO ROUTINE!

2622 C5         FCSEX:  PUSH    B       ;SAVE STATUS CODE
2623 CDA526             CALL    RESET   ;RESET DEVICES
```

```
2626 C1                 POP     B       ;RESTORE STATUS CODE
2627 78                 MOV     A,B     ;GET STATUS CODE
2628 A7                 ANA     A       ;TEST IT
2629 C9                 RET             ;RETURN TO CALLER

262A CDEC25     FCSEM:  CALL    FCS     ;PROCESS COMMAND LINE
262D 78         EMESS:  MOV     A,B     ;COPY STATUS CODE
262E A7                 ANA     A       ;TEST IT
262F C8                 RZ              ;NO ERRORS!
2630 F5                 PUSH    PSW     ;SAVE ERROR CODE & FLAGS
2631 214826             LXI     H,FERS1
2634 CDF433             CALL    OSTR    ;PRINT MESSAGE
2637 215726             LXI     H,FERS2-3
263A 48                 MOV     C,B
263B 0600               MVI     B,0
263D 09                 DAD     B
263E 0603               MVI     B,3
2640 CDC034             CALL    PSTR    ;PRINT ERROR CODE
2643 CD8B33             CALL    CRLF
2646 F1                 POP     PSW     ;RESTORE ERROR CODE & FLAGS
2647 C9                 RET             ;RETURN TO CALLER

                ; ERROR CODE DEFINITIONS :

2648 FF06010B FERS1:    DB 255,6,1,11,'FCS ERROR - E',239
264C 46435320
2650 4552524F
2654 52202D20
2658 45EF

265A            FERS2:
265A 495643             DB      'IVC'   ;INVALID COMMAND
     (0003)     EIVC    EQU     $-FERS2
265D 4D564E             DB      'MVN'   ;MISSING VOLUME NAME
     (0006)     EMVN    EQU     $-FERS2
2660 53594E             DB      'SYN'   ;SYNTAX ERROR
     (0009)     ESYN    EQU     $-FERS2
2663 444952             DB      'DIR'   ;DIRECTORY ERROR
     (000C)     EDIR    EQU     $-FERS2
2666 495650             DB      'IVP'   ;INVALID PARAMETER(S)
     (000F)     EIVP    EQU     $-FERS2
2669 4E5645             DB      'NVE'   ;NO VOLUME ENTRY IN DIRECTORY
     (0012)     ENVE    EQU     $-FERS2
266C 4D464E             DB      'MFN'   ;MISSING FILE NAME
     (0015)     EMFN    EQU     $-FERS2
266F 4D4456             DB      'MDV'   ;MISSING DEVICE NAME
     (0018)     EMDV    EQU     $-FERS2
2672 4D5652             DB      'MVR'   ;MISSING VERSION
     (001B)     EMVR    EQU     $-FERS2
2675 495644             DB      'IVD'   ;INVALID DEVICE
     (001E)     EIVD    EQU     $-FERS2
2678 4E5341             DB      'NSA'   ;NO START ADDRESS
     (0021)     ENSA    EQU     $-FERS2
267B 44464E             DB      'DFN'   ;DUPLICATE FILE NAME
     (0024)     EDFN    EQU     $-FERS2
267E 564F56             DB      'VOV'   ;VERSION NUMBER OVERFLOW
     (0027)     EVOV    EQU     $-FERS2
```

```
2681 464E46           DB      'FNF'   ;FILE NOT FOUND
     (002A)   EFNF    EQU     $-FERS2
2684 445246           DB      'DRF'   ;DIRECTORY FULL
     (002D)   EDRF    EQU     $-FERS2
2687 465244           DB      'FRD'   ;FILE READ ERROR
     (0030)   EFRD    EQU     $-FERS2
268A 465752           DB      'FWR'   ;FILE WRITE ERROR
     (0033)   EFWR    EQU     $-FERS2
268D 57535A           DB      'WSZ'   ;FILE TOO LARGE FOR WRITE
     (0036)   EWSZ    EQU     $-FERS2
2690 52535A           DB      'RSZ'   ;FILE TOO LARGE FOR READ
     (0039)   ERSZ    EQU     $-FERS2
2693 44454C           DB      'DEL'   ;DELETE ERROR
     (003C)   EDEL    EQU     $-FERS2
2696 445550           DB      'DUP'   ;ERROR DURING DUPLICATE
     (003F)   EDUP    EQU     $-FERS2
2699 53495A           DB      'SIZ'   ;DEVICE SIZES NOT SAME
     (0042)   ESIZ    EQU     $-FERS2
269C 434F50           DB      'COP'   ;ERROR DURING COPY
     (0045)   ECOP    EQU     $-FERS2
269F 495654           DB      'IVT'   ;INVALID FILE TYPE
     (0048)   EIVT    EQU     $-FERS2
26A2 424C46           DB      'BLF'   ;BAD '.LDA' FILE
     (004B)   EBLF    EQU     $-FERS2

26A5 010301  RESET:   LXI     B,103H  ;SET FLAG & COUNTER
26A8 218E36           LXI     H,HDVCT ;POINT TO HANDLER VECTOR AREA
26AB 3EC3    RS01:    MVI     A,0C3H  ;SET 'JMP' BYTE
26AD BE               CMP     M       ;VECTOR EMPTY ?
26AE C2CE26           JNZ     RS02    ;YES: SKIP IT!
26B1 E5               PUSH    H       ;SAVE HANDLER PNTR
26B2 C5               PUSH    B       ;SAVE FLAG & COUNTER
26B3 EB               XCHG            ;DE=HANDLER PNTR
26B4 21E526           LXI     H,OFFPB ;POINT TO HANDLER PB
26B7 CD012F           CALL    JMPD    ;CALL HANDLER
26BA C1               POP     B       ;RESTORE FLAG & COUNTER
26BB E1               POP     H       ;GET HANDLER PNTR
26BC E5               PUSH    H       ;SAVE HANDLER PNTR
26BD 23               INX     H       ;POINT TO DEVICE NAME ...
26BE 23               INX     H
26BF 23               INX     H
26C0 5E               MOV     E,M     ;GET 1ST CHAR. OF NAME
26C1 23               INX     H
26C2 56               MOV     D,M     ;GET 2ND CHAR. OF NAME
26C3 2AF080           LHLD    DFDV    ;GET DEFAULT DEVICE NAME
26C6 CD5334           CALL    CMPDH   ;MATCH ?
26C9 E1               POP     H       ;RESTORE HANDLER PNTR
26CA C2CE26           JNZ     RS02    ;NO MATCH: STEP TO NEXT!
26CD 04               INR     B       ;INDICATE DEFAULT DEVICE SEEN
26CE 110A00  RS02:    LXI     D,10
26D1 19               DAD     D       ;STEP TO NEXT HANDLER
26D2 0D               DCR     C       ;ANY MORE ?
26D3 C2AB26           JNZ     RS01    ;YES: LOOP!
26D6 05               DCR     B       ;WAS DEFAULT DEVICE SEEN ?
26D7 C0               RNZ             ;YES: EXIT!
26D8 2A8B36           LHLD    IDEV    ;GET INITIAL DEFAULT DEVICE
26DB 22F080           SHLD    DFDV    ;SET DEFAULT DEVICE
```

```
26DE 3A8D36           LDA     IUNT    ;GET INITIAL UNIT NO.
26E1 32F280           STA     DFUN    ;STORE IT
26E4 C9               RET             ;EXIT!

26E5 FC00   OFFPB:    DB      -4,0    ;"TURN OFF" FUNCTION


26E7 CD6034 CKEND:    CALL    SPNOR   ;SKIP SPACES
26EA C8               RZ              ;RETURN <Z> IF END OF LINE
26EB 0609   ERSYN:    MVI     B,ESYN  ;SET ERROR CODE
26ED C9               RET             ;RETURN <NZ> IF NOT END OF LINE


26EE 3E01   WRI00:    MVI     A,1     ;SET WRITE CODE
26F0 FE               DB.     0FEH    ;*** SKIP 1 BYTE ***
26F1 AF     REA00:    XRA     A       ;SET READ CODE
26F2 321381           STA     FFCN    ;SAVE FUNCTION CODE
26F5 CDDE2F           CALL    PDV     ;GET DEVICE NAME
26F8 D8               RC              ;ERROR!
26F9 CDF634           CALL    GN2Z    ;GET BLOCK NUMBER
26FC D2D02C           JNC     GM02    ;NO NUMBER: ERROR!
26FF EB               XCHG
2700 221581           SHLD    FBLK    ;STORE BLOCK NUMBER
2703 EB               XCHG
2704 CDA42C           CALL    GMPRM   ;GET MEMORY PARAMS
2707 D8               RC              ;ERROR!
2708 CDE726           CALL    CKEND   ;CHECK FOR END OF LINE
270B C0               RNZ             ;NO: ERROR!
270C EB               XCHG
270D 221781           SHLD    FBUF    ;STORE BUFFER ADDRESS
2710 60               MOV     H,B
2711 69               MOV     L,C
2712 221981           SHLD    FXBC    ;STORE BYTE COUNT
2715 3A1381           LDA     FFCN    ;GET FUNCTION CODE
2718 211181           LXI     H,FHAN
271B CDFC2E           CALL    CHDLR   ;CALL HANDLER
271E 0600   ODRET:    MVI     B,0
2720 C9               RET

                ; INITIALIZE VOLUME DIRECTORY :

2721 CDDE2F INI00:    CALL    PDV     ;PARSE DEVICE NAME
2724 D8               RC              ;ERROR: INVALID DEVICE!
2725 C8               RZ              ;ERROR: MISSING DEVICE NAME!
2726 CD6034           CALL    SPNOR   ;SKIP SPACES
2729 112081           LXI     D,DBF+3 ;POINT TO BUFFER
272C 060A             MVI     B,10    ;SET COUNTER
272E CD9534           CALL    MSTR    ;MOVE VOLUME NAME
2731 0606             MVI     B,EMVN  ;SET ERROR CODE
2733 C8               RZ              ;ERROR: MISSING VOLUME NAME!
2734 CD8834           CALL    GCMA    ;SKIP COMMA IF IT'S THERE
2737 CDF634           CALL    GN2Z    ;GET NUMBER IF IT'S THERE
273A CDE726           CALL    CKEND   ;CHECK FOR END OF LINE
273D C0               RNZ             ;NO: ERROR!
273E D5               PUSH    D       ;SAVE USER'S NUMBER
273F 211181           LXI     H,FHAN  ;POINT TO HANDLER ADDRESS
2742 E5               PUSH    H       ;SAVE PNTR
```

```
2743 3EFE              MVI     A,-2        ;SET "GET" CODE
2745 CDFC2E            CALL    CHDLR       ;CALL HANDLER
2748 E1                POP     H           ;RESTORE PNTR
2749 3EFF              MVI     A,-1        ;SET "GET SIZE" CODE
274B CDFC2E            CALL    CHDLR       ;CALL HANDLER
274E D1                POP     D           ;RESTORE USER'S NUMBER
274F 060C              MVI     B,EDIR      ;SET ERROR CODE
2751 D8                RC                  ;ERROR GETTING SIZE!
2752 E5                PUSH    H           ;SAVE DEVICE SIZE
2753 7B                MOV     A,E         ;GET USER-SPECIFIED DIR. SIZE
2754 A7                ANA     A           ;IS IT ZERO ?
2755 C26127            JNZ     INI01       ;NO: USE IT!
2758 29                DAD     H
2759 7C                MOV     A,H         ;TOTAL BLKS./128.
275A DA6627            JC      INI02       ;OVERFLOW: USE MAX.
275D FE01              CPI     1           ; ENSURE AT LEAST ONE ...
275F CE00              ACI     0
2761 FE21      INI01:  CPI     DSBUFS/128 + 1  ;VALID ?
2763 DA6827            JC      INI03           ;YES: USE IT!
2766 3E20      INI02:  MVI     A,DSBUFS/128    ;SET MAX. NO. OF DIR. BLOCKS
2768 5F        INI03:  MOV     E,A
2769 1600              MVI     D,0         ;DE=STRT BLK OF FREE SPACE
276B 210000            LXI     H,0
276E 221581            SHLD    FBLK        ;SET BLOCK NUMBER
2771 211D81            LXI     H,DBF       ;POINT TO BUFFER
2774 221781            SHLD    FBUF        ;SET BUFFER PNTR
2777 72                MOV     M,D         ;SET DIR. BLK. NO. ZERO
2778 23                INX     H
2779 3D                DCR     A
277A 77                MOV     M,A         ;SET MAX. DIR. BLK. NO.
277B 23                INX     H
277C 3641              MVI     M,41H       ;SET "VOLUME" ATTRIBUTE BYTE
277E E1                POP     H           ;GET DEVICE SIZE
277F CD5934            CALL    SUBHD       ;COMPUTE NO. OF FREE BLOCKS
2782 44                MOV     B,H
2783 4D                MOV     C,L
2784 213481            LXI     H,DBF+2+21
2787 CD962F            CALL    SFR         ;SETUP FREE ENTRY
278A CD752F            CALL    WRDIR       ;WRITE DIRECTORY BLOCK
278D D8                RC                  ;ERROR!
278E C39927            JMP     DIR01       ;GO LIST DIRECTORY!

               ; LIST DIRECTORY :
2791 CDDE2F    DIR00:  CALL    PDV         ;GET DEVICE NAME
2794 D8                RC                  ;ERROR: INVALID DEVICE!
2795 CDE726            CALL    CKEND       ;CHECK FOR END OF LINE
2798 C0                RNZ                 ;NO: ERROR!
2799 11F780    DIR01:  LXI     D,FPB       ;POINT TO FPB
279C 21F72C            LXI     H,MS1       ;POINT TO DIR. HEADER MSG.
279F CDF433            CALL    OSTR        ;PRINT IT
27A2 CDD42C            CALL    PRDEV       ;PRINT DEVICE NAME
27A5 111181            LXI     D,FHAN      ;POINT TO FHAN
27A8 CD602D            CALL    OPDIR       ;'OPEN' DIRECTORY
27AB D8                RC                  ;ERROR!
27AC C5                PUSH    B           ;SAVE ...
27AD D5                PUSH    D           ;... THINGS ...
27AE E5                PUSH    H           ;...
27AF 23                INX     H           ;SKIP TYPE CODE
```

```
2780 060A            MVI     B,10      ;SET COUNT
27B2 CDBD34          CALL    PSSTR     ;PRINT VOLUME LABEL
27B5 CDB334          CALL    PSPAC     ;PRINT A SPACE
27B8 3A1E81          LDA     DBF+1     ;GET MAX. DIR. BLK NO.
27BB 3C              INR     A         ;CHANGE TO NO. OF BLOCKS
27BC CD9B33          CALL    LBYT      ;PRINT IT
27BF 21042D          LXI     H,MS2     ;POINT TO HEADING
27C2 CDF433          CALL    OSTR      ;PRINT HEADING
27C5 E1        DIR02: POP     H         ;RESTORE ...
27C6 D1              POP     D         ;... THINGS ...
27C7 C1              POP     B         ;...
27C8 CD8B33          CALL    CRLF      ;PRINT CR&LF
27CB CDD533          CALL    RTST      ;GOT A KEY ?
27CE C2D927          JNZ     DIR05     ;NO: PROCEED !
27D1 FE0A            CPI     10        ;IS IT LF ?
27D3 CC9233          CZ      LO        ;ECHO LF !
27D6 CA1E27          JZ      GDRET     ;YES: STOP !
27D9 CD862D    DIR05: CALL   GNDE      ;GET NEXT DIRECTORY ENTRY
27DC D8              RC                ;ERROR!
27DD CA2A28          JZ      DIR04     ;FINISHED!
27E0 C5              PUSH    B         ;SAVE ...
27E1 D5              PUSH    D         ;... THINGS ...
27E2 E5              PUSH    H         ;...
27E3 46              MOV     B,M       ;GET TYPE CODE
27E4 CDCA34          CALL    PSBYT     ;PRINT TYPE CODE
27E7 3E01            MVI     A,TFREE   ;GET 'FREE' TYPE CODE
27E9 B8              CMP     B         ;IS THIS THE 'FREE' ENTRY ?
27EA CA1628          JZ      DIR03     ;YES!
27ED 0606            MVI     B,6       ;SET COUNTER
27EF CDBD34          CALL    PSSTR     ;PRINT NAME
27F2 3E2E            MVI     A,'.'     ;PRINT A ...
27F4 CD9233          CALL    LO        ;... PERIOD
27F7 0603            MVI     B,3       ;SET COUNTER
27F9 CDC034          CALL    PSTR      ;PRINT TYPE
27FC 3E3B            MVI     A,';'     ;PRINT A ...
27FE CD9233          CALL    LO        ;... SEMICOLON
2801 CDCD34          CALL    PBYT      ;PRINT VERSION NUMBER
2804 CDD534          CALL    PSNUM     ;PRINT STARTING BLOCK NUMBER
2807 CDD534          CALL    PSNUM     ;PRINT SIZE
280A CDCA34          CALL    PSBYT     ;PRINT LAST BLK BYTE COUNT
280D CDD234          CALL    P2SNUM    ;PRINT LOAD ADDRESS
2810 CDD534          CALL    PSNUM     ;PRINT START ADDRESS
2813 C3C527          JMP     DIR02     ;STEP TO NEXT ENTRY!
2816 EB        DIR03: XCHG
2817 21332D          LXI     H,MS3     ;POINT TO STRING
281A CDF433          CALL    OSTR      ;PRINT IT
281D 210A00          LXI     H,FSBK-FNAM
2820 19              DAD     D         ;POINT TO FSBK
2821 CDD534          CALL    PSNUM     ;PRINT STARTING BLOCK NUMBER
2824 CDD534          CALL    PSNUM     ;PRINT SIZE
2827 CD8B33          CALL    CRLF
282A E1        DIR04: POP     H         ;CLEAN ...
282B E1              POP     H         ;... THE ...
282C E1              POP     H         ;... STACK
282D CD8B33          CALL    CRLF
2830 0600            MVI     B,0       ;SET O.K. CODE
2832 C9              RET               ;THAT'S ALL!
```

```
                ; SAVE "IMAGE" TYPE FILE :
2833 CD7430     SAV00:  CALL    PPFSP   ;GET FILE SPECIFICATION
2836 D8                 RC              ;ERROR!
2837 CDA42C             CALL    GMPRM   ;GET MEMORY PARAMS
283A D8                 RC              ;ERROR!
283B EB                 XCHG
283C 220881             SHLD    FLAD    ;STORE LOAD ADDRESS
283F EB                 XCHG
2840 CDE734             CALL    GN1D    ;GET START ADDRESS
2843 EB                 XCHG
2844 220A81             SHLD    FSAD    ;STORE START ADDRESS
2847 EB                 XCHG
2848 CDE434             CALL    GN1Z    ;GET AUX. MEM. ADDRESS
284B CDE726             CALL    CKEND   ;CHECK FOR END OF LINE
284E C0                 RNZ             ;NO: ERROR!
284F 3E01       WF2:    MVI     A,FNEW  ;SET OPEN TYPE CODE
2851 CD862C             CALL    OPENX   ;OPEN THE FILE
2854 D8                 RC              ;ERROR!
2855 CDCC2E             CALL    WRITE   ;WRITE THE FILE
2858 D8                 RC              ;ERROR!
2859 CD262F     CLX:    CALL    CLOSE   ;CLOSE THE FILE
285C D8                 RC              ;ERROR!
285D 111A00             LXI     D,FHAN-FFB
2860 19                 DAD     D       ;POINT TO FHAN
2861 3EFD               MVI     A,-3    ;SET "SUB USER" FUNCTION
2863 CDFC2E             CALL    CHDLR   ;CALL THE HANDLER
2866 AF                 XRA     A       ;CLEAR C-FLAG
2867 47                 MOV     B,A     ;SET GOOD STATUS
2868 C9                 RET             ;EXIT

                ; LOAD A FILE :
2869 015A2D     LOA00:  LXI     B,LTYP  ;SET DEFAULT TYPE
286C CD7730             CALL    PFSPC   ;GET FILE SPECIFICATION
286F D8                 RC              ;ERROR !
2870 CDAB2A             CALL    QLDA    ;IS IT '.LDA' ?
2873 CA8F28             JZ      LOLDA   ;YES !
                ; LOAD AN "IMAGE" FILE :
2876 CD8834             CALL    GCMA    ;SKIP COMMA IF IT'S THERE
2879 CDF634             CALL    GN2Z    ;GET AUX. LOAD ADDRESS
287C CDE726             CALL    CKEND   ;CHECK FOR END OF LINE
287F C0                 RNZ             ;NO: ERROR!
2880 01FFFF     RF1:    LXI     B,0FFFFH ;SET MAX. BYTE COUNT
2883 AF         RF2:    XRA     A       ;SET OPEN TYPE CODE
2884 CD862C             CALL    OPENX   ;OPEN THE FILE
2887 D8                 RC              ;ERROR!
2888 CDA32E             CALL    READ    ;READ THE FILE
288B D8                 RC              ;ERROR!
288C C35928             JMP     CLX     ;CLOSE FILE & EXIT!

                ; LOAD A '.LDA' FILE :
288F CD8629     LOLDA:  CALL    SULD    ;SETUP FOR '.LDA' LOAD
2892 CDE726             CALL    CKEND   ;END OF LINE ?
2895 CACD28             JZ      LLDA    ;YES: LOAD ALL AT CORRECT ADDRESS !
2898 CD8834             CALL    GCMA    ;SKIP COMMA IF THERE
289B CDF634             CALL    GN2Z    ;GET 'LOWEST ADDRESS TO LOAD'
289E D2EB26             JNC     ERSYN   ;SYNTAX ERROR !
28A1 EB                 XCHG
28A2 229D81             SHLD    XFHAN   ;SAVE 'LOWEST ADDRESS TO LOAD'
```

```
28A5 2100A0          LXI     H,0A000H
28A8 229F81          SHLD    XFFCN     ;SET DEFAULT 'LOWEST MEMORY ADDRESS'
28AB EB              XCHG
28AC CDE726          CALL    CKEND     ;END OF LINE ?
28AF CACD28          JZ      LLDA      ;YES: USE DEFAULT MEMORY SPECS !
28B2 CDA42C          CALL    GMPRM     ;GET MEMORY PARAMS
28B5 D8              RC                ;ERROR !
28B6 CDE726          CALL    CKEND     ;END OF LINE ?
28B9 C0              RNZ               ;NO: ERROR !
28BA 78              MOV     A,B       ;TEST BYTE COUNT ...
28BB B1              ORA     C
28BC CAC928          JZ      LOL1      ;ZERO: USE DEFAULT 'HIGHEST' !
28BF 60              MOV     H,B       ;COPY BYTE COUNT ...
28C0 69              MOV     L,C
28C1 2B              DCX     H         ;ADJUST IT
28C2 19              DAD     D         ;COMPUTE END ADDRESS
28C3 060F            MVI     B,EIVP
28C5 D8              RC                ;ERROR !
28C6 22A181          SHLD    XFBLK     ;SAVE 'HIGHEST MEMORY ADDRESS'
28C9 EB      LOL1:   XCHG
28CA 229F81          SHLD    XFFCN     ;SAVE 'LOWEST MEMORY ADDRESS'

28CD 2A9D81  LLDA:   LHLD    XFHAN     ;GET 'LOWEST ADDRESS TO LOAD'
28D0 EB              XCHG
28D1 2A9F81          LHLD    XFFCN     ;GET 'LOWEST MEMORY ADDRESS'
28D4 CD5934          CALL    SUBHD     ;COMPUTE OFFSET
28D7 229D81          SHLD    XFHAN     ;SAVE OFFSET
28DA AF              XRA     A         ;SET OPEN TYPE
28DB CD862C          CALL    OPENX     ;OPEN THE FILE
28DE D8              RC                ;ERROR !
28DF CDC630          CALL    RWSEQI    ;INITIALIZE FOR SEQUENTIAL INPUT
28E2 EB              XCHG
28E3 210000          LXI     H,0       ;INITIAL START ADDRESS
28E6 220A81  LOL3:   SHLD    FSAD      ;STORE START ADDRESS
28E9 EB              XCHG
28EA CD2C32  LOL4:   CALL    GTBYT     ;GET A BYTE
28ED DA4729          JC      EOFOK     ;ERROR OR EOF !
28F0 4F              MOV     C,A       ;LOBYTE OF BYTE COUNT
28F1 CD2C32          CALL    GTBYT
28F4 DA4E29          JC      LER1      ;ERROR !
28F7 47              MOV     B,A       ;HIBYTE OF BYTE COUNT
28F8 B1              ORA     C         ;IS IT ZERO ?
28F9 CAEA28          JZ      LOL4      ;YES: EMPTY RECORD !
28FC 0B              DCX     B
28FD 78              MOV     A,B
28FE B1              ORA     C         ;WAS IT ONE ?
28FF CA5229          JZ      LER2      ;YES: FILE FORMAT ERROR !
2902 CD2C32          CALL    GTBYT
2905 DA4E29          JC      LER1      ;ERROR !
2908 5F              MOV     E,A       ;LOBYTE OF LOAD ADDRESS
2909 CD2C32          CALL    GTBYT
290C DA4E29          JC      LER1      ;ERROR !
290F 57              MOV     D,A       ;HIGH BYTE OF LOAD ADDRESS
2910 0B              DCX     B
2911 78              MOV     A,B
2912 B1              ORA     C         ;ANY MORE ?
2913 EB              XCHG
2914 CAE628          JZ      LOL3      ;NO: STORE START ADDRESS & LOOK FOR MORE !
```

```
2917 EB                 XCHG
2918 E5                 PUSH    H           ;SAVE FPB PTR
2919 2A9D81             LHLD    XFHAN       ;GET OFFSET
291C 19                 DAD     D           ;COMPUTE ACTUAL LOAD ADDRESS
291D EB                 XCHG
291E E1                 POP     H           ;RESTORE FPB PTR
291F CD2C32     LOL5:   CALL    GTBYT       ;GET NEXT DATA BYTE
2922 DA4E29             JC      LER1        ;ERROR !
2925 E5                 PUSH    H           ;SAVE FPB PTR
2926 F5                 PUSH    PSW         ;SAVE THE BYTE
2927 2A9F81             LHLD    XFFCN       ;GET LOW LIMIT
292A CD5334             CALL    CMPDH       ;LOAD IT ?
292D DA3629             JC      LOL6        ;NO !
2930 2AA181             LHLD    XFBLK       ;GET HIGH LIMIT
2933 CD4D34             CALL    CMPHD       ;LOAD IT ?
2936 E1         LOL6:   POP     H           ;GET BYTE BACK ...
2937 7C                 MOV     A,H         ;... INTO A
2938 E1                 POP     H           ;RESTORE FPB PTR
2939 DA3D29             JC      LOL7        ;DON'T LOAD THIS BYTE !
293C 12                 STAX    D           ;LOAD THE BYTE INTO MEMORY
293D 13         LOL7:   INX     D           ;INCREMENT LOAD ADDRESS
293E 0B                 DCX     B           ;COUNT THE BYTE
293F 78                 MOV     A,B
2940 B1                 ORA     C           ;ANY MORE ?
2941 C21F29             JNZ     LOL5        ;YES: LOOP !
2944 C3EA28             JMP     LOL4        ;NO: LOOK FOR NEXT RECORD !
2947 3F         EOFOK:  CMC                 ;CLEAR <C> STATUS BIT
2948 F5                 PUSH    PSW         ;SAVE STATUS
2949 CD5928             CALL    CLX         ;CLOSE & TURN MOTOR OFF
294C F1                 POP     PSW         ;RESTORE STATUS
294D C8                 RZ                  ;END OF FILE: O.K. !
294E 0630       LER1:   MVI     B,EFRD
2950 37                 STC                 ;INDICATE ERROR
2951 C0                 RNZ                 ;READ ERROR !
2952 064B       LER2:   MVI     B,EBLF
2954 37                 STC
2955 C9                 RET                 ;FORMAT ERROR !

                ; RUN A PROGRAM :
2956 CD7430     RUN00:  CALL    PPFSP       ;GET FILE SPECIFICATION
2959 D8                 RC                  ;ERROR!
295A CDE726             CALL    CKEND       ;CHECK FOR END OF LINE
295D C0                 RNZ                 ;NO: ERROR!
295E CDAB2A             CALL    QLDA        ;IS IT '.LDA' ?
2961 C26D29             JNZ     RUN01       ;NO !
2964 CD8629             CALL    SULD        ;SETUP FOR '.LDA' LOAD
2967 CDCD28             CALL    LLDA        ;LOAD THE FILE
296A C37C29             JMP     RUN02
296D 01572D     RUN01:  LXI     B,PTYP
2970 CDAE2A             CALL    QTYP        ;IS IT '.PRG' ?
2973 0648               MVI     B,EIVT
2975 C0                 RNZ                 ;NO: INVALID TYPE !
                ; RUN AN "IMAGE" ( TYPE '.PRG' ) PROGRAM :
2976 110000             LXI     D,0         ;SET AUX. LOAD ADDRESS
2979 CD8028             CALL    RF1         ;READ THE FILE
297C D8         RUN02:  RC                  ;ERROR!
297D 2A0A81             LHLD    FSAD        ;GET START ADDRESS
```

```
2980 7C              MOV     A,H
2981 B5              ORA     L
2982 0621            MVI     B,ENSA  ;SET ERROR CODE
2984 C8              RZ              ;NO START ADDRESS!
2985 E9              PCHL            ;GO TO THE PROGRAM!

2986 E5      SULD:   PUSH    H
2987 210000          LXI     H,0
298A 229D81          SHLD    XFHAN   ;SET 'LOWEST ADDRESS TO LOAD'
298D 229F81          SHLD    XFFCN   ;SET 'LOWEST MEMORY ADDRESS'
2990 2B              DCX     H
2991 22A181          SHLD    XFBLK   ;SET 'HIGHEST MEMORY ADDRESS'
2994 E1              POP     H
2995 C9              RET

2996 CDDE2F  DEV00:  CALL    PDV     ;PARSE DEVICE NAME
2999 D8              RC              ;ERROR!
299A CDE726          CALL    CKEND   ;CHECK FOR END OF LINE
299D C0              RNZ             ;NO: ERROR!
299E 21422D          LXI     H,MS4
29A1 CDF433          CALL    OSTR
29A4 CDD42C          CALL    PRDEV   ;PRINT DEVICE NAME
29A7 CD8B33          CALL    CRLF
29AA 21F080          LXI     H,DFDV  ;POINT TO DEFAULT DEVICE AREA
29AD 73              MOV     M,E     ;STORE ...
29AE 23              INX     H       ;... DEVICE ...
29AF 72              MOV     M,D     ;... NAME
29B0 23              INX     H
29B1 71              MOV     M,C     ;STORE UNIT NUMBER
29B2 0600            MVI     B,0     ;INDICATE NO ERRORS
29B4 C9              RET             ;EXIT!

29B5 01532D  DEL00:  LXI     B,NTYP  ;SET DEFAULT TYPE
29B8 CD7730          CALL    PFSPC   ;GET FILE SPECIFICATION
29BB D8              RC              ;ERROR!
29BC C8              RZ              ;NO VERSION: ERROR!
29BD CDE726          CALL    CKEND   ;CHECK FOR END OF LINE
29C0 C0              RNZ             ;NO: ERROR!
29C1 AF              XRA     A       ;SET OPEN TYPE CODE
29C2 CD862C          CALL    OPENX   ;OPEN THE FILE
29C5 D8              RC              ;ERROR!
29C6 2A0D81          LHLD    FDBK    ;GET DIR. BLK. NUMBER ...
29C9 2600            MVI     H,0     ;...
29CB 221581          SHLD    FBLK    ;STORE BLOCK NUMBER
29CE 211F81          LXI     H,DBF+2 ;SETUP TO ...
29D1 111181          LXI     D,FHAN  ;... CONTINUE DIR ...
29D4 0606            MVI     B,6     ;... SCAN
29D6 CD862D  DEL01:  CALL    GNDE    ;GET NEXT DIRECTORY ENTRY
29D9 D8              RC              ;ERROR!
29DA FE01            CPI     TFREE   ;IS THIS THE 'FREE' ENTRY ?
29DC C2D629          JNZ     DEL01   ;NO: KEEP LOOKING FOR IT!
29DF 110B00          LXI     D,FSBK-FATR ;POINT TO FSBK ...
29E2 19              DAD     D       ;... IN DIR ENTRY
29E3 5E              MOV     E,M     ;GET STARTING BLOCK ...
29E4 23              INX     H       ;... NUMBER OF ...
29E5 56              MOV     D,M     ;... THE 'FREE'
29E6 D5              PUSH    D       ;SAVE START BLOCK OF FREE
```

```
29E7 CD772C            CALL    CPYDV   ;COPY DEVICE INFO
29EA 2A0381            LHLD    FSBK    ;GET START BLK OF FILE
29ED 221581            SHLD    FBLK    ;SET DESTINATION START BLK
29F0 EB                XCHG
29F1 2A0581            LHLD    FSIZ    ;GET FILE SIZE
29F4 19                DAD     D       ;COMPUTE SOURCE START BLOCK
29F5 22A181            SHLD    XFBLK   ;SET SOURCE START BLOCK
29F8 EB                XCHG            ;D&E = SOURCE START BLOCK
29F9 E1                POP     H       ;GET START BLOCK OF FREE
29FA CD5934            CALL    SUBHD   ;COMPUTE NO. OF BLKS TO MOVE
29FD EB                XCHG            ;PUT RESULT IN D&E
29FE CD1A2C            CALL    MCHNK   ;MOVE THE DATA
2A01 DA9B2A            JC      DELER   ;ERROR DURING MOVE!
2A04 2A0D81            LHLD    FDBK    ;GET DIR. BLK. NUMBER ...
2A07 2600              MVI     H,0     ;...
2A09 221581            SHLD    FBLK    ;SET BLOCK NO. FOR XFER
2A0C EB                XCHG
2A0D 3A1E81            LDA     MDBLK   ;GET MAX. DIR BLK NO.
2A10 3C                INR     A       ;NO. OF DIR BLOCKS
2A11 6F                MOV     L,A
2A12 62                MOV     H,D
2A13 CD5934            CALL    SUBHD   ;COMPUTE BLOCKS TO XFER
2A16 0E80              MVI     C,80H
2A18 CDBA35            CALL    BK2BC   ;CHANGE TO BYTE COUNT
2A1B 221981            SHLD    FXBC    ;STORE BYTE COUNT
2A1E 211181            LXI     H,FHAN  ;POINT TO FHAN
2A21 CDFB2E            CALL    RD      ;READ THE DIRECTORY
2A24 DA9B2A            JC      DELER   ;DIR. READ ERROR!
2A27 218060            LXI     H,DSBUF+2+6*21 ;PNT PAST 1ST BLOCK
2A2A CD9E2A            CALL    CENTR   ;COMPUTE ENTRY PNTR
2A2D E5        DEL06:  PUSH    H       ;SAVE PNTR TO DESTINATION SLOT
2A2E 111500            LXI     D,21
2A31 19                DAD     D       ;POINT TO SOURCE SLOT
2A32 EB                XCHG
2A33 E1                POP     H       ;RESTORE DEST. SLOT PNTR
2A34 E5                PUSH    H       ;RESAVE IT
2A35 78                MOV     A,B     ;GET SLOT NUMBER
2A36 3D                DCR     A       ;IS DEST. SLOT LAST SLOT IN BLK ?
2A37 C23C2A            JNZ     DEL07   ;NO: O.K.!
2A3A 13                INX     D       ;ADJUST SOURCE ...
2A3B 13                INX     D       ;... SLOT POINTER
2A3C C5        DEL07:  PUSH    B       ;SAVE B&C
2A3D 0615              MVI     B,21    ;SET COUNT
2A3F CD4434            CALL    MOVHD   ;MOVE THE ENTRY
2A42 C1                POP     B       ;RESTORE B&C
2A43 E1                POP     H       ;GET DEST. SLOT PNTR
2A44 E5        DEL08:  PUSH    H       ;SAVE DEST. SLOT POINTER
2A45 110B00            LXI     D,FSBK-FATR
2A48 19                DAD     D       ;POINT TO FSBK IN SLOT
2A49 5E                MOV     E,M     ;GET STARTING ...
2A4A 23                INX     H       ;... BLOCK ...
2A4B 56                MOV     D,M     ;... NUMBER
2A4C E5                PUSH    H       ;SAVE POINTER
2A4D 2A0581            LHLD    FSIZ    ;GET SIZE OF DELETED FILE
2A50 EB                XCHG
2A51 CD5934            CALL    SUBHD   ;COMPUTE NEW STARTING BLK NO.
2A54 EB                XCHG
2A55 E1                POP     H       ;RESTORE POINTER
```

```
2A56 72                 MOV     M,D     ;STORE ADJUSTED ...
2A57 2B                 DCX     H       ;... STARTING ...
2A58 73                 MOV     M,E     ;... BLOCK NUMBER
2A59 E1                 POP     H       ;GET DEST. SLOT PNTR
2A5A 7E                 MOV     A,M     ;GET TYPE CODE
2A5B FE01               CPI     TFREE   ;IS THIS THE 'FREE' ?
2A5D CA6F2A             JZ      DEL09   ;YES!
2A60 111500             LXI     D,21
2A63 19                 DAD     D       ;STEP TO NEXT SLOT
2A64 05                 DCR     B       ;WAS THIS LAST SLOT IN BLOCK ?
2A65 C22D2A             JNZ     DEL06   ;NO: GO DO NEXT SLOT!
2A68 23                 INX     H       ;ADJUST ...
2A69 23                 INX     H       ;... SLOT POINTER
2A6A 0606               MVI     B,6     ;RESET SLOT COUNTER
2A6C C32D2A             JMP     DEL06   ;GO DO NEXT SLOT!
2A6F 110D00     DEL09:  LXI     D,FSIZ-FATR
2A72 19                 DAD     D       ;POINT TO SIZE OF 'FREE'
2A73 5E                 MOV     E,M     ;GET SIZE ...
2A74 23                 INX     H       ;... OF ...
2A75 56                 MOV     D,M     ;... 'FREE'
2A76 E5                 PUSH    H       ;SAVE POINTER
2A77 2A0581             LHLD    FSIZ    ;GET SIZE OF DELETED FILE
2A7A 19                 DAD     D       ;COMPUTE NEW 'FREE' BLOCKS
2A7B EB                 XCHG
2A7C E1                 POP     H       ;RESTORE POINTER
2A7D 72                 MOV     M,D     ;STORE NEW ...
2A7E 2B                 DCX     H       ;... SIZE OF ...
2A7F 73                 MOV     M,E     ;... 'FREE'
2A80 110800             LXI     D,FDBK-FSIZ
2A83 19                 DAD     D       ;POINT TO NEXT SLOT
2A84 05                 DCR     B       ;WAS THIS LAST SLOT ?
2A85 C28A2A             JNZ     DEL10   ;NO: O.K.!
2A88 23                 INX     H       ;ADJUST ...
2A89 23                 INX     H       ;... SLOT POINTER
2A8A 3600       DEL10:  MVI     M,0     ;MARK END OF DIRECTORY
2A8C 211181             LXI     H,FHAN  ;POINT TO FHAN
2A8F CDF82E             CALL    WR      ;WRITE DIRECTORY
2A92 DA9B2A             JC      DELER   ;ERROR!
2A95 CD8D2B             CALL    ERAS    ;ERASE PAGE
2A98 C39927             JMP     DIR01   ;GO LIST DIRECTORY
2A9B 063C       DELER:  MVI     B,EDEL  ;SET ERROR CODE
2A9D C9                 RET

2A9E 11EBFF     CENTR:  LXI     D,-21
2AA1 3A0E81             LDA     FDEN    ;GET ENTRY NO. OF FILE
2AA4 47                 MOV     B,A     ;COPY IT
2AA5 19         CEN01:  DAD     D       ;BACK UP ONE ENTRY
2AA6 3D                 DCR     A       ;ENOUGH ?
2AA7 C2A52A             JNZ     CEN01   ;NO: DO IT AGAIN!
2AAA C9                 RET             ;RETURN


2AAB 015A2D     GLDA:   LXI     B,LTYP
2AAE E5         GTYP:   PUSH    H       ;SAVE H
2AAF 21FF80             LXI     H,FTYP
2AB2 1E03               MVI     E,3     ;SET COUNTER
2AB4 0A         CMT1:   LDAX    B       ;GET BYTE
2AB5 03                 INX     B
```

```
2AB6 BE              CMP     M          ;COMPARE
2AB7 23              INX     H
2AB8 C2BF2A          JNZ     CMT2       ;DOES NOT MATCH !
2ABB 1D              DCR     E          ;CHECKED ALL THREE ?
2ABC C2B42A          JNZ     CMT1       ;NO: CHECK NEXT !
2ABF E1        CMT2: POP     H          ;RESTORE H
2AC0 C9              RET                ;<Z>: MATCH, <NZ>: NO MATCH !

2AC1 22A181    RENOO: SHLD     XFBLK      ;SAVE STRING PNTR
2AC4 CD6E30          CALL    PNFSP      ;GET OLD NAME
2AC7 D8              RC                 ;ERROR!
2AC8 CD0C2C          CALL    GETTO      ;GET "TO"
2ACB C0              RNZ                ;ERROR!
2ACC CD772C          CALL    CPYDV      ;COPY DEVICE INFO
2ACF CD8730          CALL    PCFSP      ;GET NEW NAME
2AD2 D8              RC                 ;ERROR!
2AD3 CDE726          CALL    CKEND      ;CHECK FOR END OF LINE
2AD6 C0              RNZ                ;NO: ERROR!
2AD7 2A9D81          LHLD    XFHAN      ;FORCE ...
2ADA 221181          SHLD    FHAN       ;... SAME ...
2ADD 3AA081          LDA     XFDRV      ;... DEVICE ...
2AE0 321481          STA     FDRV       ;...
2AE3 CD002C          CALL    OPOX       ;OPEN THE FILE
2AE6 D8              RC                 ;ERROR!
2AE7 11F980          LXI     D,FNAM     ;POINT TO NEW NAME
2AEA 21A381          LXI     H,XFBUF    ;POINT TO TEMP AREA
2AED 060A            MVI     B,10       ;SET COUNT
2AEF CD4434          CALL    MOVHD      ;COPY NEW NAME
2AF2 2AA181          LHLD    XFBLK      ;GET STRNG PNTR TO OLD NAME
2AF5 11F780          LXI     D,FPB      ;POINT TO FPB
2AF8 CD6E30          CALL    PNFSP      ;GET OLD NAME AGAIN
2AFB AF              XRA     A          ;SET OPEN TYPE CODE
2AFC CD862C          CALL    OPENX      ;OPEN OLD FILE
2AFF D8              RC                 ;ERROR!
2B00 CDB52F          CALL    GODBK      ;GET DIR BLK
2B03 D8              RC                 ;ERROR!
2B04 219E81          LXI     H,DBF+2+6*21+1
2B07 CD9E2A          CALL    CENTR      ;COMPUTE ENTRY PNTR
2B0A 11A381          LXI     D,XFBUF    ;POINT TO TEMP AREA
2B0D 060A            MVI     B,10       ;SET COUNT
2B0F CD4434          CALL    MOVHD      ;MOVE IN NEW NAME
2B12 2A0D81          LHLD    FDBK       ;GET DIR BLK NO.
2B15 2600            MVI     H,0
2B17 221581          SHLD    FBLK       ;SET BLOCK NUMBER
2B1A 219D81          LXI     H,DBFE     ;SET END PNTR
2B1D C3752F          JMP     WRDIR      ;GO WRITE DIR BLK !

2B20 22A181    COPOO: SHLD     XFBLK      ;SAVE STRING PNTR
2B23 CD6E30          CALL    PNFSP      ;GET SOURCE NAME
2B26 D8              RC                 ;ERROR!
2B27 CD0C2C          CALL    GETTO      ;GET "TO"
2B2A C0              RNZ                ;ERROR!
2B2B 22AB81          SHLD    TMP1       ;SAVE STRING PNTR
2B2E CD8730          CALL    PCFSP      ;GET DESTINATION NAME
2B31 D2382B          JNC     COPO1      ;O.K. !
2B34 3E15            MVI     A,EMFN
```

```
2B36 B8                 CMP     B               ;MISSING FILE NAME ?
2B37 C0                 RNZ                     ;NO: ERROR!
2B38 CDE726     COPO1:  CALL    CKEND           ;CHECK FOR END OF LINE
2B3B C0                 RNZ                     ;NO: ERROR!
2B3C 2AA181             LHLD    XFBLK           ;GET SOURCE PNTR
2B3F CD6E30             CALL    PNFSP           ;GET SOURCE NAME
2B42 CD772C             CALL    CPYDV           ;COPY DEVICE INFO
2B45 AF                 XRA     A               ;SET OPEN TYPE CODE
2B46 CD862C             CALL    OPENX           ;OPEN SOURCE FILE
2B49 D8                 RC                      ;ERROR!
2B4A 110381             LXI     D,FSBK          ;SET SOURCE PNTR
2B4D 21A181             LXI     H,XFBLK         ;SET DESTINATION PNTR
2B50 060A               MVI     B,FDBK-FSBK     ;SET COUNT
2B52 CD4434             CALL    MOVHD           ;COPY FILE INFO
2B55 11F780             LXI     D,FPB           ;POINT TO FPB
2B58 2AAB81             LHLD    TMP1            ;GET STRING PNTR
2B5B CD8730             CALL    PCFSP           ;GET DESTINATION NAME
2B5E 3E01               MVI     A,FNEW          ;SET OPEN TYPE CODE
2B60 CD862C             CALL    OPENX           ;OPEN DESTINATION FILE
2B63 D8                 RC                      ;ERROR!
2B64 2AA381             LHLD    XFBLK+FSIZ-FSBK ;GET SOURCE FILE SIZE
2B67 220F81             SHLD    FAUX            ;SET SIZE FOR CLOSE
2B6A EB                 XCHG
2B6B 2A0581             LHLD    FSIZ            ;GET "FREE" SIZE
2B6E CD4D34             CALL    CMPHD           ;SEE IF IT'LL FIT
2B71 0636               MVI     B,EWSZ          ;SET ERROR CODE
2B73 D8                 RC                      ;TOO BIG: ERROR!
2B74 D5                 PUSH    D               ;SAVE FILE SIZE
2B75 11A581             LXI     D,XFBLK+FLBC-FSBK ;SET SOURCE PNTR
2B78 210781             LXI     H,FLBC          ;SET DESTINATION PNTR
2B7B 0606               MVI     B,FDBK-FLBC     ;SET COUNT
2B7D CD4434             CALL    MOVHD           ;COPY FILE INFO BACK
2B80 D1                 POP     D               ;GET FILE SIZE BACK
2B81 CD1A2C             CALL    MCHNK           ;COPY THE FILE
2B84 0645               MVI     B,ECOP          ;SET ERROR CODE
2B86 D8                 RC                      ;ERROR!
2B87 21F780             LXI     H,FPB           ;POINT TO FPB
2B8A C35928             JMP     CLX             ;GO CLOSE FILE & EXIT!

2B8D CDD03F     ERAS:   CALL    SAVE            ;  SAVE REGISTERS
2B90 C33638             JMP     ERASE           ;  ERASE SCREEN AND RETURN

2B93 CDDE2F     DUPOO:  CALL    PDV             ;  GET DEVICE NAME
2B96 D8                 RC                      ;  ERROR:  INVALID DEVICE
2B97 C8                 RZ                      ;  ERROR:  NO DEVICE NAME
2B98 CD0C2C             CALL    GETTO           ;  GET 'TO'
2B9B C0                 RNZ                     ;  ERROR:
2B9C 118381             LXI     D,XFHAN+FPB-FHAN
2B9F CDDE2F             CALL    PDV             ;  GET DESTINATION DEVICE NAME
2BA2 D8                 RC                      ;  ERROR:  INVALID DEVICE
2BA3 C8                 RZ                      ;  ERROR:  NO DEVICE NAME
2BA4 CDE726             CALL    CKEND           ;  CHECK FOR END OF LINE
2BA7 C0                 RNZ                     ;  ERROR:  NOT END OF LINE

2BA8 211181             LXI     H,FHAN          ;  POINT TO FHAN
2BAB 3EFF               MVI     A,-1            ;  SET 'GET SIZE' CODE
2BAD CDFC2E             CALL    CHDLR           ;  CALL HANDLER
2BB0 063F               MVI     B,EDUP
```

```
2BB2 D8                 RC              ; ERROR: DUPLICATE ERROR

2BB3 E5                 PUSH    H       ; SAVE SOURCE DEVICE SIZE
2BB4 219D81             LXI     H,XFHAN ; POINT TO XFHAN
2BB7 3EFF               MVI     A,-1
2BB9 CDFC2E             CALL    CHDLR   ; GET SIZE
2BBC D1                 POP     D       ; RESTORE SURCE SIZE
2BBD 063F               MVI     B,EDUP
2BBF D8                 RC.             ; ERROR: DUPLICATE ERROR

2BC0 CD4D34             CALL    CMPHD   ; ARE THE SIZES THE SAME?
2BC3 0642               MVI     B,ESIZ
2BC5 C0                 RNZ             ; ERROR:  NOT SAME SIZE

2BC6 22AB81             SHLD    TMP1    ; SAVE SIZE

2BC9 AF                 XRA     A
2BCA 32F980             STA     FNAM    ; SET INVALID NAME
2BCD CD002C             CALL    OPOX    ; OPEN FILE
2BD0 DAD92B             JC      DUP02   ; ERROR: DUPLICATE ALL

2BD3 2A0381             LHLD    FSBK    ; GET START BLK NO. OF 'FREE'
2BD6 22AB81             SHLD    TMP1    ; SAVE AS NUMBER OF BLOCKS TO DUPLICATE

2BD9 2A9D81   DUP02:    LHLD    XFHAN   ; SWITCH DEVICES
2BDC 3AA081             LDA     XFDRV
2BDF 47                 MOV     B,A
2BE0 CD772C             CALL    CPYDV
2BE3 221181             SHLD    FHAN
2BE6 78                 MOV     A,B
2BE7 321481             STA     FDRV

2BEA 2AAB81             LHLD    TMP1    ; GET NO. OF BLKS TO DUPLICATE
2BED EB                 XCHG
2BEE 210000             LXI     H,0
2BF1 22A181             SHLD    XFBLK   ; SET STARTING BLOCK NUMBERS
2BF4 221581             SHLD    FBLK
2BF7 CD1A2C             CALL    MCHNK   ; MOVE THE DATA
2BFA 063F               MVI     B,EDUP
2BFC D8                 RC              ; ERROR:
2BFD 0600               MVI     B,0
2BFF C9                 RET             ; EXIT

2C00 3E01     OPOX:     MVI     A,FNEW  ;SET OPEN TYPE CODE
2C02 CD862C             CALL    OPENX   ;OPEN THE FILE
2C05 D0                 RNC             ;NO ERRORS: RETURN
2C06 3E2D               MVI     A,EDRF  ;IS IT ...
2C08 A8                 XRA     B       ;... DIRECTORY FULL ?
2C09 C8                 RZ              ;YES: O.K. RETURN!
2C0A 37                 STC             ;INDICATE ERROR
2C0B C9                 RET             ;RETURN


2C0C 0609     GETTO:    MVI     B,ESYN  ;SET ERROR CODE
2C0E CD6034             CALL    SPNOR   ;GET 1ST NON-SPACE
2C11 FE54               CPI     'T'     ;IS IT 'T' ?
2C13 C0                 RNZ             ;NO: ERROR!
2C14 23                 INX     H
```

```
2C15 7E              MOV     A,M       ;GET NEXT CHARACTER
2C16 23              INX     H
2C17 FE4F            CPI     'O'       ;IS IT 'O' ?
2C19 C9              RET               ;RETURN

2C1A 210060  MCHNK:  LXI     H,DSBUF   ;POINT TO DISPLAY BUFFER
2C1D 22A381          SHLD    XFBUF     ;SET BUFFER ...
2C20 221781          SHLD    FBUF      ;... POINTERS
2C23 D5              PUSH    D         ;SAVE NO. OF BLOCKS
2C24 3EFE            MVI     A,-2      ;SET "ADD USER" CODE
2C26 CD692C          CALL    CTWO      ;CALL HANDLERS
2C29 D1              POP     D         ;GET NO. OF BLOCKS
2C2A CD8D2B  MCHO1:  CALL    ERAS      ;ERASE PAGE
2C2D 7B              MOV     A,E       ;ANY ...
2C2E B2              ORA     D         ;... LEFT ?
2C2F CA672C          JZ      STWO      ;NO: FINISHED!
2C32 EB              XCHG
2C33 112000          LXI     D,DSBUFS/128    ;SET BUFFER SIZE
2C36 CD5934          CALL    SUBHD     ;SUB. FROM NO. LEFT
2C39 EB              XCHG
2C3A D2412C          JNC     MCHO2     ;DO FULL BUFFER!
2C3D 19              DAD     D         ;ADJUST COUNT TO ORIG.
2C3E 110000          LXI     D,0       ;SET NONE LEFT
2C41 0E80    MCHO2:  MVI     C,80H     ;SET LAST BLK B.C.
2C43 CDBA35          CALL    BK2BC     ;CONVERT TO BYTE COUNT
2C46 22A581          SHLD    XFXBC     ;SET BYTE ...
2C49 221981          SHLD    FXBC      ;... COUNTS
2C4C D5              PUSH    D         ;SAVE REMAINING BLK COUNT
2C4D 219D81          LXI     H,XFHAN   ;POINT TO SOURCE HPB
2C50 CDFB2E          CALL    RD        ;READ
2C53 D1              POP     D         ;RESTORE REMAINING BLK COUNT
2C54 D8              RC                ;ERROR!
2C55 22A181          SHLD    XFBLK     ;UPDATE THE BLOCK NUMBER
2C58 D5              PUSH    D         ;SAVE REMAINING BLK COUNT
2C59 211181          LXI     H,FHAN    ;POINT TO DESTINATION HPB
2C5C CDF82E          CALL    WR        ;WRITE
2C5F D1              POP     D         ;RESTORE REMAINING BLK COUNT
2C60 D8              RC                ;ERROR!
2C61 221581          SHLD    FBLK      ;UPDATE THE BLOCK NUMBER
2C64 C32A2C          JMP     MCHO1     ;LOOP!

2C67 3EFD    STWO:   MVI     A,-3      ;SET "SUB USER" FUNCTION
2C69 F5      CTWO:   PUSH    PSW       ;SAVE FCN CODE
2C6A 219D81          LXI     H,XFHAN   ;POINT TO SOURCE HPB
2C6D CDFC2E          CALL    CHDLR     ;CALL HANDLER
2C70 F1              POP     PSW       ;RESTORE FCN CODE
2C71 211181          LXI     H,FHAN    ;POINT TO DESTINATION HPB
2C74 C3FC2E          JMP     CHDLR     ;CALL HANDLER & RETURN

2C77 E5      CPYDV:  PUSH    H         ;SAVE H&L
2C78 2A1181          LHLD    FHAN      ;GET HANDLER ADDRESS
2C7B 229D81          SHLD    XFHAN     ;SAVE IN AUX. HPB
2C7E E1              POP     H         ;RESTORE H&L
2C7F 3A1481          LDA     FDRV      ;GET DRIVE NO.
2C82 32A081          STA     XFDRV     ;SAVE IN AUX. HPB
2C85 C9              RET               ;RETURN
```

```
2C86 21F780    OPENX:  LXI     H,FPB     ;POINT TO FPB
2C89 C5        OPENY:  PUSH    B
2C8A D5                PUSH    D
2C8B F5                PUSH    PSW       ;SAVE OPEN TYPE CODE
2C8C E5                PUSH    H
2C8D 111A00            LXI     D,FHAN-FPB
2C90 19                DAD     D         ;POINT TO FHAN
2C91 3EFE              MVI     A,-2      ;SET "ADD USER" FUNCTION
2C93 CDFC2E            CALL    CHDLR
2C96 E1                POP     H         ;RESTORE FPB PNTR
2C97 F1                POP     PSW       ;RESTORE OPEN TYPE CODE
2C98 77                MOV     M,A       ;STORE OPEN TYPE CODE
2C99 CDAB2D            CALL    OPEN      ;OPEN THE FILE
2C9C D1                POP     D
2C9D D2A22C            JNC     OPX01
2CA0 D1                POP     D
2CA1 C9                RET
2CA2 C1        OPX01:  POP     B
2CA3 C9                RET               ;RETURN

2CA4 CD8834    GMPRM:  CALL    GCMA      ;SKIP COMMA IF ANY
2CA7 CDF634            CALL    GN2Z      ;GET ONE NUMBER
2CAA D2D02C            JNC     GM02      ;NO NUMBER: ERROR!
2CAD D5                PUSH    D         ;SAVE 1ST NUMBER
2CAE CD6034            CALL    SPNOR     ;SKIP SPACES
2CB1 D62D              SUI     '-'       ;CHECK FOR '-'
2CB3 A7                ANA     A         ;CLEAR C-FLAG
2CB4 F5                PUSH    PSW       ;SAVE FLAGS
2CB5 23                INX     H
2CB6 CABD2C            JZ      GM03
2CB9 2B                DCX     H
2CBA CD8834            CALL    GCMA      ;SKIP COMMA IF NOT '-'
2CBD CDF634    GM03:   CALL    GN2Z      ;GET 2ND NUMBER
2CC0 CD8834            CALL    GCMA      ;SKIP COMMA IF THERE
2CC3 F1                POP     PSW       ;RESTORE FLAGS
2CC4 E3                XTHL
2CC5 EB                XCHG
2CC6 C2CD2C            JNZ     GM01      ;WAS NOT '-'
2CC9 CD5934            CALL    SUBHD     ;COMPUTE BYTE COUNT
2CCC 23                INX     H
2CCD E3        GM01:   XTHL
2CCE C1                POP     B
2CCF D0                RNC               ;O.K.: EXIT!
2CD0 060F      GM02:   MVI     B,EIVP    ;SET ERROR CODE
2CD2 37                STC               ;INDICATE ERROR
2CD3 C9                RET               ;RETURN!

2CD4 211A00    PRDEV:  LXI     H,FHAN-FPB
2CD7 19                DAD     D         ;POINT TO FHAN
2CD8 5E                MOV     E,M       ;GET LOBYTE OF HANDLER PNTR
2CD9 23                INX     H
2CDA 56                MOV     D,M       ;GET HIBYTE OF HANDLER PNTR
2CDB 23                INX     H
2CDC 23                INX     H
2CDD 7E                MOV     A,M       ;GET UNIT NUMBER
2CDE C630              ADI     '0'       ;CONVERT TO ASCII
2CE0 4F                MOV     C,A       ;COPY UNIT NO. TO C
```

```
2CE1 EB                 XCHG
2CE2 23                 INX     H       ;POINT TO DEVICE NAME
2CE3 23                 INX     H
2CE4 23                 INX     H
2CE5 5E                 MOV     E,M     ;GET 1ST CHARACTER
2CE6 23                 INX     H
2CE7 56                 MOV     D,M     ;GET 2ND CHARACTER
2CE8 7B                 MOV     A,E     ;PRINT ...
2CE9 CD9233             CALL    LO      ;... 1ST CHAR.
2CEC 7A                 MOV     A,D     ;PRINT ...
2CED CD9233             CALL    LO      ;... 2ND CHAR.
2CF0 79                 MOV     A,C     ;PRINT ...
2CF1 CD9233             CALL    LO      ;... UNIT NUMBER
2CF4 C3B834             JMP     PCOLN   ;PRINT ':' AND RETURN!

2CF7 0A0A4449 MS1:      DB 10,10,'DIRECTORY ',239
2CFB 52454354
2CFF 4F525920
2D03 EF
2D04 0D0A0A   MS2:      DB 13,10,10
2D07 41545220           DB 'ATR  NAME TYPE VR SBLK SIZE LBC'
2D0B 204E414D
2D0F 45205459
2D13 50452056
2D17 52205342
2D1B 4C4B2053
2D1F 495A4520
2D23 4C4243
2D26 204C4144           DB ' LADR SADR',13,10,239
2D2A 52205341
2D2E 44520D0A
2D32 EF
2D33 203C4652 MS3:      DB ' <FREE SPACE> ',239
2D37 45452053
2D3B 50414345
2D3F 3E20EF
2D42 06034445 MS4:      DB 6,3,'DEFAULT DEVICE='
2D46 4641554C
2D4A 54204445
2D4E 56494345
2D52 3D
2D53 200602EF NTYP:     DB ' ',6,2,239

2D57 505247   PTYP:     DB      'PRG'
2D5A 4C4441   LTYP:     DB      'LDA'
2D5D 535243   STYP:     DB      'SRC'



              ; DIRECTORY ACCESS ROUTINES :

              ;+
              ; OPDIR - "OPEN" DIRECTORY FOR A SCAN.
              ;
```

```
                ;  INPUTS  - D&E:  PNTR TO FHAN IN FPB
                ;  OUTPUTS - <C>   :  DIRECTORY READ ERROR,
                ;                     OR NO VOLUME ENTRY
                ;            <NC>  :  NO ERRORS AND :
                ;              A   :  ATR BYTE OF THIS ENTRY (=41H)
                ;              B   :  INTERNALLY MAINTAINED ENTRY COUNTER
                ;              D&E :  PNTR TO FHAN IN FPB (UNCHANGED)
                ;              H&L :  PNTR TO "VOLUME" ENTRY IN DIRECTORY
                ;-

2D60 210400     OPDIR:  LXI     H,FBLK-FHAN
2D63 19                 DAD     D               ;POINT TO FBLK
2D64 D5                 PUSH    D               ;SAVE PNTR TO FHAN
2D65 11802D             LXI     D,DIP           ;POINT TO INITIAL PARAMETERS
2D68 0606               MVI     B,DIPS          ;SET BYTE COUNT
2D6A CD4434             CALL    MOVHD           ;MOVE PARAMETERS
2D6D E1                 POP     H               ;GET FHAN PNTR
2D6E E5                 PUSH    H               ;RESAVE IT
2D6F 3EFE               MVI     A,-2            ;SET "ADD USER" FUNCTION
2D71 CDFC2E             CALL    CHDLR           ;CALL HANDLER
2D74 D1                 POP     D               ;RESTORE PNTR TO FHAN
2D75 CD942D             CALL    LOO1            ;GO READ 1ST DIRECTORY BLOCK
2D78 D8                 RC                      ;ERROR!
2D79 FE41               CPI     41H             ;IS THIS VOLUME ENTRY ?
2D7B C8                 RZ                      ;YES: O.K.: EXIT!
2D7C 0612               MVI     B,ENVE          ;SET ERROR CODE
2D7E 37                 STC
2D7F C9                 RET                     ;ERROR EXIT!

2D80 0000       DIP:    DW      0               ;BLOCK NUMBER
2D82 1D81               DW      DBF             ;BUFFER POINTER
2D84 8000               DW      128             ;BYTE COUNT
     (0006)     DIPS    EQU     $-DIP

                ;+
                ;  GNDE  - GET NEXT DIRECTORY ENTRY
                ;
                ;  INPUTS  - B     :  INTERNALLY MAINTAINED ENTRY COUNTER
                ;            D&E   :  PNTR TO FHAN IN FPB
                ;            H&L   :  PNTR TO CURRENT DIRECTORY ENTRY
                ;  OUTPUTS - <C>   :  DIRECTORY READ ERROR
                ;            <Z>   :  END OF DIRECTORY
                ;          <NC,NZ> :  NO ERRORS AND :
                ;            A     :  ATTRIBUTE BYTE OF THIS ENTRY
                ;            B     :  UPDATED, MUST BE PRESERVED
                ;            D&E   :  PNTR TO FHAN IN FPB (UNCHANGED)
                ;            H&L   :  PNTR TO NEXT DIRECTORY ENTRY
                ;-

2D86 05         GNDE:   DCR     B               ;ANY MORE IN THIS BLOCK ?
2D87 C2A22D             JNZ     LOO2            ;YES!
2D8A 210400             LXI     H,FBLK-FHAN
2D8D 19                 DAD     D               ;POINT TO FBLK
2D8E 3A1E81             LDA     MDBLK           ;GET MAX. DIRECTORY BLOCK NUMBER
2D91 96                 SUB     M               ;SUBTRACT CURRENT BLOCK NUMBER
2D92 C8                 RZ                      ;THIS WAS LAST BLOCK!
2D93 34                 INR     M               ;BUMP BLOCK NUMBER
2D94 D5         LOO1:   PUSH    D               ;SAVE D&E
```

```
2D95 EB                 XCHG            ;PUT PNTR IN H&L
2D96 CDFB2E             CALL    RD      ;READ NEXT BLOCK
2D99 D1                 POP     D       ;RESTORE D&E
2D9A 060C               MVI     B,EDIR  ;SET ERROR CODE
2D9C D8                 RC              ;READ ERROR!
2D9D 0606               MVI     B,6     ;SET ENTRY COUNTER
2D9F 210A81             LXI     H,DBF+2-21 ;SET DIRECTORY PNTR
2DA2 D5         L002:   PUSH    D       ;SAVE D&E
2DA3 111500             LXI     D,21    ;ENTRY SIZE
2DA6 19                 DAD     D       ;STEP TO NEXT ENTRY
2DA7 D1                 POP     D       ;RESTORE D&E
2DA8 7E                 MOV     A,M     ;GET ATTRIBUTE BYTE
2DA9 A7                 ANA     A       ;TEST ATTRIBUTE BYTE
2DAA C9                 RET             ;RETURN, NO ERRORS

                ;+
                ; OPEN - OPEN A FILE FOR INPUT OR OUTPUT
                ;-

2DAB 22F380     OPEN:   SHLD    FPBP    ;SAVE FPB PNTR
2DAE 7E                 MOV     A,M     ;GET OPEN TYPE CODE
2DAF 32F580             STA     OCODE   ;SAVE OPEN TYPE CODE
2DB2 110B00             LXI     D,FVER-FPB
2DB5 19                 DAD     D       ;POINT TO VERSION
2DB6 7E                 MOV     A,M     ;GET VERSION
2DB7 32F680             STA     OVERS   ;SAVE ORIGINAL VERSION
2DBA 110B00             LXI     D,FHAN-FSIZ-1
2DBD 23                 INX     H       ;POINT TO FSIZ ...
2DBE 23                 INX     H       ;...
2DBF 23                 INX     H       ;...
2DC0 72                 MOV     M,D     ;SET FSIZ ...
2DC1 23                 INX     H       ;... TO ...
2DC2 72                 MOV     M,D     ;... ZERO
2DC3 19                 DAD     D       ;POINT TO FHAN
2DC4 EB                 XCHG            ;PUT PNTR IN D&E
2DC5 CD602D             CALL    OPDIR   ;"OPEN" DIRECTORY
2DC8 D8                 RC              ;READ ERROR!
2DC9 CD862D     L005:   CALL    GNDE    ;GET NEXT ENTRY
2DCC D8                 RC              ;ERROR!
2DCD CA5C2E             JZ      L011    ;END OF DIRECTORY!
2DD0 E5                 PUSH    H       ;SAVE H&L
2DD1 D5                 PUSH    D       ;SAVE D&E
2DD2 EB                 XCHG            ;D&E = DIR PNTR
2DD3 2AF380             LHLD    FPBP    ;H&L = FPB PNTR
2DD6 0E0B               MVI     C,11    ;SET COUNTER
2DD8 1A                 LDAX    D       ;GET ENTRY TYPE CODE
2DD9 FE01               CPI     TFREE   ;IS IT "FREE" ?
2DDB C2152E             JNZ     L106    ;NO!
2DDE 7E                 MOV     A,M     ;GET OPEN TYPE CODE
2DDF E601               ANI     FNEW    ;"NEW" FILE ?
2DE1 CA4A2E             JZ      L010    ;NO!
2DE4 23                 INX     H       ;POINT TO ATR BYTE
2DE5 3603               MVI     M,TFILE+TPROT ;SET ATR BYTE
2DE7 C5                 PUSH    B       ;SAVE B&C
2DE8 010B00             LXI     B,FSBK-FATR
2DEB 09                 DAD     B       ;POINT TO FSBK IN FPB
2DEC EB                 XCHG            ;...
```

```
2DED 09                 DAD     B       ;POINT TO FSBK IN DIRECTORY
2DEE EB                 XCHG            ;...
2DEF 0605               MVI     B,FLAD-FSBK ;SET COUNT
2DF1 CD4434             CALL    MOVHD   ;COPY PARAMETERS
2DF4 010500             LXI     B,FDBK-FLAD
2DF7 09                 DAD     B       ;POINT TO FDBK
2DF8 C1                 POP     B       ;RESTORE B&C
2DF9 3A1D81             LDA     DBLK    ;GET DIR BLOCK NUMBER
2DFC 77                 MOV     M,A     ;SAVE IT
2DFD 23                 INX     H
2DFE 70                 MOV     M,B     ;SAVE ENTRY NUMBER
2DFF D1                 POP     D       ;RESTORE D&E
2E00 E1                 POP     H       ;RESTORE H&L
2E01 05                 DCR     B       ;IS THIS LAST SLOT ?
2E02 C25C2E             JNZ     L011    ;NO: O.K.!
2E05 4F                 MOV     C,A     ;COPY DIR BLK NO.
2E06 3A1E81             LDA     MDBLK   ;GET MAX. DIR BLK NO.
2E09 B9                 CMP     C       ;IS THIS LAST DIR BLOCK ?
2E0A C25C2E             JNZ     L011    ;NO: O.K.!
2E0D CD5C2E             CALL    L011    ;CHECK FOR OTHER ERRORS
2E10 D8                 RC              ;ERROR: RETURN!
2E11 062D               MVI     B,EDPF  ;SET ERROR CODE
2E13 37                 STC             ;INDICATE ERROR
2E14 C9                 RET             ;DIR FULL: ERROR!
2E15 23         L106:   INX     H
2E16 0D         L006:   DCR     C       ;HAVE ALL MATCHED ?
2E17 CA302E             JZ      L007    ;YES!
2E1A 13                 INX     D       ;BUMP DIR PNTR
2E1B 23                 INX     H       ;BUMP FPB PNTR
2E1C 1A                 LDAX    D       ;GET NEXT DIR BYTE
2E1D BE                 CMP     M       ;COMPARE WITH NEXT FPB BYTE
2E1E CA162E             JZ      L006    ;MATCH!
2E21 DA4A2E             JC      L010    ;IF VERS, IT'S LESS!
2E24 79                 MOV     A,C     ;GET COUNTER
2E25 3D                 DCR     A       ;WAS THIS "VERSION" BYTE ?
2E26 C24A2E             JNZ     L010    ;NO!
2E29 3AF680             LDA     OVERS   ;GET ORIGINAL VERSION
2E2C A7                 ANA     A       ;WAS IT ZERO ?
2E2D C24A2E             JNZ     L010    ;NO: EXPLICIT VERSION!
2E30 3AF580     L007:   LDA     OCODE   ;GET OPEN TYPE CODE
2E33 E601               ANI     FNEW    ;"NEW" FILE ?
2E35 CA3D2E             JZ      L008    ;NO!
2E38 1A                 LDAX    D       ;GET VERS FROM DIRECTORY
2E39 77                 MOV     M,A     ;STORE IN FPB
2E3A C34A2E             JMP     L010    ;
2E3D C5         L008:   PUSH    B       ;SAVE B&C
2E3E 060B               MVI     B,FDBK-FVER ;SET COUNT
2E40 CD4434     L009:   CALL    MOVHD   ;MOVE PARAMETERS
2E43 C1                 POP     B       ;RESTORE B&C
2E44 3A1D81             LDA     DBLK    ;GET DIR BLOCK NUMBER
2E47 77                 MOV     M,A     ;STORE IN FPB
2E48 23                 INX     H       ;...
2E49 70                 MOV     M,B     ;STORE ENTRY NUMBER
2E4A D1         L010:   POP     D       ;RESTORE D&E
2E4B E1                 POP     H       ;RESTORE H&L
2E4C 0D                 DCR     C       ;FIND EXPLICIT VERSION ?
2E4D F2C92D             JP      L005    ;NO: CONTINUE SCAN
2E50 3AF580             LDA     OCODE   ;GET OPEN TYPE CODE
```

```
2E53 E601              ANI     FNEW     ;"NEW" FILE ?
2E55 CA822E            JZ      L013     ;NO: "OLD" FILE FOUND!
2E58 0624              MVI     B,EDFN   ;SET ERROR CODE
2E5A 37                STC
2E5B C9                RET              ;DUPLICATE FILENAME ERROR!
2E5C 2AF380    L011:   LHLD    FPBP     ;GET FPB PNTR
2E5F 7E                MOV     A,M      ;GET OPEN TYPE CODE
2E60 110B00            LXI     D,FVER-FPB
2E63 19                DAD     D        ;POINT TO FVER
2E64 E601              ANI     FNEW     ;"NEW" FILE ?
2E66 3AF680            LDA     OVERS    ;GET ORIGINAL VERSION
2E69 CA752E            JZ      L012     ;NO!
2E6C A7                ANA     A        ;TEST ORIGINAL VERSION
2E6D C2822E            JNZ     L013     ;EXPLICIT VERSION REQUESTED!
2E70 34                INR     M        ;INCREMENT VERSION
2E71 0627              MVI     B,EVOV   ;SET ERROR CODE
2E73 37                STC
2E74 C8                RZ               ;VERSION OVERFLOW ERROR!
2E75 A7        L012:   ANA     A        ;TEST ORIGINAL VERSION
2E76 C27E2E            JNZ     L023     ;EXPLICIT VERSION REQUESTED!
2E79 7E                MOV     A,M      ;GET "FOUND" VERSION
2E7A A7                ANA     A        ;IS IT STILL ZERO ?
2E7B C2822E            JNZ     L013     ;NO: O.K.!
2E7E 062A      L023:   MVI     B,EFNF   ;SET ERROR CODE
2E80 37                STC
2E81 C9                RET              ;FILE NOT FOUND!
2E82 2AF380    L013:   LHLD    FPBP     ;GET FPB PNTR
2E85 E5                PUSH    H        ;SAVE H&L
2E86 111A00            LXI     D,FHAN-FPB
2E89 19                DAD     D        ;POINT TO FHAN
2E8A 3EFD              MVI     A,-3     ;SET "SUB USER" FUNCTION
2E8C CDFC2E            CALL    CHDLR    ;CALL HANDLER
2E8F E1                POP     H        ;GET FPB PNTR BACK
2E90 E5                PUSH    H        ;RESAVE IT
2E91 010C00            LXI     B,FSBK-FPB
2E94 09                DAD     B        ;POINT TO FSBK
2E95 5E                MOV     E,M      ;GET STARTING ...
2E96 23                INX     H        ;... BLOCK ...
2E97 56                MOV     D,M      ;... NUMBER
2E98 011100            LXI     B,FBLK-FSBK-1
2E9B 09                DAD     B        ;POINT TO FBLK
2E9C 73                MOV     M,E      ;STORE STARTING ...
2E9D 23                INX     H        ;... BLOCK ...
2E9E 72                MOV     M,D      ;... NUMBER
2E9F E1                POP     H        ;RESTORE FPB POINTER
2EA0 AF                XRA     A        ;INDICATE NO ERRORS
2EA1 47                MOV     B,A      ;SET GOOD STATUS CODE
2EA2 C9                RET              ;RETURN TO CALLER

;+
; READ - READ FILE. CURRENTLY ONLY "IMAGE"
;        TYPE FILES ARE SUPPORTED.
;        "OPEN" MUST BE CALLED BEFORE CALLING "READ".
;
; INPUTS  - B&C: MAX. ALLOWABLE BYTE COUNT
;           D&E: IF ZERO, THEN FILE WILL BE READ IN TO
;                MEMORY AT LOAD ADDRESS SPECIFIED IN
```

```
                ;                   DIRECTORY.  IF NON-ZERO, THEN D&E IS
                ;                   USED AS THE LOAD ADDRESS.
                ;             H&L:  PNTR TO FPB.
                ;
                ;-

2EA3 CD032F     READ:   CALL    LADSB   ;SETUP MEM. PNTR & GET SIZE
2EA6 C5                 PUSH    B       ;SAVE MAX. BYTE COUNT
2EA7 4E                 MOV     C,M     ;GET LAST BLK BC
2EA8 EB                 XCHG            ;H&L=NUMBER OF BLOCKS
2EA9 CDBA35             CALL    BK2BC   ;COMPUTE BYTE COUNT
2EAC EB                 XCHG            ;D&E=TOTAL BYTE COUNT
2EAD E3                 XTHL            ;GET MAX. BYTE COUNT
2EAE CD4D34             CALL    CMPHD   ;COMPARE WITH FILE SIZE
2EB1 E1                 POP     H       ;RESTORE H&L
2EB2 0639               MVI     B,ERSZ  ;SET ERROR CODE
2EB4 D8                 RC              ;FILE TOO LARGE: ERROR!
2EB5 011200             LXI     B,FXBC-FLBC
2EB8 09                 DAD     B       ;POINT TO FXBC
2EB9 73                 MOV     M,E     ;STORE ...
2EBA 23                 INX     H       ;... BYTE ...
2EBB 72                 MOV     M,D     ;... COUNT
2EBC 01F7FF             LXI     B,FHAN-FXBC-1
2EBF 09                 DAD     B       ;POINT TO FHAN
2EC0 CDFB2E             CALL    RD      ;READ THE FILE
2EC3 0630               MVI     B,EFRD  ;SET ERROR CODE
2EC5 2AF380     RWE:    LHLD    FPBP    ;RESTORE FPB PNTR
2EC8 D8                 RC              ;ERROR EXIT!
2EC9 0600               MVI     B,0     ;SET GOOD STATUS CODE
2ECB C9                 RET             ;EXIT!

                ;+
                ; WRITE - FILE WRITE.  CURRENTLY ONLY "IMAGE" TYPE
                ;         FILES ARE SUPPORTED.  "OPEN" MUST BE
                ;         CALLED BEFORE CALLING "WRITE".
                ;
                ;-

2ECC E5         WRITE:  PUSH    H       ;SAVE FPB PNTR
2ECD D5                 PUSH    D       ;SAVE AUX. LOAD ADDRESS
2ECE 112200             LXI     D,FXBC-FPB
2ED1 19                 DAD     D       ;POINT TO FXBC
2ED2 D1                 POP     D       ;RESTORE AUX. LOAD ADDRESS
2ED3 71                 MOV     M,C     ;STORE ...
2ED4 23                 INX     H       ;... BYTE ...
2ED5 70                 MOV     M,B     ;... COUNT
2ED6 E1                 POP     H       ;RESTORE FPB POINTER
2ED7 CD032F             CALL    LADSB   ;SETUP MEM. PNTR & GET SIZE
2EDA C5                 PUSH    B       ;PUT BYTE COUNT ON STACK
2EDB E3                 XTHL            ;NOW HL=BYTE COUNT
2EDC CDA835             CALL    BC2BK   ;CONVERT TO BLOCKS & BYTE COUNT
2EDF CD5334             CALL    CMPDH   ;COMPARE H&L WITH D&E
2EE2 EB                 XCHG            ;D&E=NUMBER OF BLOCKS
2EE3 E1                 POP     H       ;H&L=PNTR TO FLBC
2EE4 0636               MVI     B,EWSZ  ;SET ERROR CODE
2EE6 D8                 RC              ;ERROR EXIT!
2EE7 71                 MOV     M,C     ;STORE LAST BLOCK BYTE COUNT
```

```
2EE8 010800             LXI     B,FAUX-FLBC
2EEB 09                 DAD     B               ;POINT TO FAUX
2EEC 73                 MOV     M,E             ;STORE NUMBER ...
2EED 23                 INX     H               ;... OF BLOCKS ...
2EEE 72                 MOV     M,D             ;... USED
2EEF 23                 INX     H               ;POINT TO FHAN
2EF0 CDF82E             CALL    WR              ;WRITE THE FILE
2EF3 0633               MVI     B,EFWR          ;SET ERROR CODE
2EF5 C3C52E             JMP     RWE             ;TAKE COMMON EXIT!

2EF8 3E01       WR:     MVI     A,1             ;SET "WRITE" FUNCTION CODE
2EFA FE                 DB      OFEH            ;** SKIP ONE BYTE **
2EFB AF         RD:     XRA     A               ;SET "READ" FUNCTION CODE
2EFC 5E         CHDLR:  MOV     E,M             ;GET ...
2EFD 23                 INX     H               ;... HANDLER ...
2EFE 56                 MOV     D,M             ;... ADDRESS
2EFF 23                 INX     H               ;POINT TO PARAMETER BLOCK
2F00 77                 MOV     M,A             ;STORE FUNCTION CODE
2F01 D5         JMPD:   PUSH    D               ;JUMP INDIRECT ...
2F02 C9                 RET                     ;... D&E


2F03 22F380     LADSB:  SHLD    FPBP            ;SAVE FPB PNTR
2F06 7A                 MOV     A,D             ;TEST AUX. ...
2F07 B3                 ORA     E               ;... LOAD ADDRESS
2F08 C2142F             JNZ     LO14            ;NON-ZERO: USE IT!
2F0B E5                 PUSH    H               ;SAVE H&L
2F0C 111100             LXI     D,FLAD-FPB
2F0F 19                 DAD     D               ;POINT TO FLAD
2F10 5E                 MOV     E,M             ;GET LOBYTE
2F11 23                 INX     H               ;...
2F12 56                 MOV     D,M             ;GET HIBYTE
2F13 E1                 POP     H               ;RESTORE H&L
2F14 D5         LO14:   PUSH    D               ;SAVE D&E
2F15 112000             LXI     D,FBUF-FPB
2F18 19                 DAD     D               ;POINT TO FBUF
2F19 D1                 POP     D               ;RESTORE D&E
2F1A 73                 MOV     M,E             ;STORE LOAD ...
2F1B 23                 INX     H               ;... ADDRESS IN ...
2F1C 72                 MOV     M,D             ;... FBUF
2F1D 11EDFF             LXI     D,FSIZ-FBUF-1
2F20 19                 DAD     D               ;POINT TO FSIZ
2F21 5E                 MOV     E,M             ;GET NUMBER ...
2F22 23                 INX     H               ;... OF ...
2F23 56                 MOV     D,M             ;... BLOCKS
2F24 23                 INX     H               ;POINT TO FLBC
2F25 C9                 RET                     ;RETURN


                ;+
                ; CLOSE - FILE CLOSE ROUTINE.
                ;
                ;-

2F26 0600       CLOSE:  MVI     B,0             ;SET GOOD STATUS CODE
2F28 7E                 MOV     A,M             ;GET OPEN TYPE CODE
2F29 E601               ANI     FNEW            ;"NEW" FILE ?
2F2B C8                 RZ                      ;NO: RETURN!
```

```
2F2C 22F380            SHLD    FPBP      ;SAVE FPB PNTR
2F2F CDB52F            CALL    GODBK     ;GET DIRECTORY BLOCK
2F32 D8                RC               ;ERROR!
2F33 111800            LXI     D,FAUX-FPB
2F36 19                DAD     D         ;POINT TO FAUX
2F37 5E                MOV     E,M       ;GET NUMBER ...
2F38 23                INX     H         ;... OF BLOCKS ...
2F39 56                MOV     D,M       ;... IN FILE
2F3A D5                PUSH    D         ;SAVE FILE BLOCKS
2F3B 11F3FF            LXI     D,FSBK-FAUX-1
2F3E 19                DAD     D         ;POINT TO FSBK
2F3F 5E                MOV     E,M       ;GET FILE ...
2F40 23                INX     H         ;... STARTING ...
2F41 56                MOV     D,M       ;... BLOCK NUMBER
2F42 23                INX     H         ;POINT TO FSIZ
2F43 E3                XTHL             ;SAVE PNTR & GET FILE BLOCKS
2F44 EB                XCHG             ;D&E=FILE BLOCKS
2F45 19                DAD     D         ;H&L=START BLK OF FREE SPACE
2F46 E3                XTHL             ;SAVE H&L & GET PNTR
2F47 D5                PUSH    D         ;SAVE FILE BLOCKS
2F48 5E                MOV     E,M       ;GET TOTAL ...
2F49 23                INX     H         ;... BLOCKS IN ...
2F4A 56                MOV     D,M       ;... "FREE" SLOT
2F4B E3                XTHL             ;SAVE PNTR & GET FILE BLOCKS
2F4C EB                XCHG             ;H&L=OLD "FREE" BLOCKS
2F4D CD5934            CALL    SUBHD     ;COMPUTE NEW "FREE" BLOCKS
2F50 E3                XTHL             ;SAVE NEW "FREE" & GET PNTR
2F51 72                MOV     M,D       ;STORE NUMBER ...
2F52 2B                DCX     H         ;... OF BLOCKS ...
2F53 73                MOV     M,E       ;... IN FILE
2F54 11F3FF            LXI     D,FATR-FSIZ
2F57 19                DAD     D         ;POINT TO FATR
2F58 C5                PUSH    B         ;SAVE B&C
2F59 E5                PUSH    H         ;SAVE PNTR
2F5A 219D81            LXI     H,DBF+2+6*21 ;POINT PAST LAST DIR ENTRY
2F5D 11EBFF            LXI     D,-21     ;ENTRY SIZE
2F60 19          LO17: DAD     D         ;STEP BACK ONE ENTRY
2F61 05                DCR     B         ;AT CORRECT ENTRY ?
2F62 C2602F            JNZ     LO17      ;NO: STEP AGAIN!
2F65 D1                POP     D         ;GET PNTR TO FATR
2F66 0615              MVI     B,21      ;SET COUNT
2F68 CD4434            CALL    MOVHD     ;MOVE DIRECTORY ENTRY
2F6B C1                POP     B         ;GET OLD B&C
2F6C 05                DCR     B         ;WAS THIS LAST DIR. SLOT ?
2F6D CCAA2F            CZ      LO25      ;YES: STORE BLK NO., ETC.
2F70 C1                POP     B         ;GET SIZE
2F71 D1                POP     D         ;GET START BLK. NO.
2F72 CD962F            CALL    SFR       ;STORE SIZE & START BLK. NO.
2F75 111D81    WRDIR:  LXI     D,DBF     ;POINT TO DIRECTORY BUFFER
2F78 CD5934            CALL    SUBHD     ;COMPUTE DIRECTORY BYTE COUNT
2F7B EB                XCHG             ;D&E = BYTE COUNT
2F7C 2AF380            LHLD    FPBP      ;GET FPB PNTR
2F7F 012200            LXI     B,FXBC-FPB
2F82 09                DAD     B         ;POINT TO FXBC
2F83 73                MOV     M,E       ;STORE ...
2F84 23                INX     H         ;... BYTE ...
2F85 72                MOV     M,D       ;... COUNT
2F86 11F7FF            LXI     D,FHAN-FXBC-1
```

```
2F89 19              DAD     D          ;POINT TO FHAN
2F8A CDF82E          CALL    WR         ;WRITE DIRECTORY
2F8D 0600            MVI     B,0        ;SET GOOD STATUS CODE
2F8F 2AF380          LHLD    FPBP       ;GET FPB PNTR
2F92 D0              RNC                ;O.K.: EXIT!
2F93 060C            MVI     B,EDIR     ;SET ERROR CODE
2F95 C9              RET                ;ERROR EXIT!

2F96 3601    SFR:    MVI     M,TFREE    ;SET 'FREE' ATTRIBUTE BYTE
2F98 D5              PUSH    D
2F99 110B00          LXI     D,FSBK-FATR
2F9C 19              DAD     D          ;POINT TO FSBK
2F9D D1              POP     D
2F9E 73              MOV     M,E        ;STORE START ...
2F9F 23              INX     H          ;... BLOCK NUMBER ...
2FA0 72              MOV     M,D        ;... OF FREE SPACE
2FA1 23              INX     H
2FA2 71              MOV     M,C        ;STORE ...
2FA3 23              INX     H          ;... SIZE OF ...
2FA4 70              MOV     M,B        ;... FREE SPACE ...
2FA5 23              INX     H
2FA6 3680            MVI     M,128      ;STORE LBC
2FA8 23              INX     H
2FA9 C9              RET                ;RETURN

2FAA 3A1E81  L025:   LDA     MDBLK      ;GET MAX. DIR. BLK. NO.
2FAD B9              CMP     C          ;IS THIS THE LAST BLOCK ?
2FAE C8              RZ                 ;YES: RETURN!
2FAF 0C              INR     C          ;BUMP BLOCK NUMBER
2FB0 71              MOV     M,C        ;STORE IT
2FB1 23              INX     H
2FB2 77              MOV     M,A        ;STORE MAX. BLK. NO.
2FB3 23              INX     H
2FB4 C9              RET                ;RETURN TO CALLER

2FB5 2AF380  GODBK:  LHLD    FPBP       ;GET FPB PNTR
2FB8 E5              PUSH    H          ;SAVE IT
2FB9 111600          LXI     D,FDBK-FPB
2FBC 19              DAD     D          ;POINT TO FDBK
2FBD 4E              MOV     C,M        ;GET DIRECTORY BLOCK NUMBER
2FBE 23              INX     H          ;POINT TO FDEN
2FBF 46              MOV     B,M        ;GET DIRECTORY ENTRY NUMBER
2FC0 110700          LXI     D,FBLK-FDEN
2FC3 19              DAD     D          ;POINT TO FBLK
2FC4 71              MOV     M,C        ;STORE BLOCK NUMBER
2FC5 23              INX     H
2FC6 11812D          LXI     D,DIP+1    ;POINT TO REST OF PARAMETERS
2FC9 C5              PUSH    B          ;SAVE B&C
2FCA 0605            MVI     B,DIPS-1   ;SET COUNT
2FCC CD4434          CALL    MOVHD      ;STORE REST OF PARAMETERS
2FCF C1              POP     B          ;RESTORE B&C
2FD0 C5              PUSH    B          ;RESAVE B&C
2FD1 11F6FF          LXI     D,FHAN-FXBC-2
2FD4 19              DAD     D          ;POINT TO FHAN
2FD5 CDFB2E          CALL    RD         ;READ DIR BLOCK
2FD8 C1              POP     B          ;RESTORE B&C
2FD9 E1              POP     H          ;RESTORE FPB PNTR
```

```
2FDA D0                 RNC             ;NO ERRORS!
2FDB 060C               MVI     B,EDIR  ;SET ERROR CODE
2FDD C9                 RET             ;RETURN TO CALLER

                        ;+
                        ; PDV - PARSE DEVICE NAME
                        ; <C> : INVALID DEVICE
                        ; <Z> : NO DEVICE, DEFAULT USED
                        ; OTHERWISE: GOT VALID DEVICE
                        ;-

2FDE CD6034     PDV:    CALL    SPNOR   ;SKIP SPACES
2FE1 D5                 PUSH    D       ;SAVE FPB PNTR
2FE2 E5                 PUSH    H       ;SAVE STRING PNTR
2FE3 EB                 XCHG
2FE4 2AF080             LHLD    DFDV    ;GET DEFAULT DEVICE
2FE7 EB                 XCHG            ;PUT IN D&E
2FE8 3AF280             LDA     DFUN    ;GET DEFAULT UNIT NUMBER
2FEB 4F                 MOV     C,A     ;PUT IN C
2FEC 7E                 MOV     A,M     ;TEST IF ...
2FED A7                 ANA     A       ;... END OF LINE
2FEE CA1930             JZ      PDV05   ;END OF LINE!
2FF1 0604               MVI     B,4     ;SET COUNTER
2FF3 05         PDV01:  DCR     B       ;ENOUGH ?
2FF4 CA1930             JZ      PDV05   ;YES: NO DEVICE!
2FF7 23                 INX     H
UA6 2FF8 7E             MOV     A,M     ;GET NEXT CHARACTER
    2FF9 A7             ANA     A       ;TEST FOR END OF LINE
    2FFA CA1930         JZ      PDV05   ;END OF LINE: NO DEVICE!
    2FFD FE3A           CPI     ':'     ;CHECK FOR COLON
    2FFF C2F32F         JNZ     PDV01   ;NO: LOOP!
3002 E1         PDV02:  POP     H       ;RESTORE STRING PNTR
3003 3E03               MVI     A,3
3005 B8                 CMP     B
3006 CA0F30             JZ      PDV03   ;UNIT NUMBER ONLY!
UA7 3009 5E             MOV     E,M     ;GET ...
    300A 23             INX     H       ;... DEVICE ...
    300B 56             MOV     D,M     ;... NAME
    300C 23             INX     H
    300D 0E30           MVI     C,'0'   ;SET UNIT ZERO
300F 05         PDV03:  DCR     B
3010 05                 DCR     B
3011 CA1630             JZ      PDV04   ;NO UNIT NUMBER!
3014 4E                 MOV     C,M     ;GET UNIT NUMBER
3015 23                 INX     H
3016 23         PDV04:  INX     H       ;SKIP OVER COLON
3017 E5                 PUSH    H       ;SAVE STRING POINTER
3018 A7                 ANA     A       ;SET FLAGS TO <NZ>
3019 E1         PDV05:  POP     H       ;GET STRING PNTR
301A E3                 XTHL            ;SWAP
301B F5                 PUSH    PSW     ;SAVE FLAGS
301C 79                 MOV     A,C     ;COPY UNIT NUMBER
301D D630               SUI     '0'     ;CONVERT FROM ASCII
301F F5                 PUSH    PSW     ;SAVE UNIT NUMBER
3020 E5                 PUSH    H       ;SAVE ORIGINAL D&E
3021 218E36             LXI     H,HDVCT ;POINT TO DEVICE TABLE
3024 0E03               MVI     C,3     ;SET NUMBER OF SLOTS
```

```
3026 E5        PDV06:  PUSH    H           ;SAVE HANDLER PNTR
3027 3EC3              MVI     A,0C3H      ;SET 'JMP' BYTE
3029 AE                XRA     M           ;VECTOR EMPTY ?
302A C25530            JNZ     PDV07       ;YES: SKIP IT!
302D 23                INX     H           ;POINT TO ASCII DEVICE NAME ...
302E 23                INX     H
302F 23                INX     H
3030 7E                MOV     A,M         ;GET 1ST CHAR. OF NAME
3031 BB                CMP     E           ;MATCH ?
3032 C25530            JNZ     PDV07       ;NO: GO TRY NEXT DEVICE!
3035 23                INX     H
3036 7E                MOV     A,M         ;GET 2ND CHAR. OF NAME
3037 BA                CMP     D           ;MATCH ?
3038 C25530            JNZ     PDV07       ;NO: GO TRY NEXT DEVICE!
303B 23                INX     H           ;POINT TO NUMBER OF UNITS
303C 66                MOV     H,M         ;GET NUMBER OF UNITS
303D C1                POP     B           ;GET HANDLER PNTR
303E D1                POP     D           ;RESTORE D&E
303F F1                POP     PSW         ;GET UNIT NUMBER
3040 BC                CMP     H           ;VALID UNIT NUMBER ?
3041 D26230            JNC     PDV08       ;NO: ERROR!
3044 211A00            LXI     H,FHAN-FPB
3047 19                DAD     D           ;POINT TO FHAN IN FPB
3048 71                MOV     M,C         ;STORE ...
3049 23                INX     H           ;... HANDLER ...
304A 70                MOV     M,B         ;... ADDRESS
304B 23                INX     H
304C 23                INX     H
304D 77                MOV     M,A         ;STORE UNIT NUMBER
304E F1                POP     PSW         ;RESTORE FLAGS
304F E1                POP     H           ;RESTORE STRING PNTR
3050 37                STC                 ;CLEAR ...
3051 3F                CMC                 ;... C-FLAG
3052 0618              MVI     B,EMDV      ;SET ERROR CODE
3054 C9                RET                 ;EXIT!
3055 E1        PDV07:  POP     H           ;GET HANDLER PNTR
3056 79                MOV     A,C         ;SAVE COUNTER
3057 010A00            LXI     B,10
305A 09                DAD     B           ;STEP TO NEXT HANDLER
305B 4F                MOV     C,A         ;COPY COUNTER BACK
305C 0D                DCR     C           ;ANY MORE DEVICES ?
305D C22630            JNZ     PDV06       ;YES: TRY NEXT DEVICE!
3060 D1                POP     D           ;RESTORE D&E
3061 F1                POP     PSW         ;DISCARD UNIT NUMBER
3062 F1        PDV08:  POP     PSW         ;RESTORE FLAGS
3063 E1                POP     H           ;RESTORE STRING PNTR
3064 061E              MVI     B,EIVD      ;SET ERROR CODE
3066 37                STC                 ;INDICATE ENVALID DEVICE
3067 C9                RET                 ;EXIT!

3068 015D2D    PSFSP:  LXI     B,STYP      ;SET DEFAULT TYPE 'SRC'
306B C37730            JMP     PFSPC
306E 01532D    PNFSP:  LXI     B,NTYP      ;SET BLANK DEFAULT TYPE
3071 C37730            JMP     PFSPC
3074 01572D    PPFSP:  LXI     B,PTYP      ;SET DEFAULT TYPE 'PRQ'
3077 E5        PFSPC:  PUSH    H           ;SAVE STRING PNTR
3078 D5                PUSH    D           ;SAVE FPB PNTR
```

```
3079 C5              PUSH    B           ;SAVE DEFAULT TYPE PNTR
307A 210800          LXI     H,FTYP-FPB
307D 19              DAD     D           ;POINT TO TYPE
307E EB              XCHG                ;PNTR IN D&E
307F E1              POP     H           ;GET DEFAULT TYPE PNTR
3080 0603            MVI     B,3         ;SET COUNT
3082 CD9534          CALL    MSTR        ;MOVE DEFAULT TYPE IN
3085 D1              POP     D           ;RESTORE FPB PNTR
3086 E1              POP     H           ;RESTORE STRING PNTR
3087 CDDE2F  PCFSP:  CALL    PDV         ;PARSE DEVICE NAME
308A D8              RC                  ;ERROR!
308B D5              PUSH    D           ;SAVE FPB PNTR
308C 13              INX     D
308D 13              INX     D           ;POINT TO NAME
308E CD7234          CALL    LODG        ;LETTER OR DIGIT ?
3091 D2C130          JNC     PFS04       ;NO: NO NAME!
3094 0606            MVI     B,6         ;SET COUNT
3096 CD9534          CALL    MSTR        ;MOVE NAME IN
3099 7E              MOV     A,M         ;GET NEXT CHAR.
309A FE2E            CPI     '.'         ;IS THERE A TYPE ?
309C CAA530          JZ      PFS01       ;YES!
309F 13              INX     D           ;SKIP ...
30A0 13              INX     D           ;... OVER ...
30A1 13              INX     D           ;... TYPE
30A2 C3AB30          JMP     PFS02       ;
30A5 23      PFS01:  INX     H           ;SKIP OVER '.'
30A6 0603            MVI     B,3         ;SET COUNT
30A8 CD9534          CALL    MSTR        ;MOVE TYPE IN
30AB AF      PFS02:  XRA     A           ;CLEAR A
30AC 12              STAX    D           ;STORE ZERO VERSION NUMBER
30AD 7E              MOV     A,M         ;GET NEXT CHAR.
30AE FE3B            CPI     ';'         ;IS THERE A VERSION ?
30B0 C2BB30          JNZ     PFS03       ;NO!
30B3 23              INX     H           ;SKIP OVER ';'
30B4 D5              PUSH    D           ;SAVE PNTR TO VERSION
30B5 CDF634          CALL    GN2Z        ;GET NUMBER
30B8 7B              MOV     A,E         ;USE LOBYTE OF NUMBER
30B9 D1              POP     D           ;RESTORE PNTR TO VERSION
30BA 12              STAX    D           ;STORE VERSION NUMBER
30BB 1A      PFS03:  LDAX    D           ;GET VERSION NUMBER
30BC D1              POP     D           ;RESTORE FPB PNTR
30BD 061B            MVI     B,EMVR      ;SET STATUS CODE
30BF A7              ANA     A           ;TEST VERSION NUMBER
30C0 C9              RET                 ;RETURN TO CALLER
30C1 D1      PFS04:  POP     D           ;RESTORE FPB PNTR
30C2 0615            MVI     B,EMFN      ;SET ERROR CODE
30C4 37              STC                 ;INDICATE ERROR
30C5 C9              RET                 ;RETURN TO CALLER


;+
;     THIS IS AN INITIAL SET OF ROUTINES TO FACILITATE FILE
; OPERATIONS WITH THE FCS SYSTEM.  ROUTINES ARE PROVIDED
; FOR SEQUENTIAL BYTE ACCESS, SEQUENTIAL RECORD ACCESS,
; AND BLOCK ACCESS.
;
;     CERTAIN PARAMETERS IN THE FILE PARAMETER BLOCK (FPB)
```

```
;  MUST BE SET UP BEFORE USING ANY OF THESE ROUTINES.  THESE
;  PARAMETERS SHOULD BE SET UP AFTER CALLING 'OPEN' TO OPEN
;  THE FILE, BUT BEFORE ANY CALL TO ANY OF THESE ACCESS
;  ROUTINES.
;
;     THE PARAMETERS ARE :
;
;  FBUF - SHOULD BE SET TO THE ADDRESS OF THE USER-PROVIDED
;           BLOCK BUFFER.
;
;  FXBC - FOR SEQUENTIAL ACCESS, SHOULD BE SET TO THE SIZE
;           (NUMBER OF BYTES) OF THE USER-PROVIDED BLOCK BUFFER.
;           THE SIZE SHOULD BE A MULTIPLE OF THE SYSTEM STANDARD
;           BLOCK SIZE, 128 DECIMAL.
;         FOR DIRECT BLOCK ACCESS, SHOULD BE SET TO THE NUMBER
;           OF BYTES TO BE TRANSFERRED.
;
;     WHEN USING SEQUENTIAL BYTE OR RECORD ACCESS, THESE
;  PARAMETERS SHOULD NOT BE CHANGED DURING THE I/O OPERATIONS.
;     WHEN USING DIRECT BLOCK ACCESS, THESE PARAMETERS MAY
;  BE SET TO THE DESIRED VALUES FOR THE TRANSFER, PRIOR TO
;  EACH CALL TO RBLK, RBLKI, WBLK, OR WBLKI.  THEY'RE VALUES
;  ARE PRESERVED BY THE BLOCK I/O ROUTINES, SO THEY
;  DO NOT NEED TO BE RESET PRIOR TO EACH CALL UNLESS
;  YOU SPECIFICALLY WANT TO CHANGE EITHER THE LOCATION
;  OR THE SIZE OF THE BUFFER.
;
;    SEE THE INDIVIDUAL DESCRIPTION FOR DETAILS ON EACH
;  OF THE FOLLOWING ACCESS ROUTINES.
;-


;+
;  *** RWSEQI *** - "REWIND SEQUENTIAL INPUT" ROUTINE
;
;   RWSEQI IS USED TO "REWIND" A SEQUENTIAL INPUT FILE.
;
;   RWSEQI MUST BE CALLED BEFORE THE FIRST CALL TO ANY
;   OF THE SEQUENTIAL BYTE OR RECORD INPUT ROUTINES !
;
;  INPUTS:   HL => FPB
;
;  OUTPUTS:  A - LOST
;            BC, DE - UNCHANGED
;            HL => FPB
;
;  STATUS:   NONE
;-

30C6 E5        RWSEQI:  PUSH    H          ;SAVE FPB PTR
30C7 D5                 PUSH    D
30C8 AF                 XRA     A          ;CLEAR A REGISTER
30C9 111E00             LXI     D,FBLK-FPB
30CC 19                 DAD     D          ;POINT TO FBLK
30CD 77                 MOV     M,A        ;SET VIRTUAL BLOCK = 0 ...
30CE 23                 INX     H
30CF 77                 MOV     M,A
30D0 11F9FF             LXI     D,FAUX-FBLK-1
```

```
30D3 19                 DAD     D       ;POINT TO AUX. BYTE COUNT
30D4 77                 MOV     M,A     ;INITIALIZE AUX. BYTE COUNT ...
30D5 23                 INX     H
30D6 77                 MOV     M,A
30D7 D1                 POP     D
30D8 E1                 POP     H       ;RESTORE FPB PTR
                ;
30D9 E5         ZPTR:   PUSH    H       ;SAVE FPB PTR
30DA D5                 PUSH    D
30DB 112400             LXI     D,FPTR-FPB
30DE 19                 DAD     D       ;POINT TO RELATIVE BUFFER POINTER
30DF 3600               MVI     M,0     ;ZERO IT ...
30E1 23                 INX     H
30E2 3600               MVI     M,0
30E4 D1                 POP     D
30E5 E1                 POP     H       ;RESTORE FPB PTR
30E6 C9                 RET


                ;+
                ; *** INSEQO *** "INITIALIZE SEQUENTIAL OUTPUT" ROUTINE
                ;
                ;  INSEQO IS USED TO INITIALIZE A NEWLY CREATED OPEN FILE
                ; FOR SEQUENTIAL BYTE OR RECORD OUTPUT OPERATIONS.
                ;
                ;  INSEQO MUST BE CALLED BEFORE THE FIRST CALL TO ANY OF
                ;  THE SEQUENTIAL BYTE OR RECORD OUTPUT ROUTINES !
                ;
                ; INPUTS:   HL => FPB
                ;
                ; OUTPUTS:  A - LOST
                ;           BC,DE - UNCHANGED
                ;           HL => FPB
                ;
                ; STATUS:   NONE
                ;-

30E7 CDC630     INSEQO: CALL    RWSEQI  ;INITIALIZE THINGS
30EA D5                 PUSH    D
30EB C5                 PUSH    B
30EC 110000             LXI     D,0     ;VIRTUAL BLOCK ZERO
30EF CDF530             CALL    CBC     ;SETUP THINGS
30F2 C1                 POP     B
30F3 D1                 POP     D
30F4 C9                 RET             ;EXIT !
                ;
30F5 E5         CBC:    PUSH    H       ;SAVE FPB PTR
30F6 010E00             LXI     B,FSIZ-FPB
30F9 09                 DAD     B       ;POINT TO FSIZ
30FA 7E                 MOV     A,M     ;GET IT ...
30FB 23                 INX     H
30FC 66                 MOV     H,M
30FD 6F                 MOV     L,A
30FE CD5334             CALL    CMPDH   ;COMPARE WITH VIRTUAL BLOCK NUMBER
3101 DA0831             JC      CBC1    ;O.K.: VALID BLOCK NUMBER !
3104 E1                 POP     H       ;RESTORE FPB PTR
3105 AF                 XRA     A
3106 37                 STC
```

```
3107 C9                 RET                     ;<C><Z>: INVALID BLOCK NUMBER !
3108 CD5934    CBC1:    CALL    SUBHD           ;COMPUTE NUMBER OF REMAINING BLOCKS
310B EB                 XCHG
310C E1                 POP     H               ;GET FPB PTR
310D E5                 PUSH    H               ;SAVE FPB PTR
310E 012200             LXI     B,FXBC-FPB
3111 09                 DAD     B               ;POINT TO FXBC
3112 7E                 MOV     A,M             ;GET BUFFER SIZE ...
3113 23                 INX     H
3114 66                 MOV     H,M
3115 6F                 MOV     L,A
3116 CDA835             CALL    BC2BK           ;CONVERT TO NUMBER OF BLOCKS
3119 CD4D34             CALL    CMPHD           ;ENTIRE BUFFER O.K. ?
311C DA2731             JC      CBC2            ;YES !
311F E1                 POP     H               ;GET FPB PTR
3120 E5                 PUSH    H               ;SAVE FPB PTR
3121 011000             LXI     B,FLBC-FPB
3124 09                 DAD     B               ;POINT TO LAST BLOCK BYTE COUNT
3125 4E                 MOV     C,M             ;GET IT
3126 EB                 XCHG                    ;HL = REMAINING BLOCKS
3127 CDBA35    CBC2:    CALL    BK2BC           ;CONVERT TO BYTE COUNT
312A EB                 XCHG
312B E1                 POP     H               ;GET FPB PTR
312C E5                 PUSH    H               ;SAVE FPB PTR
312D 011800             LXI     B,FAUX-FPB
3130 09                 DAD     B               ;POINT TO AUX. BYTE COUNT
3131 73                 MOV     M,E             ;SET IT ...
3132 23                 INX     H
3133 72                 MOV     M,D
3134 E1                 POP     H               ;RESTORE FPB PTR
3135 C9                 RET                     ;<NC>: O.K. !

                ;+
                ; *** CLSEQO *** "CLOSE SEQUENTIAL OUTPUT" ROUTINE
                ;
                ;  CLSEQO IS USED TO CLOSE A NEWLY CREATED SEQUENTIAL
                ;  OUTPUT FILE. THE REMAINING UNWRITTEN PART OF THE BLOCK
                ;  BUFFER IS WRITTEN OUT, IF ANY, AND THE FINAL FILE SIZE
                ;  IS CALCULATED AND APPROPRIATE INFORMATION UPDATED IN
                ;  THE FPB. THEN 'CLOSF' IS CALLED TO ENTER THE FILE INTO
                ;  THE DIRECTORY.
                ;
                ; INPUTS:   HL => FPB
                ;
                ; OUTPUTS:  A,BC,DE - LOST
                ;           HL => FPB IF NO ERRORS, ELSE HL LOST
                ;
                ; STATUS:   <NC> - NO ERRORS, B=0
                ;           <C>  - FILE WRITE ERROR OR DIRECTORY WRITE ERROR,
                ;                  WITH B = SYSTEM ERROR CODE
                ;-

3136 E5         CLSEQO: PUSH    H               ;SAVE FPB PTR
3137 112400             LXI     D,FPTR-FPB
313A 19                 DAD     D               ;POINT TO RELATIVE BUFFER POINTER
313B 5E                 MOV     E,M             ;GET IT ...
313C 23                 INX     H
```

```
313D 56              MOV     D,M
313E 01EBFF          LXI     B,FLBC-FPTR-1
3141 09              DAD     B       ;POINT TO LAST BLOCK BYTE COUNT
3142 EB              XCHG
3143 CDA835          CALL    BC2BK   ;CONVERT TO BLOCKS & LAST BLOCK BYTE COUNT
3146 EB              XCHG
3147 71              MOV     M,C     ;SET LAST BLOCK BYTE COUNT
3148 010E00          LXI     B,FBLK-FLBC
314B 09              DAD     B       ;POINT TO NEXT VIRTUAL BLOCK
314C 7E              MOV     A,M     ;GET IT ...
314D 23              INX     H
314E 66              MOV     H,M
314F 6F              MOV     L,A
3150 19              DAD     D       ;COMPUTE ACTUAL FILE SIZE
3151 EB              XCHG
3152 E1              POP     H       ;GET FPB PTR
3153 E5              PUSH    H       ;SAVE FPB PTR
3154 010E00          LXI     B,FSIZ-FPB
3157 09              DAD     B       ;POINT TO FILE SIZE
3158 4E              MOV     C,M     ;GET IT ...
3159 23              INX     H
315A 46              MOV     B,M
315B 72              MOV     M,D     ;SET ACTUAL FILE SIZE ...
315C 2B              DCX     H
315D 73              MOV     M,E
315E E1              POP     H       ;RESTORE FPB PTR
315F C5              PUSH    B       ;SAVE ORIGINAL FILE SIZE
3160 CD0232          CALL    IWBKI   ;WRITE REMAINING PART OF BUFFER
3163 C1              POP     B       ;GET ORIGINAL FILE SIZE
3164 F5              PUSH    PSW     ;SAVE STATUS
3165 E5              PUSH    H       ;SAVE FPB PTR
3166 110E00          LXI     D,FSIZ-FPB
3169 19              DAD     D       ;POINT TO FILE SIZE
316A 5E              MOV     E,M     ;GET ACTUAL FILE SIZE ...
316B 23              INX     H
316C 56              MOV     D,M
316D 70              MOV     M,B     ;RESTORE ORIGINAL FILE SIZE ...
316E 2B              DCX     H
316F 71              MOV     M,C
3170 010A00          LXI     B,FAUX-FSIZ
3173 09              DAD     B       ;POINT TO ACTUAL FILE SIZE
3174 73              MOV     M,E     ;SET ACTUAL FILE SIZE FOR CLOSE ...
3175 23              INX     H
3176 72              MOV     M,D
3177 E1              POP     H       ;RESTORE FPB PTR
3178 F1              POP     PSW     ;RESTORE STATUS
3179 0633            MVI     B,EFWR
317B D8              RC              ;<C>: FILE WRITE ERROR !
317C C3262F          JMP     CLOSE   ;CLOSE THE FILE AND EXIT !


;+
;  *** RBLK *** "READ BLOCK" ROUTINE
;  *** WBLK *** "WRITE BLOCK" ROUTINE
;
;    RBLK AND WBLK ARE USED TO READ/WRITE TO/FROM A
;    SPECIFIED VIRTUAL BLOCK NUMBER IN A FILE.
;
```

```
                ;  INPUTS:   HL => FPB
                ;          FBLK =  DESIRED STARTING VIRTUAL BLOCK NUMBER
                ;          FBUF => BLOCK BUFFER
                ;          FXBC = NUMBER OF BYTES TO READ/WRITE
                ;
                ;  OUTPUTS:  A - LOST
                ;            BC,DE - UNCHANGED
                ;            HL => FPB
                ;            FBLK,FBUF,FXBC - UNCHANGED
                ;
                ;  STATUS:   <NC><Z> - NO ERRORS:
                ;              FAUX= NUMBER OF BYTES TRANSFERRED = (FXBC).
                ;            <NC><NZ> - TRANSFER TRUNCATED BY END-OF-FILE
                ;              FAUX= NUMBER OF BYTES TRANSFERRED < (FXBC).
                ;            <C><Z> - VIRTUAL BLOCK NOT WITHIN FILE:
                ;              FAUX UNCHANGED.
                ;            <C><NZ><M> - READ/WRITE ERROR:
                ;              FAUX = NUMBER OF BYTES ATTEMPTED.
                ;-

317F 3E01       WBLK:   MVI     A,1         ;SET WRITE FUNCTION CODE
3181 FE                 DB      OFEH        ;*** SKIP ONE BYTE *** (CPI)
3182 AF         RBLK:   XRA     A           ;SET READ FUNCTION CODE
3183 D5                 PUSH    D
3184 C5                 PUSH    B
3185 CD8F31             CALL    RWBCM       ;DO THE TRANSFER
3188 C1                 POP     B
3189 D1                 POP     D
318A C9                 RET                 ;EXIT !

318B 3E01       IWBK:   MVI     A,1         ;SET WRITE FUNCTION CODE
318D FE                 DB      OFEH        ;*** SKIP ONE BYTE *** (CPI)
318E AF         IRBK:   XRA     A           ;SET READ FUNCTION CODE
318F E5         RWBCM:  PUSH    H           ;SAVE FPB PTR
3190 111C00             LXI     D,FFCN-FPB
3193 19                 DAD     D           ;POINT TO HANDLER FUNCTION CODE
3194 77                 MOV     M,A         ;SET FUNCTION CODE
3195 110200             LXI     D,FBLK-FFCN
3198 19                 DAD     D           ;POINT TO VIRTUAL BLOCK NUMBER
3199 5E                 MOV     E,M         ;GET IT ...
319A 23                 INX     H
319B 56                 MOV     D,M
319C E1                 POP     H           ;GET FPB PTR
319D CDF530             CALL    CBC         ;CALL COMMON ROUTINE
31A0 D8                 RC                  ;<C><Z>: INVALID BLOCK NUMBER !
31A1 012200             LXI     B,FXBC-FPB
31A4 09                 DAD     B           ;POINT TO FXBC
31A5 4E                 MOV     C,M         ;GET REQUESTED BYTE COUNT ...
31A6 23                 INX     H
31A7 46                 MOV     B,M
31A8 C5                 PUSH    B           ;SAVE REQUESTED BYTE COUNT
31A9 72                 MOV     M,D         ;SET BYTE COUNT FOR XFER
31AA 2B                 DCX     H
31AB 73                 MOV     M,E
31AC 01EAFF             LXI     B,FSBK-FXBC
31AF 09                 DAD     B           ;POINT TO STARTING BLOCK NUMBER
31B0 5E                 MOV     E,M         ;GET IT ...
31B1 23                 INX     H
```

```
31B2 56              MOV     D,M
31B3 011100          LXI     B,FBLK-FSBK-1
31B6 09              DAD     B           ;POINT TO VIRTUAL BLOCK NUMBER
31B7 4E              MOV     C,M         ;GET IT ...
31B8 23              INX     H
31B9 46              MOV     B,M
31BA C5              PUSH    B           ;SAVE VIRTUAL BLOCK NUMBER
31BB EB              XCHG
31BC 09              DAD     B           ;COMPUTE ABSOLUTE BLOCK NUMBER
31BD EB              XCHG
31BE 72              MOV     M,D         ;SET ABSOLUTE BLOCK NUMBER ...
31BF 2B              DCX     H
31C0 73              MOV     M,E
31C1 E5              PUSH    H           ;SAVE PTR TO FBLK
31C2 01FCFF          LXI     B,FHAN-FBLK
31C5 09              DAD     B           ;POINT TO FHAN
31C6 5E              MOV     E,M         ;GET HANDLER ADDRESS ...
31C7 23              INX     H
31C8 56              MOV     D,M
31C9 23              INX     H
31CA CD012F          CALL    JMPD        ;PERFORM THE READ/WRITE
31CD E1              POP     H           ;GET PTR TO FBLK
31CE D1              POP     D           ;GET VIRTUAL BLOCK NUMBER
31CF 73              MOV     M,E         ;RESTORE VIRTUAL BLOCK NUMBER ...
31D0 23              INX     H
31D1 72              MOV     M,D
31D2 F5              PUSH    PSW         ;SAVE HANDLER STATUS
31D3 010300          LXI     B,FXBC-FBLK-1
31D6 09              DAD     B           ;POINT TO FXBC
31D7 F1              POP     PSW         ;RESTORE HANDLER STATUS
31D8 C1              POP     B           ;GET REQUESTED BYTE COUNT
31D9 5E              MOV     E,M         ;GET ATTEMPTED BYTE COUNT ...
31DA 23              INX     H
31DB 56              MOV     D,M
31DC 70              MOV     M,B         ;RESTORE REQUESTED BYTE COUNT ...
31DD 2B              DCX     H
31DE 71              MOV     M,C
31DF DAEC31          JC      RWB1        ;ERROR DURING TRANSFER !
31E2 E5              PUSH    H           ;SAVE POINTER TO FXBC
31E3 60              MOV     H,B
31E4 69              MOV     L,C
31E5 CD4D34          CALL    CMPHD       ;CHECK IF ATTEMPTED = REQUESTED
31E8 E1              POP     H           ;RESTORE POINTER TO FXBC
31E9 C3EF31          JMP     RWB2        ;FINISH CLEANUP & EXIT
31EC F6FF    RWB1:   ORI     -1
31EE 37              STC                 ;<C><NZ><M>: I/O ERROR STATUS
31EF F5      RWB2:   PUSH    PSW         ;SAVE STATUS
31F0 01DEFF          LXI     B,FPB-FXBC
31F3 09              DAD     B           ;RESTORE FPB PTR
31F4 F1              POP     PSW         ;RESTORE CONDITION CODES
31F5 C9              RET                 ;EXIT !


;+
;  *** RBLKI *** "READ BLOCK & INCREMENT" ROUTINE
;  *** WBLKI *** "WRITE BLOCK & INCREMENT" ROUTINE
;
;   RBLKI AND WBLKI ARE IDENTICAL IN FUNCTION TO
```

```
                ;   RBLK AND WBLK EXCEPT: IF NO ERRORS OCCURRED, THEN
                ;   FBLK IS SET TO THE NEXT VIRTUAL BLOCK NUMBER
                ;   FOR SEQUENTIAL ACCESS.  IF ANY ERRORS OCCURRED,
                ;   THEN FBLK IS UNCHANGED.
                ;-

31F6 3E01       WBLKI:  MVI     A,1         ;SET WRITE FUNCTION CODE
31F8 FE                 DB      OFEH        ;*** SKIP ONE BYTE *** (CPI)
31F9 AF         RBLKI:  XRA     A           ;SET READ FUNCTION CODE
31FA D5                 PUSH    D
31FB C5                 PUSH    B
31FC CD0632             CALL    RWICM       ;DO THE TRANSFER
31FF C1                 POP     B
3200 D1                 POP     D
3201 C9                 RET                 ;EXIT !

3202 3E01       IWBKI:  MVI     A,1         ;SET WRITE FUNCTION CODE
3204 FE                 DB      OFEH        ;*** SKIP ONE BYTE *** (CPI)
3205 AF         IRBKI:  XRA     A           ;SET READ FUNCTION CODE
3206 CD8F31     RWICM:  CALL    RWBCM       ;DO THE TRANSFER
3209 D8                 RC                  ;INVALID BLOCK OR READ/WRITE ERROR !
320A F5                 PUSH    PSW         ;SAVE CONDITION CODES
320B 011E00             LXI     B,FBLK-FPB
320E 09                 DAD     B           ;POINT TO VIRTUAL BLOCK NUMBER
320F E5                 PUSH    H           ;SAVE PTR TO FBLK
3210 7E                 MOV     A,M         ;GET VIRTUAL BLOCK NUMBER
3211 23                 INX     H
3212 66                 MOV     H,M
3213 6F                 MOV     L,A
3214 EB                 XCHG                ;HL = BYTES TRANSFERED
3215 CDA835             CALL    BC2BK       ;CONVERT TO BLOCKS
3218 19                 DAD     D           ;COMPUTE NEXT VIRTUAL BLOCK NUMBER
3219 EB                 XCHG
321A E1                 POP     H           ;GET PTR TO FBLK
321B 73                 MOV     M,E         ;SET NEXT VIRTUAL BLOCK NUMBER ...
321C 23                 INX     H
321D 72                 MOV     M,D
321E 01E1FF             LXI     B,FPB-FBLK-1
3221 09                 DAD     B           ;RESTORE FPB PTR
3222 F1                 POP     PSW         ;RESTORE CONDITION CODES
3223 C9                 RET                 ;EXIT: EITHER <NC><Z> OR <NC><NZ> !


                ;+
                ;  *** GTBYT *** "GET BYTE" ROUTINE
                ;
                ;   GTBYT IS USED TO READ SEQUENTIAL BYTES FROM AN OPEN
                ;   FILE.
                ;
                ;   RWSEQI MUST HAVE BEEN CALLED BEFORE THE FIRST CALL
                ;   TO GTBYT !
                ;
                ;  INPUTS:  HL => FPB
                ;
                ;  OUTPUTS:  A = THE BYTE, IF NO ERRORS
                ;            BC,DE = UNCHANGED
                ;            HL => FPB
                ;
```

```
                ;  STATUS:   <NC> - NO ERRORS, A = THE BYTE
                ;            <C><Z> - END OF FILE
                ;            <C><NZ><M> - READ ERROR
                ;-

3224 E1         GB1:    POP     H       ;RESTORE FPB PTR
3225 CDF931             CALL    RBLKI   ;READ NEXT BUFFERFULL
3228 D8                 RC              ;END-OF-FILE OR READ ERROR !
3229 CDD930             CALL    ZPTR    ;RESET RELATIVE BUFFER POINTER
322C E5         GTBYT:  PUSH    H       ;SAVE FPB PNTR
322D CD9232             CALL    BCHK    ;NEED TO READ NEXT BUFFERFULL ?
3230 D22432             JNC     GB1     ;YES !
3233 AF                 XRA     A       ;SET GOOD CODES
3234 7E                 MOV     A,M     ;GET THE BYTE
3235 E1                 POP     H       ;RESTORE FPB PTR
3236 C9                 RET             ;<NC>: NO ERROR !


                ;+
                ;  *** PTBYT *** "PUT BYTE" ROUTINE
                ;
                ;   PTBYT IS USED TO WRITE SEQUENTIAL BYTES TO AN OPEN FILE.
                ;
                ;   INSEQO MUST BE CALLED AFTER OPEN AND BEFORE THE FIRST
                ;   CALL TO PTBYT !
                ;
                ;  INPUTS:  A = THE BYTE
                ;           HL => FPB
                ;
                ;  OUTPUTS: A = THE BYTE
                ;           BC, DE - UNCHANGED
                ;           HL => FPB
                ;
                ;  STATUS:  <NC> - NO ERRORS
                ;           <C><Z> - FILE FULL, BYTE NOT WRITTEN
                ;           <C><NZ><M> - WRITE ERROR
                ;-

3237 E1         PB1:    POP     H       ;GET THE BYTE
3238 EB                 XCHG            ;D = THE BYTE
3239 E3                 XTHL            ;GET FPB PTR, SAVE ORIG. DE
323A C5                 PUSH    B       ;SAVE BC
323B D5                 PUSH    D       ;SAVE THE BYTE
323C CD0232             CALL    IWBKI   ;WRITE THE BUFFER OUT
323F D4F530             CNC     CBC     ;CALL COMMON ROUTINE IF NO ERROR
3242 D1                 POP     D       ;GET THE BYTE BACK
3243 7A                 MOV     A,D     ;... INTO A
3244 C1                 POP     B       ;RESTORE BC
3245 D1                 POP     D       ;RESTORE DE
3246 D8                 RC              ;WRITE ERROR OR FILE FULL !
3247 CDD930             CALL    ZPTR    ;RESET RELATIVE BUFFER POINTER
324A E5         PTBYT:  PUSH    H       ;SAVE FPB PTR
324B F5                 PUSH    PSW     ;SAVE THE BYTE
324C CD9232             CALL    BCHK    ;NEED TO WRITE OUT BLOCK BUFFER ?
324F D23732             JNC     PB1     ;YES !
3252 F1                 POP     PSW     ;GET THE BYTE
3253 77                 MOV     M,A     ;PUT BYTE INTO BUFFER
3254 E1                 POP     H       ;RESTORE FPB PTR
```

```
3255 A7                 ANA     A       ;SET GOOD CODES
3256 C9                 RET             ;<NC>: NO ERRORS !

                ;+
                ; *** GAREC *** "GET ASCII RECORD" ROUTINE
                ;
                ;  GAREC IS USED TO READ SEQUENTIAL RECORDS FROM AN
                ;  ASCII FILE.  A RECORD IS A STRING OF ASCII CHARACTERS
                ;  TERMINATED BY EITHER A LINE FEED (10.) OR A
                ;  FORM FEED (12.) .
                ;
                ;  RWSEQI MUST HAVE BEEN CALLED BEFORE THE FIRST CALL
                ;  TO GAREC !
                ;
                ; INPUTS:  HL => FPB
                ;          BC => RECORD BUFFER
                ;          DE =  RECORD BUFFER LENGTH (BYTES)
                ;
                ; OUTPUTS: HL => FPB
                ;          BC => PAST LAST BYTE STORED
                ;          DE = NUMBER OF BYTES READ
                ;          A - LOST
                ;
                ; STATUS:  <NC> - NO ERRORS
                ;          <C><Z> - END OF FILE
                ;          <C><NZ><M> - READ ERROR
                ;          <C><NZ><P> - BUFFER WAS FILLED, BUT A VALID
                ;                       TERMINATOR WAS NOT SEEN.  THE NEXT
                ;                       CALL TO GAREC WILL START WITH THE
                ;                       NEXT SEQUENTIAL BYTE.
                ;-

3257 D5         GAREC:  PUSH    D       ;SAVE BUFFER LENGTH
3258 CD2C32     GAR1:   CALL    GTBYT   ;GET NEXT BYTE
325B DA7232             JC      GAR2    ;ERROR OR EOF !
325E 02                 STAX    B       ;STORE THE BYTE
325F 03                 INX     B
3260 1B                 DCX     D       ;COUNT IT
3261 FE0A               CPI     10      ;LINE FEED ?
3263 CA7232             JZ      GAR2    ;YES: <NC>: NO ERRORS !
3266 FE0C               CPI     12      ;FORM FEED ?
3268 CA7232             JZ      GAR2    ;YES: <NC>: NO ERRORS !
326B 7A                 MOV     A,D
326C B3                 ORA     E       ;RECORD BUFFER FULL ?
326D C25832             JNZ     GAR1    ;NO:  GET ANOTHER BYTE !
3270 3C                 INR     A
3271 37                 STC             ;<C><NZ><P>: LINE TOO LONG
3272 E3         GAR2:   XTHL            ;GET BUFFER LENGTH
3273 F5                 PUSH    PSW     ;SAVE STATUS
3274 CD5934             CALL    SUBHD   ;COMPUTE BYTE COUNT
3277 EB                 XCHG            ;DE = BYTE COUNT
3278 F1                 POP     PSW     ;RESTORE STATUS
3279 E1                 POP     H       ;RESTORE FPB PTR
327A C9                 RET             ;EXIT !

                ;+
```

```
                ;  *** PVREC *** "PUT VARIABLE LENGTH RECORD" ROUTINE
                ;
                ;   PVREC IS USED TO PUT A VARIABLE LENGTH RECORD INTO A
                ;   SYSTEM STANDARD 'VARIABLE LENGTH RECORD, SEQUENTIAL' FILE.
                ;   WITHIN THE FILE, EACH RECORD CONSISTS OF A TWO BYTE BYTE
                ;   COUNT, LOW BYTE FIRST, FOLLOWED BY THAT NUMBER OF DATA
                ;   BYTES.
                ;
                ;  INPUTS:  HL => FPB
                ;           BC => RECORD BUFFER
                ;           DE = RECORD LENGTH (BYTES)
                ;
                ;  OUTPUTS: HL => FPB
                ;           BC => PAST LAST BYTE IN RECORD BUFFER
                ;           DE = 0 IF NO ERRORS
                ;           A - LOST
                ;
                ;  STATUS:  <NC> - NO ERRORS
                ;           <C><Z> - TRANSFER TERMINATED BY END OF FILE - FILE FULL
                ;           <C><NZ><M> - WRITE ERROR
                ;-

327B 7B         PVREC:  MOV     A,E
327C CD4A32             CALL    PTBYT   ;PUT LOBYTE OF BYTE COUNT
327F D8                 RC              ;ERROR OR EOF !
3280 7A                 MOV     A,D
3281 CD4A32             CALL    PTBYT   ;PUT HIBYTE OF BYTE COUNT
3284 D8                 RC              ;ERROR OF EOF !
                ;  FALL INTO PTREC !


                ;+
                ;  *** PTREC *** "PUT UNFORMATTED RECORD" ROUTINE
                ;
                ;   PTREC IS USED TO PUT AN 'UNFORMATTED' RECORD INTO A
                ;   SEQUENTIAL FILE.  OPERATION OF PTREC IS IDENTICAL TO
                ;   PVREC, ABOVE, EXCEPT THAT THE TWO BYTE BYTE COUNT
                ;   IS NOT WRITTEN INTO THE FILE.
                ;-

3285 0A         PTREC:  LDAX    B       ;GET NEXT BYTE
3286 CD4A32             CALL    PTBYT   ;PUT THE BYTE
3289 D8                 RC              ;ERROR OR EOF !
328A 03                 INX     B
328B 1B                 DCX     D       ;COUNT THE BYTE
328C 7A                 MOV     A,D
328D B3                 ORA     E       ;RECORD ALL DONE ?
328E C28532             JNZ     PTREC   ;NO:  PUT ANOTHER BYTE !
3291 C9                 RET             ;<NC>:  NO ERRORS !


                ;+
                ;  * BCHK *
                ;-

3292 D5         BCHK:   PUSH    D
3293 C5                 PUSH    B
3294 111800             LXI     D,FAUX-FPB
```

```
3297 19                 DAD     D       ;POINT TO AUX. BYTE COUNT
3298 5E                 MOV     E,M     ;GET IT ...
3299 23                 INX     H
329A 56                 MOV     D,M
329B 010B00             LXI     B,FPTR-FAUX-1
329E 09                 DAD     B       ;POINT TO RELATIVE BUFFER POINTER
329F E5                 PUSH    H       ;SAVE POINTER TO FPTR
32A0 7E                 MOV     A,M     ;GET IT ...
32A1 23                 INX     H
32A2 66                 MOV     H,M
32A3 6F                 MOV     L,A
32A4 EB                 XCHG
32A5 CD5334             CALL    CMPDH   ;NEED TO DO A TRANSFER ?
32A8 E1                 POP     H       ;GET POINTER TO FPTR
32A9 D2BB32             JNC     BCHK1   ;<NC>: NEED A TRANSFER !
32AC 13                 INX     D       ;INCREMENT RELATIVE BUFFER POINTER
32AD 73                 MOV     M,E     ;SAVE UPDATED PTR ...
32AE 23                 INX     H
32AF 72                 MOV     M,D
32B0 1B                 DCX     D       ;RESTORE RELATIVE BUFFER POINTER
32B1 01FBFF             LXI     B,FBUF-FPTR-1
32B4 09                 DAD     B       ;POINT TO BUFFER PTR
32B5 7E                 MOV     A,M     ;GET IT ...
32B6 23                 INX     H
32B7 66                 MOV     H,M
32B8 6F                 MOV     L,A
32B9 19                 DAD     D       ;INDEX INTO BUFFER
32BA 37                 STC             ;<C>: NO TRANSFER NEEDED
32BB C1         BCHK1:  POP     B
32BC D1                 POP     D
32BD C9                 RET             ;<C>: NO TRANSFER , <NC>: TRANSFER !

                ; FCS COMMAND INPUT ROUTINE

32BE 360D       ESCG:   MVI     M,13    ;SETUP INPUT FLAG
32C0 7D                 MOV     A,L
32C1 21F981             LXI     H,LOFL  ;SET UP FCS OUTPUT TO PORT
32C4 360E               MVI     M,14
32C6 C3D132             JMP     ESCDG   ; CONTINUE WITH COMMON

32C9 360D       ESCD:   MVI     M,13    ;SETUP INPUT FLAG
32CB 7D                 MOV     A,L
32CC 21F981             LXI     H,LOFL  ;SET UP FCS OUTPUT FLAG
32CF 3600               MVI     M,0     ; TO BE CRT
32D1 FEDF       ESCDG:  CPI     KBDFL AND 255
32D3 21E181             LXI     H,FCSFL ;SET UP FCS ECHO FLAG
32D6 3600               MVI     M,0     ;SET FOR ECHO TO SCREEN
32D8 CAE932             JZ      L1      ;YES KEYBOARD!
32DB FEF1               CPI     BASFL AND 255
32DD 360C               MVI     M,12    ;DON'T ECHO IF BASIC INPUT
32DF CAE932             JZ      L1      ;YES BASIC !
32E2 360E               MVI     M,14    ; ECHO TO OUTPUT PORT
32E4 21F981             LXI     H,LOFL  ;SET LO TO OUTPUT PORT
32E7 360E               MVI     M,14
32E9 214780     L1:     LXI     H,BUFP  ;SET BUFFER PNTR
32EC 22F681             SHLD    TEMP4   ;SAVE IT
```

```
32EF CDA526          CALL    RESET   ;
32F2 3E0B            MVI     A,11
32F4 218233          LXI     H,PRMPT ;POINT TO PROMPT
32F7 CD7933    L2:   CALL    FCSOUT
32FA 7E              MOV     A,M
32FB A7              ANA     A
32FC C8              RZ              ;FINISHED WITH PROMPT!
32FD 23              INX     H
32FE C3F732          JMP     L2      ;LOOP!

3301 7B        FCSX: MOV     A,E     ;GET CHARACTER
3302 FE1B            CPI     27      ;IS IT ESCAPE ?
3304 CA7233          JZ      EFCS    ;YES!
3307 FE7F            CPI     127     ;RUB OUT ?
3309 CA3433          JZ      L4      ;YES !
330C FE20            CPI     32      ;CONTROL CHARACTER ?
330E DA2A33          JC      L3      ;YES!
3311 D6FF            SUI     255     ;  IS IT BASIC EXIT ?
3313 CA7233          JZ      EFCS    ;  YES, EXIT
3316 2AF681          LHLD    TEMP4
3319 3E81            MVI     A,(BUFP AND 255)+58
331B BD              CMP     L       ;BUFFER FULL ?
331C 3E07            MVI     A,7
331E CA7933          JZ      FCSOUT  ;YES: BEEP !
3321 73              MOV     M,E     ;STORE CHARACTER IN BUFFER
3322 23              INX     H       ;BUMP POINTER
3323 22F681          SHLD    TEMP4   ;SAVE BUFFER PNTR
3326 73              MOV     A,E     ;GET CHAR.
3327 C37933          JMP     FCSOUT  ;GO ECHO IT
332A FE0B      L3:   CPI     11      ;ERASE LINE ?
332C CAE932          JZ      L1      ;YES!
332F FE1A            CPI     26      ;CURSOR LEFT ?
3331 C24E33          JNZ     ELIN    ;NO: TERMINATE THE LINE!
3334 2AF681    L4:   LHLD    TEMP4
3337 3E47            MVI     A,BUFP AND 255
3339 BD              CMP     L       ;BUFFER EMPTY ?
333A CAE932          JZ      L1      ;YES: ERASE LINE!
333D 2B              DCX     H       ;BUMP PNTR BACK
333E 22F681          SHLD    TEMP4   ;SAVE BUFFER PNTR
3341 CD4933          CALL    L5      ;PRINT CURSOR LEFT
3344 3E20            MVI     A,32    ;SPACE
3346 CD7933          CALL    FCSOUT  ;PRINT IT
3349 3E1A      L5:   MVI     A,26    ;CURSOR LEFT
334B C37933          JMP     FCSOUT  ;PRINT IT & RETURN

334E CD7933    ELIN: CALL    FCSOUT  ;ECHO
3351 3E0A            MVI     A,10
3353 CD7933          CALL    FCSOUT
3356 7E              MOV     A,M
3357 F5              PUSH    PSW
3358 3600            MVI     M,0
335A 22DE80          SHLD    TEMPHL
335D 2AF681          LHLD    TEMP4
3360 3600            MVI     M,0
3362 214780          LXI     H,BUFP  ;SET BUFFER PNTR
3365 7E              MOV     A,M
3366 A7              ANA     A       ;EMPTY ?
3367 C42A26          CNZ     FCSEM   ;NO: GO TO FCS!
```

```
336A 2ADE80             LHLD    TEMPHL
336D F1                 POP     PSW
336E 77                 MOV     M,A
336F C3E932             JMP     L1

3372 77         EFCS:   MOV     M,A     ;SETUP INPUT FLAG
3373 21F981             LXI     H,LOFL  ; CLEAR LO FLAG
3376 3600               MVI     M,0
3378 C9                 RET

3379 E5         FCSOUT: PUSH    H       ;SAVE MEMORY POINTER
337A 21E181             LXI     H,FCSFL ;SETUP FOR FCS OUTPUT
337D CD8C39             CALL    PROCES  ;ECHO CHAR IN A REG.
3380 E1                 POP     H       ;RESTORE MEMORY POINTER
3381 C9                 RET

3382 06034643 PRMPT:    DB 6,3,'FCS>',6,2,0
3386 533E0602
338A 00
                ; UTILITY SUBROUTINES - VERSION 4.0
                [redacted]

     (0008)     SSIDA   EQU     08H
     (0010)     SSOBE   EQU     10H

338B 3E0D       CRLF:   MVI     A,13    ;ASCII CR
338D CD9233             CALL    LO
3390 3E0A               MVI     A,10    ;ASCII LF

3392 CDD03F     LO:     CALL    SAVE    ;SAVE ALL REGS.
3395 21F981             LXI     H,LOFL
3398 C38C39             JMP     PROCES

339B F5         LBYT:   PUSH    PSW     ;SAVE BYTE
339C 0F                 RRC
339D 0F                 RRC
339E 0F                 RRC
339F 0F                 RRC
33A0 CDA433             CALL    LHXD    ;LIST 1ST HEX DIGIT
33A3 F1                 POP     PSW     ;RETRIEVE BYTE
33A4 CDAA33     LHXD:   CALL    B2HEX   ;CONVERT VALUE TO HEX DIGIT
33A7 C39233             JMP     LO      ;LIST DIGIT AND RETURN!

33AA E60F       B2HEX:  ANI     0FH     ;CLEANSE VALUE
33AC C690               ADI     90H
33AE 27                 DAA
33AF CE40               ACI     40H
33B1 27                 DAA
33B2 C9                 RET             ;RETURN: A=ASCII HEX DIGIT

33B3 D630       NIBL:   SUI     '0'
33B5 D8                 RC              ;NOT HEX: RETURN <C>
33B6 FE0A               CPI     10
33B8 3F                 CMC
33B9 D0                 RNC             ;VALID: RETURN <NC>
```

```
33BA FE11               CPI     'A'-'0'
33BC D8                 RC                      ;NOT HEX: RETURN <C>
33BD D607               SUI     'A'-'9'-1
33BF FE10               CPI     16
33C1 3F                 CMC
33C2 C9                 RET                     ;HEX: <NC>, NOT HEX: <C>

33C3 DB03       S1OUT:  IN      STAT5           ;GET STATUS
33C5 E610               ANI     SSOBE           ;CAN WE SEND ?
33C7 CAC333             JZ      S1OUT           ;NO: WAIT!
33CA DB01               IN      EXTIN           ;IS CLEAR TO SEND HI ?
33CC E680               ANI     80H
33CE CAC333             JZ      S1OUT
33D1 7B                 MOV     A,E             ;GET CHARACTER TO SEND
33D2 D306               OUT     TXBUF           ;SEND THE CHARACTER
33D4 C9                 RET                     ;RETURN

33D5 E5         RTST:   PUSH    H
33D6 D5                 PUSH    D
33D7 C5                 PUSH    B
33D8 3AFF81     RTST2:  LDA     READY
33DB FE50               CPI     50H
33DD CAD833             JZ      RTST2
33E0 FE80               CPI     80H
33E2 3AFE81             LDA     KBCHA
33E5 C1                 POP     B
33E6 D1                 POP     D
33E7 E1                 POP     H
33E8 C9                 RET

                ; SEND A MESSAGE STRING TO "LO" DEVICE.  BYTE VALUE OF
                ; 239 DECIMAL TERMINATES THE STRING.
                ; ALSO SUPPORTS REPEAT LOOPS OF THE FORM :
                ;       ...,237,N,D1,D2,...,DM,238,...
                ; WHERE N IS THE REPEAT COUNT FOR THE STRING OF
                ; BYTES:  D1,D2,...,DM .

33E9 7E         BSTR:   MOV     A,M
33EA 23                 INX     H
33EB FEEF               CPI     239
33ED C8                 RZ
33EE CD6B39             CALL    CRTUBE
33F1 C3E933             JMP     BSTR

33F4 7E         OSTR:   MOV     A,M             ;GET NEXT BYTE
33F5 23                 INX     H
33F6 FEED               CPI     237             ;SPECIAL CODE ?
33F8 DA0434             JC      OSTR1           ;A<237:  GO SEND BYTE
33FB CA0A34             JZ      OSTR2           ;A=237:  START REPEAT LOOP
33FE FEEF               CPI     239             ;SPECIAL CODE ?
3400 C8                 RZ                      ;A=239:  END:  RETURN
3401 DA1034             JC      OSTR4           ;A=238:  END OF REPEAT LOOP
3404 CD9233     OSTR1:  CALL    LO              ;SEND BYTE
3407 C3F433             JMP     OSTR            ;GET NEXT BYTE
340A 56         OSTR2:  MOV     D,M             ;GET REPEAT COUNT
340B 23                 INX     H
340C E5         OSTR3:  PUSH    H               ;SAVE START POINTER
```

```
340D C3F433             JMP     OSTR        ;GET NEXT BYTE
3410 15         OSTR4:  DCR     D           ;FINISHED REPEAT ?
3411 CA1834             JZ      OSTR5       ;YES!
3414 E1                 POP     H           ;RESTORE START POINTER
3415 C30C34             JMP     OSTR3       ;REPEAT AGAIN!
3418 F1         OSTR5:  POP     PSW         ;CLEAN THE STACK
3419 C3F433             JMP     OSTR        ;GET NEXT BYTE!

                ; SHORT WAIT: 0.5 MS. / COUNT :

                ;STIM    EQU     50          ;FOR 'WAIT' STATE IN ROM
    (003F)      STIM    EQU     63          ;FOR NO 'WAIT' STATE IN ROM

341C F5         WATS:   PUSH    PSW
341D 3E3F               MVI     A,STIM
341F 3D         WS1:    DCR     A
3420 C21F34             JNZ     WS1
3423 F1                 POP     PSW
3424 3D                 DCR     A
3425 C21C34             JNZ     WATS
3428 C9                 RET

                ; LONG WAIT: 20 MS. / COUNT :

                ;LTIM    EQU     2347        ;FOR 'WAIT' STATE IN ROM
    (0B5A)      LTIM    EQU     2906        ;FOR NO 'WAIT' STATE IN ROM

3429 E5         WATL:   PUSH    H
342A 215A0B     WL1:    LXI     H,LTIM
342D 2D         WL2:    DCR     L
342E C22D34             JNZ     WL2
3431 25                 DCR     H
3432 C22D34             JNZ     WL2
3435 3D                 DCR     A
3436 C22A34             JNZ     WL1
3439 E1                 POP     H
343A C9                 RET

343B 7E         MOVDH:  MOV     A,M
343C 12                 STAX    D
343D 23                 INX     H
343E 13                 INX     D
343F 05                 DCR     B
3440 C23B34             JNZ     MOVDH
3443 C9                 RET

3444 1A         MOVHD:  LDAX    D
3445 77                 MOV     M,A
3446 13                 INX     D
3447 23                 INX     H
3448 05                 DCR     B
3449 C24434             JNZ     MOVHD
344C C9                 RET

344D 7C         CMPHD:  MOV     A,H
```

```
344E BA             CMP     D
344F C0             RNZ
3450 7D             MOV     A,L
3451 BB             CMP     E
3452 C9             RET

3453 7A     CMPDH:  MOV     A,D
3454 BC             CMP     H
3455 C0             RNZ
3456 7B             MOV     A,E
3457 BD             CMP     L
3458 C9             RET

3459 7D     SUBHD:  MOV     A,L
345A 93             SUB     E
345B 6F             MOV     L,A
345C 7C             MOV     A,H
345D 9A             SBB     D
345E 67             MOV     H,A
345F C9             RET

3460 7E     SPNOR:  MOV     A,M     ;GET NEXT CHAR.
3461 23             INX     H
3462 FE20           CPI     32      ;IS IT A SPACE ?
3464 CA6034         JZ      SPNOR   ;YES: IGNORE IT!
3467 2B             DCX     H
3468 A7             ANA     A       ;TEST FOR END OF LINE
3469 C9             RET             ;RETURN TO CALLER

346A 7E     LET:    MOV     A,M     ;GET CHAR.
346B FE5B           CPI     'Z'+1   ;IS IT A LETTER ?
346D D0             RNC             ;NO!
346E FE41           CPI     'A'     ;CHECK OTHER END
3470 3F             CMC             ;SET C CORRECTLY
3471 C9             RET             ;RETURN: C SET IF LETTER

3472 CD6A34 LODQ:   CALL    LET     ;IS IT A LETTER ?
3475 D8             RC              ;YES: RETURN!

3476 7E     DIG:    MOV     A,M     ;GET CHAR.
3477 FE3A           CPI     '9'+1   ;IS IT A DIGIT ?
3479 D0             RNC             ;NO!
347A FE30           CPI     '0'     ;CHECK OTHER END
347C 3F             CMC             ;SET C CORRECTLY
347D C9             RET             ;RETURN: C SET IF DIGIT

347E CD6A34 LTNOR:  CALL    LET     ;IS NEXT CHAR. A LETTER ?
3481 23             INX     H
3482 DA7E34         JC      LTNOR   ;YES: IGNORE IT!
3485 2B             DCX     H
3486 A7             ANA     A       ;TEST FOR END OF LINE
3487 C9             RET             ;RETURN

3488 CD6034 QCMA:   CALL    SPNOR   ;GET FIRST NON-SPACE
348B D62C           SUI     ','     ;CHECK FOR COMMA
348D A7             ANA     A       ;CLEAR C-BIT
348E C0             RNZ             ;NO COMMA: EXIT <NC>
348F 23             INX     H       ;SKIP OVER COMMA
```

```
3490 CD6034            CALL    SPNOR   ;GET NEXT NON-SPACE
3493 37                STC             ;INDICATE COMMA SEEN
3494 C9                RET             ;COMMA: EXIT <C>

3495 48        MSTR:   MOV     C,B     ;COPY ORIG. COUNT
3496 CD7234    MST01:  CALL    LODG    ;LETTER OR DIGIT ?
3499 D2A334            JNC     MST02   ;NO!
349C 12                STAX    D       ;STORE CHAR.
349D 23                INX     H
349E 13                INX     D
349F 05                DCR     B       ;ENOUGH ?
34A0 C29634            JNZ     MST01   ;NO: LOOP!
34A3 79        MST02:  MOV     A,C     ;GET ORIG. COUNT
34A4 90                SUB     B       ;COMPUTE BYTE COUNT
34A5 F5                PUSH    PSW     ;SAVE IT
34A6 3E20              MVI     A,32    ;SET SPACE
34A8 C3AD34            JMP     MST04
34AB 12        MST03:  STAX    D       ;STORE A SPACE
34AC 13                INX     D
34AD 05        MST04:  DCR     B       ;ANY MORE TO DO ?
34AE F2AB34            JP      MST03   ;YES: LOOP!
34B1 F1                POP     PSW     ;RESTORE BYTE COUNT
34B2 C9                RET             ;RETURN

34B3 3E20      PSPAC:  MVI     A,32    ;SET SPACE
34B5 C39233            JMP     LO      ;PRINT A SPACE

34B8 3E3A      PCOLN:  MVI     A,':'   ;SET COLON
34BA C39233            JMP     LO      ;PRINT A COLON

34BD CDB334    PSSTR:  CALL    PSPAC   ;PRINT A SPACE
34C0 7E        PSTR:   MOV     A,M     ;GET NEXT CHARACTER
34C1 23                INX     H
34C2 CD9233            CALL    LO      ;PRINT IT
34C5 05                DCR     B       ;ENOUGH ?
34C6 C2C034            JNZ     PSTR    ;NO: LOOP!
34C9 C9                RET             ;RETURN

34CA CDB334    PSBYT:  CALL    PSPAC   ;PRINT A SPACE
34CD 7E        PBYT:   MOV     A,M     ;GET NEXT BYTE
34CE 23                INX     H
34CF C39B33            JMP     LBYT    ;PRINT THE BYTE

34D2 CDB334    P2SNUM: CALL    PSPAC   ;PRINT A SPACE
34D5 CDB334    PSNUM:  CALL    PSPAC   ;PRINT A SPACE
34D8 7E        PNUM:   MOV     A,M     ;GET LOBYTE
34D9 F5                PUSH    PSW     ;SAVE LOBYTE
34DA 23                INX     H
34DB 7E                MOV     A,M     ;GET HIBYTE
34DC 23                INX     H
34DD CD9B33            CALL    LBYT    ;PRINT HIBYTE
34E0 F1                POP     PSW     ;GET LOBYTE
34E1 C39B33            JMP     LBYT    ;PRINT LOBYTE

34E4 110000    GN1Z:   LXI     D,0     ;SET ZERO
34E7 CD8834    GN1D:   CALL    QCMA    ;LOOK FOR COMMA
34EA 3F                CMC
34EB D0                RNC             ;NO NUMBER: EXIT <NC>
```

```
34EC CDF934             CALL    GN2D    ;LOOK FOR NUMBER
34EF D0                 RNC             ;NO NUMBER: EXIT <NC>
34F0 F5                 PUSH    PSW     ;SAVE FLAGS
34F1 CD8834             CALL    GCMA    ;LOOK FOR TRAILING COMMA
34F4 F1                 POP     PSW     ;RESTORE CODES
34F5 C9                 RET             ;GOT NUMBER: EXIT <C>

34F6 110000     GN2Z:   LXI     D,0     ;SET ZERO
34F9 CD6034     GN2D:   CALL    SPNOR   ;GET 1ST NON-SPACE
34FC CDB333             CALL    NIBL    ;CHECK IF HEX
34FF 3F                 CMC
3500 D0                 RNC             ;NO NUMBER: EXIT <NC>
3501 110000             LXI     D,0     ;INITIALIZE NUMBER TO ZERO
3504 23         GO1:    INX     H       ;SKIP CHARACTER
3505 EB                 XCHG
3506 29                 DAD     H       ;MULTIPLY ...
3507 29                 DAD     H       ;... BY ...
3508 29                 DAD     H       ;... SIXTEEN ...
3509 29                 DAD     H
350A B5                 ORA     L       ;MERGE NEW NIBBLE
350B 6F                 MOV     L,A
350C EB                 XCHG
350D 7E                 MOV     A,M     ;GET NEXT CHARACTER
350E CDB333             CALL    NIBL    ;CHECK IF HEX
3511 D20435             JNC     GO1     ;YES: LOOP!
3514 7A                 MOV     A,D     ;TEST ...
3515 B3                 ORA     E       ;... NUMBER
3516 37                 STC             ;INDICATE NUMBER SEEN
3517 C9                 RET             ;GOT NUMBER: EXIT <C>

                ; GENERAL MATH ROUTINES :

3518 85         ADHLA:  ADD     L
3519 6F                 MOV     L,A
351A D0                 RNC
351B 24                 INR     H
351C C9                 RET

351D 7B         ANHD:   MOV     A,E
351E A5                 ANA     L
351F 6F                 MOV     L,A
3520 7A                 MOV     A,D
3521 A4                 ANA     H
3522 67                 MOV     H,A
3523 C9                 RET

3524 2B         NEGH:   DCX     H
3525 7D         NOTH:   MOV     A,L
3526 2F                 CMA
3527 6F                 MOV     L,A
3528 7C                 MOV     A,H
3529 2F                 CMA
352A 67                 MOV     H,A
352B C9                 RET

352C 7B         ORHD:   MOV     A,E
352D B5                 ORA     L
```

```
352E 6F                 MOV     L,A
352F 7A                 MOV     A,D
3530 B4                 ORA     H
3531 67                 MOV     H,A
3532 C9                 RET

3533 7B         XORHD:  MOV     A,E
3534 AD                 XRA     L
3535 6F                 MOV     L,A
3536 7A                 MOV     A,D
3537 AC                 XRA     H
3538 67                 MOV     H,A
3539 C9                 RET

353A CD5435     SHLHD:  CALL    SCMN
353D C8                 RZ
353E 29         SL1:    DAD     H
353F 1D                 DCR     E
3540 C23E35             JNZ     SL1
3543 C9                 RET

3544 CD5435     SHRHD:  CALL    SCMN
3547 C8                 RZ
3548 AF         SR1:    XRA     A
3549 7C                 MOV     A,H
354A 1F                 RAR
354B 67                 MOV     H,A
354C 7D                 MOV     A,L
354D 1F                 RAR
354E 6F                 MOV     L,A
354F 1D                 DCR     E
3550 C24835             JNZ     SR1
3553 C9                 RET

3554 EB         SCMN:   XCHG
3555 3EF0               MVI     A,0F0H
3557 A3                 ANA     E
3558 B2                 ORA     D
3559 C25E35             JNZ     ZH
355C AB                 XRA     E
355D C9                 RET
355E AF         ZH:     XRA     A
355F 67                 MOV     H,A
3560 6F                 MOV     L,A
3561 C9                 RET

3562 C5         MULHD:  PUSH    B
3563 44                 MOV     B,H
3564 4D                 MOV     C,L
3565 210000             LXI     H,0
3568 3E10               MVI     A,16
356A F5         M1:     PUSH    PSW
356B 29                 DAD     H
356C 7B                 MOV     A,E
356D 17                 RAL
356E 5F                 MOV     E,A
356F 7A                 MOV     A,D
3570 17                 RAL
```

```
3571 57              MOV     D,A
3572 D27A35          JNC     M2
3575 09              DAD     B
3576 D27A35          JNC     M2
3579 13              INX     D
357A F1        M2:   POP     PSW
357B 3D              DCR     A
357C C26A35          JNZ     M1
357F C1              POP     B
3580 C9              RET

3581 C5        DIVHD: PUSH   B
3582 EB              XCHG
3583 42              MOV     B,D
3584 4B              MOV     C,E
3585 110000          LXI     D,0
3588 3E10            MVI     A,16
358A F5        D1:   PUSH    PSW
358B 29              DAD     H
358C 7B              MOV     A,E
358D 17              RAL
358E 5F              MOV     E,A
358F 7A              MOV     A,D
3590 17              RAL
3591 57              MOV     D,A
3592 7B              MOV     A,E
3593 91              SUB     C
3594 5F              MOV     E,A
3595 7A              MOV     A,D
3596 98              SBB     B
3597 57              MOV     D,A
3598 D29F35          JNC     D2
359B EB              XCHG
359C 09              DAD     B
359D EB              XCHG
359E 23              INX     H
359F F1        D2:   POP     PSW
35A0 3D              DCR     A
35A1 C28A35          JNZ     D1
35A4 C1              POP     B
35A5 C32535          JMP     NOTH

                ;
                ; BLOCK STRUCTURED DEVICE HANDLER UTILITY
                ; SUBROUTINES - VERSION 1.0
                ;
                ; COPYRIGHT (C) 1977
                ; BY INTELLIGENT SYSTEMS CORPORATION
                ;

                ; CONSTANTS :

(0003)          CRTR    EQU     3       ;"CHUNK" RETRY COUNT

                ;+
                ; BC2BK - CONVERT TOTAL BYTE COUNT TO NUMBER OF
```

```
                ;           BLOCKS AND LAST BLOCK BYTE COUNT.
                ; INPUTS  - H&L: TOTAL BYTE COUNT
                ; OUTPUTS - H&L: NUMBER OF BLOCKS
                ;           C  : LAST BLOCK BYTE COUNT
                ;-

35A8 7C         BC2BK:  MOV     A,H
35A9 B5                 ORA     L
35AA CAB235             JZ      BCO1
35AD 2B                 DCX     H
35AE 3E7F               MVI     A,7FH
35B0 A5                 ANA     L
35B1 3C                 INR     A
35B2 4F         BCO1:   MOV     C,A
35B3 AF                 XRA     A
35B4 29                 DAD     H
35B5 17                 RAL
35B6 6C                 MOV     L,H
35B7 67                 MOV     H,A
35B8 23                 INX     H
35B9 C9                 RET

                ;+
                ; BK2BC - CONVERT NUMBER OF BLOCKS AND LAST BLOCK
                ;         BYTE COUNT TO TOTAL BYTE COUNT.
                ; INPUTS  - H&L: NUMBER OF BLOCKS
                ;           C  : LAST BLOCK BYTE COUNT
                ; OUTPUTS - H&L: TOTAL BYTE COUNT
                ;-

35BA 2B         BK2BC:  DCX     H
35BB 7C                 MOV     A,H
35BC 1F                 RAR
35BD 7D                 MOV     A,L
35BE 1F                 RAR
35BF 47                 MOV     B,A
35C0 3E00               MVI     A,0
35C2 67                 MOV     H,A
35C3 1F                 RAR
35C4 6F                 MOV     L,A
35C5 09                 DAD     B
35C6 C9                 RET

35C7 C1         BSBO1:  POP     B       ;GET CALLER'S RETURN ADDR.
35C8 E5                 PUSH    H       ;SAVE PBLK PNTR
35C9 D5                 PUSH    D       ;SAVE DEVICE NAME
35CA 11F135             LXI     D,BSO2
35CD D5                 PUSH    D       ;SET RETURN ADDRESS
35CE C5                 PUSH    B       ;SAVE CALLER'S RETURN ADDR.
35CF 4F                 MOV     C,A     ;COPY NUMBER OF UNITS
35D0 11E580             LXI     D,TFCN  ;POINT TO TEMP STORAGE AREA
35D3 0608               MVI     B,8     ;SET COUNT
35D5 CD3B34             CALL    MOVDH   ;MOVE PARAMETERS
35D8 E1                 POP     H       ;GET CALLER'S RETURN ADDR.
35D9 3AE680             LDA     TDRV    ;GET DRIVE NUMBER
35DC 0606               MVI     B,EIVU  ;SET ERROR CODE
```

```
35DE B9                 CMP     C       ;VALID UNIT NUMBER ?
35DF D0                 RNC             ;NO: ERROR EXIT!
35E0 3AE580             LDA     TFCN    ;GET FUNCTION CODE
35E3 FE02               CPI     2       ;IS IT VALID ?
35E5 DAED35             JC      BS01    ;YES!
35E8 0603               MVI     B,EIVF  ;SET ERROR CODE
35EA FEFC               CPI     -4      ;IS IT VALID ?
35EC D8                 RC              ;NO: ERROR EXIT!
35ED 0600       BS01:   MVI     B,SOK   ;SET GOOD STATUS CODE
35EF A7                 ANA     A       ;TEST FUNCTION CODE
35F0 E9                 PCHL            ;RETURN TO CALLER!

35F1 78         BS02:   MOV     A,B     ;GET STATUS CODE
35F2 A7                 ANA     A       ;ANY ERRORS ?
35F3 C2FC35             JNZ     HANER   ;YES!
35F6 D1                 POP     D       ;DISCARD DEVICE NAME
35F7 D1                 POP     D       ;DISCARD PBLK PNTR
35F8 2AE780             LHLD    TBLK    ;GET BLOCK NUMBER
35FB C9                 RET             ;EXIT!
35FC 213136     HANER:  LXI     H,ERS1
35FF CDF433             CALL    OSTR    ;PRINT ERROR STRING
3602 213436             LXI     H,ERS2-3
3605 48                 MOV     C,B     ;COPY ERROR CODE
3606 0600               MVI     B,0
3608 09                 DAD     B       ;POINT TO ERROR STRING
3609 0603               MVI     B,3     ;SET COUNT
360B CDC034             CALL    PSTR    ;PRINT 3 ERROR CHARACTERS
360E 210000             LXI     H,0
3611 39                 DAD     SP      ;POINT TO DEV. NAME ON STACK
3612 0602               MVI     B,2     ;SET COUNT
3614 CDBD34             CALL    PSSTR   ;PRINT DEVICE NAME
3617 E1                 POP     H       ;DISCARD DEVICE NAME
3618 E1                 POP     H       ;GET PBLK PNTR
3619 46                 MOV     B,M     ;GET FUNCTION CODE
361A 23                 INX     H
361B 7E                 MOV     A,M     ;GET UNIT NUMBER
361C CD9B33             CALL    LBYT    ;PRINT UNIT NUMBER
361F CDB834             CALL    PCOLN   ;PRINT A COLON
3622 78                 MOV     A,B     ;GET FUNCTION CODE
3623 CD9B33             CALL    LBYT    ;PRINT FUNCTION CODE
3626 21E780             LXI     H,TBLK  ;POINT TO BLOCK NUMBER
3629 CDD534             CALL    PSNUM   ;PRINT BLOCK NUMBER
362C CD8B33             CALL    CRLF    ;PRINT CR & LF
362F 37                 STC             ;INDICATE ERROR
3630 C9                 RET             ;EXIT!

3631 FF06010B   ERS1:   DB 255,6,1,11,'E',239
3635 45EF
     (0000)     SOK     EQU     0       ; "GOOD" STATUS CODE
3637 495646     ERS2:   DB 'IVF'
     (0003)     EIVF    EQU     $-ERS2
363A 495655             DB 'IVU'
     (0006)     EIVU    EQU     $-ERS2
363D 424C4B             DB 'BLK'
     (0009)     EBLK    EQU     $-ERS2
3640 534B46             DB 'SKF'
     (000C)     ESKF    EQU     $-ERS2
3643 434642             DB 'CFB'
```

```
     (000F)   ECFB     EQU      $-ERS2
3646 445359           DB 'DSY'
     (0012)   EDSY     EQU      $-ERS2
3649 444353           DB 'DCS'
     (0015)   EDCS     EQU      $-ERS2
364C 4D454D           DB 'MEM'
     (0018)   EMEM     EQU      $-ERS2
364F 564659           DB 'VFY'
     (001B)   EVFY     EQU      $-ERS2


3652 3AE580   BSB08:   LDA      TFCN     ;GET FUNCTION CODE
3655 3D                DCR      A        ;TEST IT
3656 A7                ANA      A        ;CLEAR C STATUS BIT
3657 F8                RM                ;READ: RETURN!
3658 CA5F36            JZ       BS04     ;WRITE: PROCESS!
365B 32E580            STA      TFCN     ;WAS VERIFY: SET WRITE
365E C9                RET               ;RETURN
365F 2AEB80   BS04:    LHLD     TBC      ;GET CURRENT BYTE COUNT
3662 EB                XCHG
3663 2AE280            LHLD     OBC      ;GET OLD BYTE COUNT
3666 22EB80            SHLD     TBC      ;RESET BYTE COUNT TO OLD VALUE
3669 CD5934            CALL     SUBHD    ;COMPUTE DIFFERENCE
366C EB                XCHG
366D 2AE980            LHLD     TMEM     ;GET CURRENT MEMORY PNTR
3670 CD5934            CALL     SUBHD    ;ADJUST IT
3673 22E980            SHLD     TMEM     ;RESET MEM PNTR TO OLD VALUE
3676 EB                XCHG
3677 CDA835            CALL     BC2BK    ;CONVERT TO NUMBER OF BLOCKS
367A EB                XCHG
367B 2AE780            LHLD     TBLK     ;GET CURRENT BLOCK NUMBER
367E CD5934            CALL     SUBHD    ;ADJUST IT
3681 22E780            SHLD     TBLK     ;RESET BLK NUMBER TO OLD VALUE
3684 3E02              MVI      A,2      ;SET VERIFY FUNCTION CODE
3686 32E580            STA      TFCN     ;STORE IT
3689 37                STC               ;INDICATE "RESET"
368A C9                RET               ;RETURN

     (368B)            TBADDR   EQU      $        ;SETUP FOR CONTINUATION


                       ;32 LINES BY 64 CHARACTERS
                       [redacted]; COMPUCOLOR II SOFTWARE VER 6.78

                       ;COPYRIGHT (C) 1974,1975,1976,1977,1978
                       ;BY COMPUCOLOR CORP.
                       ;PROGRAMMED BY CHARLES A MUENCH

                       ;8080 CPU CLOCK=17.9712/9=1.9968 MHZ
                       ;MIN TIME INTERVAL=16X.5008=8.0128 US
                       ;MIN COUNT FOR INTERVAL TIMERS=64.1 US
                       ;15.6 COUNTS=1 MS
                       ;78 COUNTS=5 MS
                       ;156 COUNTS=10 MS
                       ;195 COUNTS=12.5 MS
                       ;255 COUNTS =16.346 MS
```

```
                        ;195 COUNTS X 8 =100 MS.
                        ;            X 40 =.5 SEC.
                        ;            X 80 = 1 SEC.
                        ;            X 160 = 2 SEC.
                        ;            X 240 = 3 SEC.

(0020)  NOLIN   EQU     32
(0040)  NOCHA   EQU     64
(0000)  MFIOA   EQU     0
(0000)  RXBUF   EQU     MFIOA
(0001)  EXTIN   EQU     MFIOA+1
(0002)  RSTBF   EQU     MFIOA+2
(0003)  STAT5   EQU     MFIOA+3
(0004)  COMND   EQU     MFIOA+4
(0005)  BAUD    EQU     MFIOA+5
(0006)  TXBUF   EQU     MFIOA+6
(0007)  EXTOT   EQU     MFIOA+7
(0008)  MASK    EQU     MFIOA+8
(0009)  TIME1   EQU     MFIOA+9
(000A)  TIME2   EQU     MFIOA+10
(000B)  TIME3   EQU     MFIOA+11
(000C)  TIME4   EQU     MFIOA+12
(000D)  TIME5   EQU     MFIOA+13
(0080)  FPROM   EQU     80H
(0080)  PRM0    EQU     FPROM+0
(0081)  PRM1    EQU     FPROM+1
(0082)  PRM2    EQU     FPROM+2
(0083)  PRM3    EQU     FPROM+3
(0084)  PRM4    EQU     FPROM+4
(0085)  PRM5    EQU     FPROM+5
(0086)  PRM6    EQU     FPROM+6
(0060)  CRTCHIP EQU     60H
(006C)  CRT0    EQU     CRTCHIP+0
(0061)  CRT1    EQU     CRTCHIP+1
(0062)  CRT2    EQU     CRTCHIP+2
(0063)  CRT3    EQU     CRTCHIP+3
(0064)  CRT4    EQU     CRTCHIP+4
(0065)  CRT5    EQU     CRTCHIP+5
(0066)  CRT6    EQU     CRTCHIP+6
(0066)  ROLDA   EQU     CRT6
(006C)  CRXDA   EQU     CRTCHIP+12
(006D)  CRYDA   EQU     CRTCHIP+13
(006A)  STOPIT  EQU     CRTCHIP+10
(006E)  STARTIT EQU     CRTCHIP+14

(0040)  BASICW  EQU     0040H
(0046)  BASICE  EQU     0046H
(0052)  BASICI  EQU     0052H
(0055)  BASEX   EQU     0055H
(0058)  AUTOX   EQU     0058H

3688 (81AF)             ORG     CRTRAM

81AF (0001)     XOUT0:  DS      1       ;CHEAP DISK USER
81B0 (0001)     XOUT1:  DS      1
81B1 (0001)     CTRK0:  DS      1
81B2 (0001)     CTRK1:  DS      1
```

```
81B3 (0001)   AUCNT:   DS    1
81B4 (0001)   BUCNT:   DS    1
81B5 (0001)   CUCNT0:  DS    1
81B6 (0001)   CUCNT1:  DS    1
81B7 (0001)   PUP:     DS    1       ;POWER UP FLAG
81B8 (0001)   HEX:     DS    1       ; BINARY FRACTIONS OF A SECOND
81B9 (0001)            DS    1       ; SECONDS
81BA (0001)            DS    1       ; MINUTES
81BB (0001)            DS    1       ; HOURS (24)
81BC (0003)   DISPCK:  DS    3       ; JMP TO DISPLAY CLOCK
81BF (0003)   ESCCRT:  DS    3       ; CRT ESCAPE 30 JUMP
81C2 (0003)   OUTCRT:  DS    3
81C5 (0003)   INPCRT:  DS    3       ;FREE INPUT TABLE JMP
81C8 (0003)   RST1J:   DS    3       ;TIMER 2 JMP VECTOR

              ;                     RAM IS CLEARED FROM HERE TO END (8200H)

81CB (0002)   VCRAD:   DS    2
81CD (0001)   ROLLN:   DS    1
81CE (0002)   VFILL:   DS    2
81D0 (0002)   BFILL:   DS    2
81D2 (0002)   VHLAD:   DS    2
81D4 (0002)   BHLAD:   DS    2
81D6 (0001)   EXTBF:   DS    1
81D7 (0001)   SEC15:   DS    1
81D8 (0002)   PCRAD:   DS    2
81DA (0001)   PLOFL:   DS    1
81DB (0001)   PSTAT:   DS    1
81DC (0001)   ROLFL:   DS    1
81DD (0001)   DUPLX:   DS    1
81DE (0001)   THRUFL:  DS    1
81DF (0001)   KBDFL:   DS    1
81E0 (0001)   CMASK:   DS    1       ;CURRENT MASK REGISTER SETTING
81E1 (0001)   FCSFL:   DS    1       ;FILE CONTROL SYSTEM OUTPUT FLAG
81E2 (0001)   CRATE:   DS    1       ;CURRENT BAUD RATE SETTING
81E3 (0001)   INPFL:   DS    1
81E4 (0001)   LKC:     DS    1
81E5 (0001)   NKC:     DS    1
81E6 (0001)   COLFL:   DS    1
81E7 (0003)   JUMP:    DS    3
81EA (0001)   XTWO:    DS    1
81EB (0001)   YTWO:    DS    1
81EC (0001)   XDATA:   DS    1
81ED (0001)   YDATA:   DS    1
81EE (0001)   ODDFL:   DS    1
81EF (0001)   XZERO:   DS    1
81F0 (0001)   YZERO:   DS    1
81F1 (0001)   BASFL:   DS    1
81F2 (0001)   TEMP0:   DS    1
81F3 (0001)   TEMP1:   DS    1
81F4 (0001)   TEMP2:   DS    1
81F5 (0001)   TEMP3:   DS    1
81F6 (0001)   TEMP4:   DS    1
81F7 (0001)   TEMP5:   DS    1
81F8 (0001)   OUTFL:   DS    1
81F9 (0002)   LOFL:    DS    2
81FB (0002)   OUTHL:   DS    2
81FD (0001)   MS150:   DS    1
```

```
                81FE (0001)     KBCHA:  DS      1
                81FF (0001)     READY:  DS      1

                8200 (0000)             ORG     0
                0000            TIMX1:
                0000 C36C37     START:  JMP     STARX
                0003 5200       INITAD: DW      BASICI  ;BASIC INITIAL ADDRESS
                0005 0F         TAU:    DB      15      ; TIME BASE FOR 8/15 SECONDS
                0006 00         MODE:   DB      0
                0007 01         PROLL:  DB      1       ;0=PAGE   1=ROLL

                0008 (0008)             ORG     8
                0008 C3C881     TIMX2:  JMP     RST1J
                000B D836       AESCTB: DW      ESCTB   ;ADDRESS OF ESCAPE TABLE
                000D 0D         CARET:  DB      0DH
                000E B0         CHDEL:  DB      -80
                000F 08         DELTA:  DB      8       ; NUMERATOR FOR 8/15 SECONDS

                0010 (0010)             ORG     10H
                0010 C30538     XINTR:  JMP     UPDATE  ; CLOCK UPDATE
                0013 (0003)             DS      3
                0016 C0         RATEA:  DB      0C0H    ;9600BAUD+1 STOP
                0017 08         RCOMD:  DB      08H     ;BAUD  1X=08H    8X=18H

                0018 (0018)             ORG     18H
                0018 C3A63E     TIMX3:  JMP     TIM3X
                001B F4         TPROG:  DB      -12     ;150MS
                001C 05         CHTIM:  DB      5
                001D FF         KEDEL:  DB      255
                001E FF81       KERDY:  DW      READY

                0020 (0020)             ORG     20H
                0020 C37439     RXSER:  JMP     ISERL
                0023 40                 DB      64      ;NUMBER OF CHARACTERS
                0024 C3D533     KEYTEST: JMP    RTST
                0027 14         STEPCD: DB      STPTIM  ;CHEAP DISK STEP RATE

                0028 (0028)             ORG     28H
                0028 C3613A     TXSER:  JMP     OSERL
                002B C34F39     KEYCO:  JMP     KBREP   ;PROCESS KEY CODE
                002E 04         CDMK:   DB      4       ; 4 FOR PROPER CLOCK OPERATION
                002F A0         RATEB:  DB      0A0H    ;4800 BAUD +1 STOP BIT

                0030 (0030)             ORG     30H
                0030 C3D13A     TIMX4:  JMP     TIM4X
                0033 C36B39     BASOUT: JMP     CRTUBE  ;PROCESS CHAR WITH BASIC FLAG
                0036 4280       ACRTSP: DW      STACK   ;ADDRESS OF TOP OF STACK FOR CRT MODE


                0038 (0038)             ORG     38H
                0038            TIMX5:
                0038 C36837     BEGEX:  JMP     BEGIN
                003B C3B63A     BREAK:  JMP     BRAKE   ;PROCESS BREAK CODE
                003E FE81       KCHAR:  DW      KBCHA

                0040 (368B)             ORG     TBADDR
```

MEM LOCATIONS 0001–0003 → re #3767(H) deletions 6-78

START OF UA4 ↓

THE REST MISSING – BASIC LISTING!

```
                        ;       DISK HANDLER VALUES AND VECTORS

x | 368B 4344       IDEV:   DB      'CD'    ;INITIAL DEFAULT DEVICE
  | 368D 30         IUNT:   DB      '0'     ;INITIAL DEFAULT UNIT

  | 368E C31C21     HDVCT:  JMP     CDHD    ;JMP TO HANDLER
  | 3691 4344       CDNM:   DB      'CD'    ;DEVICE NAME
x | 3693 02         CDNU:   DB      2       ;NUMBER OF UNITS
  | 3694 0A         CDSEC:  DB      NSEC
  | 3695 50         CDK:    DB      UPTIM/STPTIM
  | 3696 (0002)             DS      2

    3698 FF                 DB      255
    3699 (0009)             DS      9

    36A2 FF                 DB      255
    36A3 (0005)             DS      5


    36A8 0A3A5800 TBLOO:    DW      FREE,AUTOX      ;@,A 0,1
    36AC 993D093A           DW      POTON,CRSXY     ;B,C 2,3
    36B0 0A3A0A3A           DW      FREE,FREE       ;D,E/4,5
    36B4 093AC33A           DW      CCI,BEL         ;F,G/6,7
    36B8 6E38B138           DW      HOME,TAB        ;H,I/8,9
    36BC BD38EC3A           DW      LF,ELINE        ;J,K/10,11
    36C0 36387238           DW      ERASE,CR        ;L,M/12,13
    36C4 E8384639           DW      A7ON,BA7OF      ;N,O/14,15
    36C8 963AB538 TBL24:    DW      PRINT,CRSRT     ;X,Y/24,25
    36CC F0389093A          DW      CRSLT,ESCAP     ;Z,ESC/26,27
    36D0 F6383B39           DW      CRSUP,COMOF     ;\,]/28,29
    36D4 3A393F39           DW      COMON,BLINK     ;^,_/

14040(D) 36D8 0A3A093A ESCTB:   DW      FREE,BLIND      ;@,A / 0,1
    36DC FD395D3B           DW      CHPLO,TXPEN     ;B,C / 2,3
    36E0 C9324600           DW      ESCD,BASICE     ;D,E / 4,5
    36E4 4E3ABE32           DW      FULL,ESCG       ;F,G / 6,7
    36E8 4C3A0090           DW      HALF,9000H      ;H,I / 8,9
    36EC 903A853A           DW      DOWN,ROLL       ;J,K / 10,11
    36F0 4F3AFE39           DW      LOCAL,TMODE     ;L,M / 12,13
    36F4 FE390A3A           DW      TMODE,FREE      ;N,O / 14,15
    36F8 0A3A0A3A 0040      DW 4000(H) FREE,FREE    ;P,Q / 16,17
    36FC 093A00A0           DW      BRATE,0A000H    ;R,S / 18,19
    3700 00820A3A           DW      8200H,FREE      ;T,U / 20,21
    3704 0A3A4000           DW      FREE,BASICW     ;V,W / 22,23
    3708 863A093A           DW      PAGE,TEST       ;X,Y / 24,25
    370C 0A3A0A3A 0048      DW 4800(H) FREE,VISIB   ;Z,[ / 26,27
    3710 0A3A0A3A 0050 0058 DW 5000(H) FREE,FREE 5700(H) ;\,] / 28,29
14100(D) 3714 BF81B725      DW      ESCCRT,CRTMO    ;^,_ / 30,31

    3718 593A533A OUTBL:    DW      BEGOT,KEYOT     ;0,1
    371C 0A3B133B           DW      LTPEN,XYMIT     ;2,3
    3720 453B4E3B           DW      LINE,CARR       ;4,5
    3724 C281563B           DW      OUTCRT,CRET     ;6,7


    3728 DC39BC39 INPTB:    DW      SVCHA,SBCHA     ;0,1
    372C B23C0B3A           DW      PLOTX,CURSO     ;2,3
```

DISABLED IN V.2.79

CORRECTIONS FOR NEWMAN, FOUND - FITTED - 1984

```
3730 033E1F3A          DW      CPLOX,VCRSY     ;4,5
3734 013A2A3A          DW      CCIX,BCRSX      ;6,7
3738 373A053A          DW      BCRSY,BCCIX     ;8,9
373C C581C581          DW      INPCRT,INPCRT   ;10,11
3740 0A3A0133          DW      FREEX,FCSX      ;12,13
3744 C333C581          DW      S1OUT,INPCRT    ;14,15
3748 C581C581          DW      INPCRT,INPCRT   ;16,17
374C 6F3AC581          DW      BRATX,INPCRT    ;18,19
3750 C581C581          DW      INPCRT,INPCRT   ;20,21
3754 C5815500          DW      INPCRT,BASEX    ;22,23
3758 C5813138          DW      INPCRT,TESTX    ;24,25
375C C581AA3A          DW      INPCRT,ESCAX    ;26,27
3760 C581C581          DW      INPCRT,INPCRT   ;28,29
3764 C581C581          DW      INPCRT,INPCRT   ;30,31

                       ;BEGEX LOOP TIME ,NO WAIT = 33 US
                       ;WITH PROM WAIT = 40.5 US

3768 FB        BEGIN:  EI
3769 C33800            JMP     BEGEX

376C 113800    STARX:  LXI     D,BEGEX ; SET RETURN ON STACK
376F 2A3600            LHLD    ACRTSP  ; SET UP STACK ADRESS
3772 F9                SPHL
3773 D5                PUSH    D
3774 D36A              OUT     STOPIT  ; RESET CRT CHIP
3776 DB80              IN      PRM0    ; COPY PROM PARAMETERS
3778 D360              OUT     CRT0    ; INTO CRT REGISTERS
377A DB81              IN      PRM1
377C D361              OUT     CRT1
377E DB82              IN      PRM2
3780 D362              OUT     CRT2
3782 DB83              IN      PRM3
3784 D363              OUT     CRT3
3786 DB84              IN      PRM4
3788 D364              OUT     CRT4
378A DB85              IN      PRM5
378C D365              OUT     CRT5
378E DB86              IN      PRM6
3790 D366              OUT     CRT6
3792 D36E              OUT     STARTIT ; RESTART CRT CHIP
3794 3E03              MVI     A,3
3796 D304              OUT     COMND   ;RESET MFIOA
3798 3AB781            LDA     PUP     ; IS THIS REAL POWERUP ?
379B FE97              CPI     97H
379D C2B137            JNZ     PWRUP   ; YES !
37A0 3E80              MVI     A,080H  ; IS THIS USER RESET (COMMAND) ?
37A2 D307              OUT     EXTOT
37A4 DB01              IN      EXTIN
37A6 E630              ANI     30H
37A8 2A1637            LHLD    OUTBL-2 ; SETUP CRTMODE AS EXIT POINT
37AB C2C037            JNZ     CRTSET  ; NO RESET TO CRT MODE
37AE C3BD37            JMP     INIBAS  ; YES, INITIALIZE BASIC ONLY
37B1 21B881    PWRUP:  LXI     H,HEX
37B4 0604              MVI     B,4
37B6 3600              MVI     M,0
37B8 23                INX     H
```

```
37B9 05                 DCR     B
37BA C2B637             JNZ     $-4
37BD 2A0300     INIBAS: LHLD    INITAD
37C0 E5         CRTSET: PUSH    H       ; SAVE EXIT POINT
37C1 21CB81             LXI     H,VCRAD
37C4 AF                 XRA     A
37C5 77                 MOV     M,A
37C6 2C                 INR     L
37C7 C2C537             JNZ     $-2
37CA 2EE7               MVI     L,JUMP AND 255
37CC 36C3               MVI     M,0C3H
37CE 2ECF               MVI     L,VFILL+1 AND 255
37D0 3607               MVI     M,7
37D2 2ED1               MVI     L,BFILL+1 AND 255
37D4 34                 INR     M
37D5 3A0700             LDA     PROLL
37D8 CD863A             CALL    PAGE
37DB CD3638             CALL    ERASE
37DE CDE03A             CALL    TIM4Y
37E1 3A1600             LDA     RATEA
37E4 D305               OUT     BAUD
37E6 32E281             STA     CRATE
37E9 D30B               OUT     TIME3
37EB 3A0600             LDA     MODE
37EE 32DD81             STA     DUPLX
37F1 CD593A             CALL    BEGOT
37F4 CDA526             CALL    RESET
37F7 C9                 RET             ; JUMP TO ENTRY POINT

37F8 2ACE81             LHLD    VFILL
37FB EB                 XCHG
37FC 2ACB81     GETBC:  LHLD    VCRAD
37FF 4D                 MOV     C,L
3800 44                 MOV     B,H
3801 2AD281             LHLD    VHLAD
3804 C9                 RET

3805 CDD03F     UPDATE: CALL    SAVE    ; INTERUPT ROUTINE
3808 21B881             LXI     H,HEX   ; UPDATE FRACTION
380B 3A0F00             LDA     DELTA
380E 47                 MOV     B,A
380F 3A0500             LDA     TAU
3812 4F                 MOV     C,A
3813 7E                 MOV     A,M     ; SUBTRACT DELTA
3814 90                 SUB     B
3815 77                 MOV     M,A     ; STORE NEW FRACTION
3816 D0                 RNC             ; RETURN IF NO CARRY
3817 81                 ADD     C       ; ADD TAU
3818 77                 MOV     M,A     ; REPLACE NEW FRACTION
3819 23                 INX     H       ; POINT AT SECONDS
381A 34                 INR     M       ; INCREMENT SECONDS
381B 7E                 MOV     A,M     ; AND LOAD THEM
381C D63C               SUI     60      ; COMPARE TO 60
381E D8                 RC              ; SECONDS ARE GOOD : EXIT
381F 77                 MOV     M,A     ; MAPS 60 TO 0
3820 23                 INX     H       ; POINT AT MINUTES
3821 34                 INR     M       ; INCREMENT MINUTES
3822 7E                 MOV     A,M     ; AND LOAD THEM
```

MEM LOCATION	15D		171D	
380C	0F	change to	AB	← 217D
3810	05	" "	D9	
3811	00	" "	01	

WINDER EPROM MOD TO CHANGE 60HZ TO 50HZ.

```
3813 D63C              SUI     60      ; COMPARE TO 60
3815 D8                RC             ; MINUTES ARE GOOD : EXIT
3816 77                MOV     M,A    ; MAPS 60 TO 0
3817 23                INX     H      ; POINT AT HOURS
3818 34                INR     M      ; INCREMENT HOURS
3819 7E                MOV     A,M    ; AND LOAD THEM
381A FE19              CPI     25     ; COMPARE TO 25
381C D8                RC             ; HOURS ARE GOOD : EXIT
381D D618              SUI     24
381F 77                MOV     M,A    ; MAPS 25 TO 1
3820 C9                RET            ; FIX HOURS AND EXIT ANYWAY

3821 77        TESTX:  MOV     M,A
3822 7B                MOV     A,E
3823 C33B38            JMP     EARS+2
3826 CD6438    ERASE:  CALL    NOROL
3829 3E20      EARS:   MVI     A,20H  ;SPACE
382B 210070            LXI     H,X80
382E 0600              MVI     B,0    ;NOLIN*NOCHA/8
3830 E5                PUSH    H
3831 2ACE81            LHLD    VFILL
3834 EB                XCHG
3835 B3                ORA     E
3836 5F                MOV     E,A
3837 210000            LXI     H,0
383A 39                DAD     SP
383B 22F281            SHLD    TEMPO
383E E1                POP     H
383F F3                DI
3840 F9                SPHL
3841 D5        ERALP:  PUSH    D
3842 D5                PUSH    D
3843 D5                PUSH    D
3844 D5                PUSH    D
3845 D5                PUSH    D
3846 D5                PUSH    D
3847 D5                PUSH    D
3848 D5                PUSH    D
3849 05                DCR     B
384A C25138            JNZ     ERALP
384D 2AF281            LHLD    TEMPO
3850 F9                SPHL
3851 FB                EI
3852 E1                POP     H
3853 C9                RET
3854 210070    NOROL:  LXI     H,X80
3857 22D481            SHLD    BHLAD
385A AF                XRA     A
385B 32CD81            STA     ROLLN
385E 21CD81    HOME:   LXI     H,ROLLN
3861 46                MOV     B,M
3862 4F        CR:     MOV     C,A
3863 CD9738    SVCRS:  CALL    LN5X
3866 22D281            SHLD    VHLAD
3869 21CB81            LXI     H,VCRAD
386C 71                MOV     M,C
386D 23                INX     H
386E 70                MOV     M,B
```

```
387F 23                 INX     H
3880 7E                 MOV     A,M
3881 3D                 DCR     A
3882 E61F               ANI     31
3884 D366               OUT     ROLDA
3886 78                 MOV     A,B
3887 D36D               OUT     CRYDA
3889 79                 MOV     A,C
388A D36C               OUT     CRXDA
388C C9                 RET
388D CD433A     CRSY:   CALL    TESTB-2
3890 3ACD81             LDA     ROLLN
3893 80                 ADD     B
3894 CD453A             CALL    TESTB
3897 3ACE81     LN5X:   LDA     VFILL
389A 07                 RLC
389B B0                 ORA     B
389C 47                 MOV     B,A
389D 78                 MOV     A,B
389E 07         LN5Y:   RLC
389F 07                 RLC
                ;       ADD     B       ;A=5B
38A0 218003             LXI     H,X80/(32)
38A3 85                 ADD     L
38A4 6F                 MOV     L,A
38A5 29                 DAD     H
38A6 29                 DAD     H
38A7 29                 DAD     H
38A8 29                 DAD     H
38A9 29                 DAD     H
38AA 79                 MOV     A,C
38AB 07                 RLC
38AC 85                 ADD     L
38AD 6F                 MOV     L,A
38AE D0                 RNC
38AF 24                 INR     H
38B0 C9                 RET
38B1 3E07       TAB:    MVI     A,7
38B3 B1                 ORA     C
38B4 4F                 MOV     C,A
38B5 0C         CRSRT:  INR     C
38B6 3EC0               MVI     A,-NOCHA
38B8 81                 ADD     C
38B9 D27338             JNC     SVCRS
38BC 4F                 MOV     C,A
38BD 04         LF:     INR     B
38BE CD463A             CALL    TESTB+1
38C1 3ADC81     IROLL:  LDA     ROLFL
38C4 A7                 ANA     A
38C5 CA7338             JZ      SVCRS
38C8 21CD81             LXI     H,ROLLN
38CB 78                 MOV     A,B
38CC AE                 XRA     M
38CD C27338             JNZ     SVCRS
38D0 34                 INR     M
38D1 34                 INR     M
38D2 3EE0               MVI     A,-NOLIN
38D4 CD333A             CALL    TESTC+6
```

```
38D7 23              INX     H
38D8 4E              MOV     C,M
38D9 C5              PUSH    B
38DA AF              XRA     A
38DB 77              MOV     M,A
38DC CDF83A          CALL    EARLN
38DF 018000          LXI     B,NOCHA*2
38E2 CD013B          CALL    EARLN+9
38E5 F1              POP     PSW
38E6 47              MOV     B,A
38E7 F0              RP
38E8 21CE81  A7ON:   LXI     H,VFILL
38EB 3680            MVI     M,80H
38ED C37338          JMP     SVCRS
38F0 0D      CRSLT:  DCR     C
38F1 F27338          JP      SVCRS
38F4 0E3F            MVI     C,NOCHA-1
38F6 3ACE81  CRSUP:  LDA     VFILL
38F9 A7              ANA     A
38FA F2FE38          JP      $+4
38FD 05              DCR     B
38FE 05              DCR     B
38FF F27338          JP      SVCRS
3902 061F            MVI     B,NOLIN-1
3904 C37338          JMP     SVCRS
3907 E3      COLOR:  XTHL
3908 CDFC37          CALL    GETBC
390B 5F              MOV     E,A
390C 21A836          LXI     H,TBL00
390F FE10            CPI     16
3911 DA9639          JC      CODE
3914 FE18            CPI     24
3916 219836          LXI     H,TBL24-48
3919 D29639          JNC     CODE
391C E1              POP     H
391D 2EE6            MVI     L,COLFL AND 255
391F CD2C39          CALL    COLW
3922 CA2839          JZ      BKCOL
3925 3EF8            MVI     A,0F8H
3927 50              MOV     D,B
3928 A6      BKCOL:  ANA     M
3929 B2              ORA     D
392A 77              MOV     M,A
392B C9              RET
392C E607    COLW:   ANI     7
392E 47              MOV     B,A
392F 07              RLC
3930 07              RLC
3931 07              RLC
3932 57              MOV     D,A
3933 6E              MOV     L,M
3934 2D              DCR     L
3935 2ECF            MVI     L,VFILL+1 AND 255
3937 3EC7            MVI     A,0C7H
3939 C9              RET
393A 3C      COMON:  INR     A
393B 2EE6    COMOF:  MVI     L,COLFL AND 0FFH
393D 77              MOV     M,A
```

```
393E C9              RET
393F 3E40    BLINK:  MVI     A,40H
3941 2ECF            MVI     L,VFILL+1 AND 255
3943 B6              ORA     M
3944 77              MOV     M,A
3945 C9              RET
3946 2ECE    BA7OF:  MVI     L,VFILL AND 255
3948 77              MOV     M,A
3949 3EBF            MVI     A,0BFH
394B 2C              INR     L
394C A6              ANA     M
394D 77              MOV     M,A
394E C9              RET

394F 5F      KBREP:  MOV     E,A     ;PUT CHAR IN E
3950 3ADD81          LDA     DUPLX
3953 A7              ANA     A       ;LOCAL=0
3954 3EDF            MVI     A,0DFH  ;T5,T4,-,RX,T3,RTC,T2,T1
3956 21F881          LXI     H,OUTFL
3959 CA6039          JZ      $+7     ;HALF=1
395C 3601            MVI     M,1
395E 3EF7            MVI     A,0F7H  ;T5,T4,TX,RX,-,RTC,T2,T1
3960 D308            OUT     MASK
3962 32E081          STA     CMASK   ;SAVE CURRENT MASK SETTING
3965 2EDF            MVI     L,KBDFL AND 255
3967 F8              RM              ;FULL=-1
3968 C39139          JMP     ISERX

396B CDD03F  CRTUBE: CALL    SAVE    ;SAVE ALL REGS.
396E 21F181          LXI     H,BASFL
3971 C38C39          JMP     PROCES

                     ;ISERL TO INPTB TIME = 103 US, NO WAIT
                     ;WITH PROM WAIT = 122.5 US

3974 CDD03F  ISERL:  CALL    SAVE    ;SAVE ALL REGISTERS
3977 21FF81          LXI     H,READY
397A 3650            MVI     M,50H
397C DB03            IN      STAT5
397E E601            ANI     1
3980 C0              RNZ
3981 3680            MVI     M,80H
3983 2B              DCX     H
3984 DB00            IN      RXBUF
3986 77              MOV     M,A
3987 DB00            IN      RXBUF
3989 21E381          LXI     H,INPFL
398C 5F      PROCES: MOV     E,A
398D 3E01            MVI     A,1
398F D307            OUT     EXTOT
3991 7E      ISERX:  MOV     A,M
3992 E5              PUSH    H
3993 212837          LXI     H,INPTB
3996 07      CODE:   RLC
3997 85              ADD     L
3998 6F              MOV     L,A
3999 7C              MOV     A,H
399A CE00            ACI     0
```

```
399C 67                  MOV     H,A
399D 7E                  MOV     A,M
399E 23                  INX     H
399F 66                  MOV     H,M
39A0 6F                  MOV     L,A
39A1 E3                  XTHL
39A2 3AD681    CODE2:    LDA     EXTBF
39A5 D307                OUT     EXTOT
39A7 AF                  XRA     A
39A8 C9                  RET

39A9 7B        ZERFL:    MOV     A,E
39AA E67F                ANI     7FH
39AC FE20                CPI     32
39AE DA0739              JC      COLOR
39B1 FE60                CPI     96
39B3 D8                  RC
39B4 2EE6      SP64C:    MVI     L,COLFL AND 255
39B6 6E                  MOV     L,M
39B7 2D                  DCR     L
39B8 C0                  RNZ
39B9 D660                SUI     96
39BB C9                  RET

                         ;SBCHA NO 2X CHA = 84 US ,NO WAIT
                         ;SBCHA WITH SP64 CHA ADD 18 US ,NO WAIT

39BC CDA939    SBCHA:    CALL    ZERFL    ;34 US MIN
39BF 2AD081              LHLD    BFILL    ;41 US MIN WAIT
39C2 EB                  XCHG             ;52 US MAX
39C3 2AD481              LHLD    BHLAD    ;63 US MAX WAIT
39C6 B3                  ORA     E
39C7 77                  MOV     M,A
39C8 23                  INX     H
39C9 72                  MOV     M,D
39CA 23                  INX     H
39CB FCF139              CM      STOR2
39CE 3E80      TESHI:    MVI     A,(X80+NOLIN*NOCHA*2) SHR 8
39D0 BC                  CMP     H
39D1 22D481              SHLD    BHLAD
39D4 C0                  RNZ
39D5 210070              LXI     H,X80
39D8 22D481              SHLD    BHLAD
39DB C9                  RET

39DC CDA939    SVCHA:    CALL    ZERFL
39DF CDE839              CALL    STOR1-3
39E2 FCF139              CM      STOR2
39E5 C3E781              JMP     JUMP
39E8 CDF837              CALL    GETEC-4
39EB B3        STOR1:    ORA     E
39EC 77                  MOV     M,A
39ED 23                  INX     H
39EE 72                  MOV     M,D
39EF 23                  INX     H
39F0 C9                  RET
39F1 E5        STOR2:    PUSH    H
39F2 C5                  PUSH    B
```

```
39F3 017EFF             LXI     B,-NOCHA*2-2
39F6 09                 DAD     B
39F7 C1                 POP     B
39F8 77                 MOV     M,A
39F9 23                 INX     H
39FA 72                 MOV     M,D
39FB E1                 POP     H
39FC C9                 RET

39FD 1C         CHPLO:  INR     E
39FE 1C         TMODE:  INR     E
39FF 73                 MOV     M,E
3A00 C9                 RET

                        ;CCIX TRICKEY BYTE SAVER

3A01 77         CCIX:   MOV     M,A
3A02 2ECF               MVI     L,VFILL+1 AND 255
3A04 0636               MVI     B,36H   ;TRICK
3A06 012ECF             LXI     B,0CF2EH        ;TRICK
3A09 (3A05)             ORG     $-4

3A05 3601       BCCIX:  MVI     M,1
3A07 2ED1               MVI     L,BFILL+1 AND 255

3A09            ESCAP:
3A09            BRATE:
3A09            BLIND:
3A09            CRSXY:
3A09            CCI:
3A09            TEST:
3A09            CHAIN:
3A09 73                 MOV     M,E
3A0A            FREE:
3A0A            FREEX:
3A0A C9         VISIB:  RET

3A0B 7B         CURSO:  MOV     A,E
3A0C FE51               CPI     81
3A0E 3607               MVI     M,7
3A10 D2193A             JNC     B7ON
3A13 3605               MVI     M,5
3A15 2ECB               MVI     L,VCRAD AND 255
3A17 77                 MOV     M,A
3A18 C9                 RET
3A19 E680       B7ON:   ANI     80H
3A1B 2ED0               MVI     L,BFILL AND 255
3A1D 77                 MOV     M,A
3A1E C9                 RET

3A1F 77         VCRSY:  MOV     M,A
3A20 2ECB               MVI     L,VCRAD AND 255
3A22 4E                 MOV     C,M
3A23 7B                 MOV     A,E
3A24 CD8D38             CALL    CRSY
3A27 C37638             JMP     SVCRS+3
3A2A 34         BCRSX:  INR     M
3A2B 2ED4               MVI     L,BHLAD AND 255
```

```
3A2D 7B        TESTC:  MOV     A,E
3A2E E67F              ANI     127
3A30 77                MOV     M,A
3A31 3EC0              MVI     A,-NOCHA
3A33 86                ADD     M
3A34 D0                RNC
3A35 77                MOV     M,A
3A36 C9                RET
3A37 34        BCRSY:  INR     M
3A38 2ED4              MVI     L,BHLAD AND 255
3A3A 4E                MOV     C,M
3A3B 7B                MOV     A,E
3A3C CD8D38            CALL    CRSY
3A3F 22D481            SHLD    BHLAD
3A42 C9                RET

3A43 E63F              ANI     31 OR NOLIN
3A45 47        TESTB:  MOV     B,A
3A46 3EE0              MVI     A,-NOLIN
3A48 80                ADD     B
3A49 D0                RNC
3A4A 47                MOV     B,A
3A4B C9                RET

3A4C 3E02      HALF:   MVI     A,2
3A4E 3D        FULL:   DCR     A
3A4F 32DD81    LOCAL:  STA     DUPLX
3A52 C9                RET

3A53 77        KEYOT:  MOV     M,A
3A54 3AFE81            LDA     KBCHA
3A57 D306              OUT     TXBUF
3A59 3EDF      BEGOT:  MVI     A,0DFH  ;T5,T4,-,RX,T3,RTC,T2,T1
3A5B D308              OUT     MASK
3A5D 32E081            STA     CMASK   ;SAVE   CURRENT MASK SETTING
3A60 C9                RET
3A61 CDD03F    OSERL:  CALL    SAVE    ;SAVE ALL REGISTERS
3A64 21F881            LXI     H,OUTFL
3A67 E5                PUSH    H
3A68 7E                MOV     A,M
3A69 211837            LXI     H,OUTBL
3A6C C39639            JMP     CODE
3A6F 77        BRATX:  MOV     M,A
3A70 3E07              MVI     A,7
3A72 A3                ANA     E
3A73 C8                RZ
3A74 5F                MOV     E,A
3A75 3E80              MVI     A,80H
3A77 07                RLC
3A78 1D                DCR     E
3A79 C2773A            JNZ     $-2
3A7C 2ECE              MVI     L,VFILL AND 255
3A7E B6                ORA     M
3A7F D305              OUT     BAUD
3A81 32E281            STA     CRATE   ;SAVE CURRENT BAUD RATE
3A84 C9                RET

3A85 3C        ROLL:   INR     A
```

```
3A86 21B538    PAGE:   LXI     H,CRSRT
3A89 32DC81            STA     ROLFL
3A8C 22E881            SHLD    JUMP+1
3A8F C9                RET
3A90 21BD38    DOWN:   LXI     H,LF
3A93 C3893A            JMP     PAGE+3

3A96 2EF8      PRINT:  MVI     L,OUTFL AND 255
3A98 3605              MVI     M,5
3A9A 21FF7F            LXI     H,X80+NOLIN*NOCHA*2-1
3A9D 77                MOV     M,A
3A9E 2F                CMA
3A9F 2B                DCX     H
3AA0 77                MOV     M,A
3AA1 CDA33D            CALL    NROLL
3AA4 2AD281            LHLD    VHLAD
3AA7 C3323B            JMP     TXOUT+2

3AAA 77        ESCAX:  MOV     M,A
3AAB 3E1F              MVI     A,31
3AAD E5                PUSH    H
3AAE 21D836            LXI     H,ESCTB
3AB1 A3                ANA     E
3AB2 5F                MOV     E,A
3AB3 C39639            JMP     CODE
3AB6 CD4C3A    BRAKE:  CALL    HALF
3AB9 3A1700            LDA     RCOMD
3ABC F602              ORI     2
3ABE D304              OUT     COMND
3AC0 C3C83A            JMP     BEL+5
3AC3 3E40      BEL:    MVI     A,40H
3AC5 CDE63A            CALL    TIM4Z
3AC8 3A1B00            LDA     TPROG
3ACB 32FD81            STA     MS150
3ACE C3D83A            JMP     TIM4X+7

3AD1 CDD03F    TIM4X:  CALL    SAVE    ;SAVE ALL REGISTERS
3AD4 21FD81            LXI     H,MS150
3AD7 34                INR     M
3AD8 CAE03A            JZ      TIM4Y
3ADB 3EC3              MVI     A,195
3ADD D30C              OUT     TIME4
3ADF C9                RET
3AE0 3A1700    TIM4Y:  LDA     RCOMD
3AE3 D304              OUT     COMND
3AE5 AF                XRA     A
3AE6 32D681    TIM4Z:  STA     EXTBF
3AE9 D307              OUT     EXTOT
3AEB C9                RET

3AEC CDF83A    ELINE:  CALL    EARLN
3AEF 1C                INR     E
3AF0 1D                DCR     E
3AF1 0180FF            LXI     B,-NOCHA*2
3AF4 FA013B            JM      EARLN+9
3AF7 C9                RET

3AF8 CD7238    EARLN:  CALL    CR
```

```
3AF8 2AD281             LHLD    VHLAD
3AFE 0180F0             LXI     B,NOCHA*2-NOCHA*2*NOLIN
3B01 09                 DAD     B
3B02 010008             LXI     B,NOCHA*32
3B05 3E20               MVI     A,20H
3B07 C34038             JMP     EARS+7

3B0A 2AFB81   LTPEN:    LHLD    OUTHL
3B0D 3EF7               MVI     A,TEMP5 AND 255
3B0F BD                 CMP     L
3B10 C3253B             JMP     TWOUT

3B13 2AFB81   XYMIT:    LHLD    OUTHL
3B16 7D                 MOV     A,L
3B17 E67F               ANI     127
3B19 CA3D3B             JZ      FEED
3B1C AF                 XRA     A
3B1D BE       TXCONT:   CMP     M
3B1E C2253B             JNZ     TWOUT
3B21 2B                 DCX     H
3B22 2F                 CMA
3B23 BE                 CMP     M
3B24 23                 INX     H
3B25 7E       TWOUT:    MOV     A,M
3B26 23                 INX     H
3B27 23                 INX     H
3B28 C2303B             JNZ     TXOUT
3B2B 21F881             LXI     H,OUTFL
3B2E 3607               MVI     M,7
3B30 D306     TXOUT:    OUT     TXBUF
3B32 3EFF               MVI     A,0FFH  ;T5T4,TX,RX,T3,RTC,T2,T1
3B34 D308               OUT     MASK
3B36 32E081             STA     CMASK   ;SAVE CURRENT MASK
3B39 22FB81             SHLD    OUTHL
3B3C C9                 RET

3B3D 21F881   FEED:     LXI     H,OUTFL
3B40 3E0A               MVI     A,10    ;LINEFEED
3B42 C3473B             JMP     LINE+2
3B45 3E0D     LINE:     MVI     A,13    ;CARRAGE RETURN
3B47 34                 INR     M       ;INC FLAG VALUE
3B48 2AFB81             LHLD    OUTHL
3B4B C3303B             JMP     TXOUT

3B4E 35       CARR:     DCR     M
3B4F 35                 DCR     M
3B50 2AFB81             LHLD    OUTHL
3B53 C31D3B             JMP     TXCONT

3B56 77       CRET:     MOV     M,A
3B57 3A0D00             LDA     CARET   ;CR
3B5A C3573A             JMP     KEYOT+4

3B5D 77       TXPEN:    MOV     M,A
3B5E CDFC37             CALL    GETBC
3B61 5E                 MOV     E,M
3B62 23                 INX     H
3B63 56                 MOV     D,M
```

```
3B64 21F881          LXI     H,OUTFL
3B67 3602            MVI     M,2
3B69 2EF7            MVI     L,TEMP5 AND 255
3B6B 73              MOV     M,E
3B6C 2D              DCR     L
3B6D 72              MOV     M,D
3B6E 2D              DCR     L
3B6F 3606            MVI     M,6
3B71 2D              DCR     L
3B72 3ACD81          LDA     ROLLN
3B75 2F              CMA
3B76 3C              INR     A
3B77 80              ADD     B
3B78 F27D3B          JP      $+5
3B7B C620            ADI     NOLIN
3B7D 77              MOV     M,A
3B7E 2D              DCR     L
3B7F 71              MOV     M,C
3B80 3E03            MVI     A,3
3B82 C3303B          JMP     TXOUT

3B85 07        SHIFF:RLC
3B86 D2943B          JNC     LS2
3B89 5F              MOV     E,A
3B8A 2C              INR     L
3B8B 7D              MOV     A,L
3B8C FE80            CPI     NOCHA*2
3B8E DA933B          JC      LS1
3B91 AF              XRA     A
3B92 6F              MOV     L,A
3B93 7B        LS1:  MOV     A,E
3B94 07        LS2:  RLC
3B95 D2A03B          JNC     LS4
3B98 2D              DCR     L
3B99 2C              INR     L
3B9A C29F3B          JNZ     LS3
3B9D 2E80            MVI     L,NOCHA*2
3B9F 2D        LS3:  DCR     L
3BA0 07        LS4:  RLC
3BA1 D2AC3B          JNC     LS6
3BA4 25              DCR     H
3BA5 24              INR     H
3BA6 C2AB3B          JNZ     LS5
3BA9 2680            MVI     H,NOLIN*4
3BAB 25        LS5:  DCR     H
3BAC 07        LS6:  RLC
3BAD 5F              MOV     E,A
3BAE D2BB3B          JNC     LS8
3BB1 24              INR     H
3BB2 7C              MOV     A,H
3BB3 FE80            CPI     NOLIN*4
3BB5 DABA3B          JC      LS7
3BB8 AF              XRA     A
3BB9 67              MOV     H,A
3BBA 7B        LS7:  MOV     A,E
3BBB E6F0      LS8:  ANI     0F0H
3BBD C9              RET
```

```
3BBE CDE33B   VECTY:  CALL   INVY-2
3BC1 C32D3E           JMP    VECTO
3BC4 2AEA81   INVEC:  LHLD   XTWO
3BC7 CD853B           CALL   SHIFF
3BCA 22EC81           SHLD   XDATA
3BCD C8               RZ
3BCE 4D               MOV    C,L
3BCF 44               MOV    B,H
3BD0 2AEF81           LHLD   XZERO
3BD3 7B               MOV    A,E
3BD4 CD853B           CALL   SHIFF
3BD7 22EF81           SHLD   XZERO
3BDA C8               RZ
3BDB E5               PUSH   H
3BDC CDE53C           CALL   PUTYD-6
3BDF C1               POP    B
3BE0 C32F3E           JMP    VECTO+2

3BE3 2EF0             MVI    L,YZERO AND 255
3BE5 47       INVY:   MOV    B,A
3BE6 3E7F             MVI    A,NOLIN*4-1
3BE8 90               SUB    B
3BE9 FE80             CPI    NOLIN*4
3BEB 77               MOV    M,A
3BEC 47               MOV    B,A
3BED D8               RC
3BEE C680             ADI    NOLIN*4
3BF0 C3E93B           JMP    $-7
3BF3 2EEF     PXZER:  MVI    L,XZERO AND 255
3BF5 FE80             CPI    NOCHA*2
3BF7 77               MOV    M,A
3BF8 D8               RC
3BF9 D680             SUI    NOCHA*2
3BFB 77               MOV    M,A
3BFC C9               RET
3BFD CDE53B   PLPTY:  CALL   INVY
3C00 1E67             MVI    E,XYTAB AND 255
3C02 C3EB3C           JMP    PUTYD

3C05 CD2C3D   PBINX:  CALL   INCXY
3C08 CA113C           JZ     $+9
3C0B CD113C           CALL   $+6
3C0E CD483D           CALL   MOVXY
3C11 79               MOV    A,C
3C12 2646             MVI    H,46H    ;TRICK
3C14 (3C13)           ORG    $-1

3C13 46       BARXM:  MOV    B,M
3C14 21EF81           LXI    H,XZERO
3C17 BE               CMP    M
3C18 CAE63C           JZ     PUTYD-5
3C1B 1E5F             MVI    E,BARTX AND 255
3C1D CDE83C           CALL   PUTYD-3
3C20 3E0F             MVI    A,0FH
3C22 A3               ANA    E
3C23 07               RLC
3C24 07               RLC
3C25 07               RLC
```

```
3C26 07                 RLC
3C27 B3                 ORA     E
3C28 5F                 MOV     E,A
3C29 79                 MOV     A,C
3C2A 07                 RLC
3C2B 4F                 MOV     C,A
3C2C 3AEF81             LDA     XZERO
3C2F 47                 MOV     B,A
3C30 AF         BXLOP:  XRA     A
3C31 B9                 CMP     C
3C32 C8                 RZ
3C33 0D                 DCR     C
3C34 2B                 DCX     H
3C35 79                 MOV     A,C
3C36 B8                 CMP     B
3C37 D8                 RC
3C38 CAC63D             JZ      PUTXZ
3C3B 0D                 DCR     C
3C3C CD133D             CALL    PXYCH
3C3F C3303C             JMP     BXLOP
3C42 CDE53B     BARYM:  CALL    INVY
3C45 21F081             LXI     H,YZERO
3C48 AE                 XRA     M
3C49 E6FC               ANI     252
3C4B 3AEC81             LDA     XDATA
3C4E CAAD3D             JZ      PUTZZ
3C51 1E57               MVI     E,BARTY AND 255
3C53 CDE83C             CALL    PUTYD-3
3C56 3AF081             LDA     YZERO
3C59 E6FC               ANI     252
3C5B 0F                 RRC
3C5C 0F                 RRC
3C5D 4F                 MOV     C,A     ;C=YZERO SHR 2
3C5E 7B                 MOV     A,E
3C5F 07                 RLC
3C60 1E0F               MVI     E,0FH
3C62 D2673C             JNC     $+5
3C65 1EF0               MVI     E,0F0H
3C67 79         BYLOP:  MOV     A,C
3C68 04                 INR     B
3C69 B8                 CMP     B
3C6A D8                 RC
3C6B CCCD3D             CZ      PUTYZ
3C6E 3E80               MVI     A,NOCHA*2
3C70 85                 ADD     L
3C71 6F                 MOV     L,A
3C72 D2763C             JNC     $+4
3C75 24                 INR     H
3C76 CD123D             CALL    PXYCH-1
3C79 C3673C             JMP     BYLOP
3C7C CD2C3D     PLINC:  CALL    INCXY
3C7F CAE53C             JZ      PUTYD-6
3C82 CDE53C             CALL    PUTYD-6
3C85 CD483D             CALL    MOVXY
3C88 C3E53C             JMP     PUTYD-6
3C8B CD2C3D     PBINY:  CALL    INCXY
3C8E CA973C             JZ      $+9
3C91 CD973C             CALL    $+6
```

```
3C94 CD483D            CALL    MOVXY
3C97 78                MOV     A,B
3C98 C3453C            JMP     BARYM+3
3C9B 21EE81  PUTXD:    LXI     H,ODDFL
3C9E FE80              CPI     NOCHA*2
3CA0 DAA53C            JC      $+5
3CA3 D680              SUI     NOCHA*2
3CA5 4F                MOV     C,A
3CA6 E601              ANI     1          ;ODD=1
3CA8 77                MOV     M,A        ;ODDFL
3CA9 2EEC              MVI     L,XDATA AND 255
3CAB 71                MOV     M,C
3CAC 79                MOV     A,C
3CAD 1F                RAR
3CAE 2ED8              MVI     L,PCRAD AND 255
3CB0 77                MOV     M,A
3CB1 C9                RET
3CB2 7B      PLOTX:    MOV     A,E
3CB3 EEFF              XRI     255
3CB5 C2C13C            JNZ     $+12
3CB8 77                MOV     M,A
3CB9 2ECF              MVI     L,VFILL+1 AND 255
3CBB 3E7F              MVI     A,7FH
3CBD A6                ANA     M
3CBE C3A23D            JMP     NROLL-1
3CC1 2EDA              MVI     L,PLOFL AND 255
3CC3 35                DCR     M
3CC4 CAFB3D            JZ      ASCPL
3CC7 34                INR     M
3CC8 FE10              CPI     16
3CCA DA2A3D            JC      INCXY-2
3CCD 7E                MOV     A,M
3CCE 07                RLC
3CCF 86                ADD     M
3CD0 21693D            LXI     H,PLTAB-6
3CD3 85                ADD     L
3CD4 6F                MOV     L,A
3CD5 7E                MOV     A,M        ;PLOFL NEW VALUE
3CD6 2C                INR     L
3CD7 4E                MOV     C,M        ;PLOT SUBR ADDR
3CD8 2C                INR     L
3CD9 46                MOV     B,M
3CDA 21DA81            LXI     H,PLOFL
3CDD 86                ADD     M
3CDE 77                MOV     M,A        ;PLOFL NEW VAL
3CDF 7B                MOV     A,E
3CE0 C5                PUSH    B
3CE1 47                MOV     B,A
3CE2 2EED              MVI     L,YDATA AND 255
3CE4 C9                RET

3CE5 79                MOV     A,C
3CE6 1E67              MVI     E,XYTAB AND 255
3CE8 CD9B3C            CALL    PUTXD
3CEB 2EEE    PUTYD:    MVI     L,ODDFL AND 255
3CED 78                MOV     A,B        ;YDATA
3CEE FE80              CPI     NOLIN*4
3CF0 DAF53C            JC      $+5
```

```
3CF3 D680              SUI     NOLIN*4
3CF5 47                MOV     B,A
3CF6 E603              ANI     3          ;A=0,1,2,3
3CF8 07                RLC
3CF9 83                ADD     E
3CFA 86                ADD     M
3CFB 6F                MOV     L,A
3CFC 263D              MVI     H,BARTY SHR 8
3CFE 5E                MOV     E,M        ;PLOT CHARACTER
3CFF 21ED81            LXI     H,YDATA
3D02 70                MOV     M,B
3D03 78                MOV     A,B
3D04 E6FC              ANI     252
3D06 0F                RRC
3D07 0F                RRC
3D08 47                MOV     B,A
3D09 2ED8              MVI     L,PCRAD AND 255
3D0B 4E                MOV     C,M
3D0C 2ECF              MVI     L,VFILL+1 AND 255
3D0E 56                MOV     D,M
3D0F CD9E3B            CALL    LN5Y
3D12 2C                INR     L
3D13 7E        PXYCh:  MOV     A,M        ;STATUS WORD
3D14 72                MOV     M,D
3D15 2D                DCR     L
3D16 AA                XRA     D
3D17 7E                MOV     A,M
3D18 73                MOV     M,E
3D19 F8                RM      ;OLD CHA NO PL
3D1A CA213D            JZ      ORCHA
3D1D EEFF              XRI     255
3D1F C8                RZ      ;OLD CH DIF COL
3D20 2F                CMA
3D21 AB        ORCHA:  XRA     E
3D22 77                MOV     M,A
3D23 3AE681            LDA     COLFL
3D26 A7                ANA     A
3D27 C0                RNZ
3D28 7B                MOV     A,E
3D29 B6                ORA     M
3D2A 77                MOV     M,A
3D2B C9                RET

               ;RETURN FOR PXYCH

3D2C 2AEC81    INCXY:  LHLD    XDATA
3D2F CD853B            CALL    SHIFF
3D32 22EC81            SHLD    XDATA
3D35 CAA43E            JZ      LL10
3D38 44                MOV     B,H
3D39 4D                MOV     C,L
3D3A 7B                MOV     A,E
3D3B CD853B            CALL    SHIFF
3D3E 22EA81            SHLD    XTWO
3D41 C0                RNZ
3D42 22EC81            SHLD    XDATA
3D45 44                MOV     B,H
3D46 4D                MOV     C,L
```

```
3D47 C9                 RET
3D48 2AEA81   MOVXY:    LHLD    XTWO
3D4B 22EC81             SHLD    XDATA
3D4E 44                 MOV     B,H
3D4F 4D                 MOV     C,L
3D50 C9                 RET

3D51 07       BARTZ:    DB      07
3D52 70                 DB      70H
3D53 06                 DB      06
3D54 60                 DB      60H
3D55 03                 DB      03
3D56 30                 DB      30H
3D57 0F       BARTY:    DB      0FH
3D58 F0                 DB      0F0H
3D59 0E                 DB      0EH
3D5A E0                 DB      0E0H
3D5B 0C                 DB      0CH
3D5C C0                 DB      0C0H
3D5D 08                 DB      8H
3D5E 80                 DB      80H
3D5F 01       BARTX:    DB      1
3D60 11                 DB      11H
3D61 02                 DB      2
3D62 22                 DB      22H
3D63 04                 DB      4H
3D64 44                 DB      44H
3D65 08                 DB      8H
3D66 88                 DB      88H
3D67 01       XYTAB:    DB      1
3D68 10                 DB      10H
3D69 02                 DB      2
3D6A 20                 DB      20H
3D6B 04                 DB      4
3D6C 40                 DB      40H
3D6D 08                 DB      8
3D6E 80                 DB      80H
3D6F 01       PLTAB:    DB      1
3D70 9B3C               DW      PUTXD   ; (2) PLOT X
3D72 FF                 DB      -1
3D73 FD3B               DW      PLPTY   ; (3) PLOT Y
3D75 00                 DB      0
3D76 7C3C               DW      PLINC   ; (4) INC-XY
3D78 01                 DB      1
3D79 F33B               DW      PXZER   ; (5) BAR X
3D7B 01                 DB      1
3D7C E53B               DW      INVY    ; (6) BAR X
3D7E FF                 DB      -1
3D7F 133C               DW      BARXM   ; (7) BAR X
3D81 00                 DB      0
3D82 053C               DW      PBINX   ; (8) INC BAR X
3D84 01                 DB      1
3D85 E33B               DW      INVY-2  ; (9) BAR Y
3D87 01                 DB      1
3D88 9B3C               DW      PUTXD   ;(10) BAR Y
3D8A FF                 DB      -1
3D8B 423C               DW      BARYM   ;(11) BAR Y
3D8D 00                 DB      0
```

```
3D8E 8B3C              DW      PBINY   ;(12) INC BAR Y
3D90 01                DB      1
3D91 F33B              DW      PXZER   ;(13) VECT XO
3D93 FF                DB      -1
3D94 BE3B              DW      VECTY   ;(14) VECT YO
3D96 00                DB      0
3D97 C43B              DW      INVEC   ;(15) INC VECT

3D99 73        POTON:  MOV     M,E
3D9A 2EDA              MVI     L,PLOFL AND 255
3D9C 73                MOV     M,E
3D9D 2ECF              MVI     L,VFILL+1 AND 255
3D9F 3E80              MVI     A,80H
3DA1 B6                ORA     M
3DA2 77                MOV     M,A
3DA3 AF        NROLL:  XRA     A
3DA4 32CD81            STA     ROLLN
3DA7 CDFC37            CALL    GETBC
3DAA C37938            JMP     SVCRS+6

3DAD 4F        PUTZZ:  MOV     C,A
3DAE 78                MOV     A,B
3DAF BE                CMP     M
3DB0 CAE53C            JZ      PUTYD-6
3DB3 3E03              MVI     A,3
3DB5 A6                ANA     M
3DB6 FE02              CPI     2
3DB8 79                MOV     A,C
3DB9 1E51              MVI     E,BARTZ AND 255
3DBB CAE83C            JZ      PUTYD-3
3DBE 1E55              MVI     E,BARTZ+4 AND 255
3DC0 FAE83C            JM      PUTYD-3
3DC3 C3513C            JMP     BARYM+15

3DC6 3EF0      PUTXZ:  MVI     A,0F0H
3DC8 A3                ANA     E
3DC9 5F                MOV     E,A
3DCA C3133D            JMP     PXYCH
3DCD 3AF081    PUTYZ:  LDA     YZERO
3DD0 E603              ANI     3
3DD2 CAE43D            JZ      PUTEZ-2
3DD5 FE02              CPI     2
3DD7 3E77              MVI     A,77H
3DD9 CAE63D            JZ      PUTEZ
3DDC 3EFF              MVI     A,0FFH
3DDE F2E63D            JP      PUTEZ
3DE1 3E33              MVI     A,33H
3DE3 F2E63D            JP      PUTEZ   ;TRICK
3DE6 (3DE4)            ORG     $-2

3DE4 3E11              MVI     A,11H
3DE6 A3        PUTEZ:  ANA     E
3DE7 5F                MOV     E,A
3DE8 C9                RET

3DE9 3EDF      PLOKB:  MVI     A,KBDFL AND 255
3DEB BD                CMP     L
3DEC C0                RNZ
```

```
3DED 3EB0              MVI     A,0B0H
3DEF A3                ANA     E
3DF0 EEB0              XRI     0B0H
3DF2 C0                RNZ
3DF3 7B                MOV     A,E
3DF4 F6F0              ORI     0F0H
3DF6 5F                MOV     E,A
3DF7 32FE81            STA     KBCHA
3DFA C9                RET
3DFB 34        ASCPL:  INR     M
3DFC 7B                MOV     A,E
3DFD CDE839            CALL    STOR1-3
3E00 C3E781            JMP     JUMP

3E03 7B        CPLOX:  MOV     A,E
3E04 FE10              CPI     16          ;32 FOR#
3E06 DADC39            JC      SVCHA
3E09 FE18              CPI     24          ;31H FOR #
3E0B D2DC39            JNC     SVCHA       ;RNC FOR #
3E0E E607              ANI     7
3E10 C6FC              ADI     0FCH
3E12 17                RAL
3E13 C667              ADI     XYTAB AND 255
3E15 6F                MOV     L,A
3E16 263D              MVI     H,XYTAB SHR 8
3E18 5E                MOV     E,M
3E19 2AD281            LHLD    VHLAD
3E1C 2C                INR     L
3E1D 3ACF81            LDA     VFILL+1
3E20 F680              ORI     80H
3E22 57                MOV     D,A
3E23 7E                MOV     A,M
3E24 72                MOV     M,D
3E25 2D                DCR     L
3E26 AA                XRA     D
3E27 7E                MOV     A,M
3E28 73                MOV     M,E
3E29 F8                RM
3E2A AB                XRA     E
3E2B 77                MOV     M,A
3E2C C9                RET

                       ; START POINT   X1-XDATA, Y1-YDATA
                       ; END POINT     X0-XZERO, Y0-YZERO

3E2D 2D        VECTO:  DCR     L           ;POINT TO X0
3E2E 4E                MOV     C,M         ;GET X0
3E2F 2AEC81            LHLD    XDATA       ;L=X1, H=Y1
3E32 22EA81            SHLD    XTWO        ;STORE PRESENT POSITION
3E35 E5                PUSH    H           ;SAVE PRESENT POSITION
3E36 78                MOV     A,B         ;GET NEW Y
3E37 94                SUB     H           ;COMPUTE YCHANGE
3E38 2601              MVI     H,1         ;ASSUME YSTEP OF +1
3E3A D2413E            JNC     LL1         ;YCHANGE IS + OR 0
3E3D 26FF              MVI     H,-1        ;SET YSTEP OF -1
3E3F 2F                CMA                 ;COMPUTE MAGNITUDE ...
3E40 3C                INR     A           ;... OF YCHANGE
3E41 57        LL1:    MOV     D,A         ;SAVE YCHANGE MAGNITUDE
```

```
3E42 79              MOV     A,C       ;GET NEW X
3E43 95              SUB     L         ;COMPUTE XCHANGE
3E44 2E01            MVI     L,1       ;ASSUME XSTEP OF +1
3E46 D24D3E          JNC     LL2       ;XCHANGE IS + OR 0
3E49 2EFF            MVI     L,-1      ;SET XSTEP OF -1
3E4B 2F              CMA               ;COMPUTE MAGNITUDE ...
3E4C 3C              INR     A         ;... OF XCHANGE
3E4D 5F        LL2:  MOV     E,A       ;SAVE XCHANGE MAGNITUDE
3E4E 22F281          SHLD    TEMPO     ;STORE XSTEP AND YSTEP
3E51 C1              POP     B         ;GET PRESENT POSITION
3E52 BA              CMP     D         ;XCHANGE >= YCHANGE ?
3E53 D27D3E          JNC     LL6       ;YES
3E56 7A              MOV     A,D       ;GET YCHANGE
3E57 A7              ANA     A         ;CLEAR C-BIT
3E58 1F              RAR               ;HALVE YCHANGE
3E59 93        LL3:  SUB     E         ;SUBTRACT SMALL CHANGE
3E5A D2653E          JNC     LL4       ;DON'T STEP X THIS TIME
3E5D 82              ADD     D         ;ADD LARGE CHANGE
3E5E F5              PUSH    PSW       ;SAVE COUNTER
3E5F 3AF281          LDA     TEMPO     ;GET XSTEP
3E62 81              ADD     C         ;ADD X
3E63 4F              MOV     C,A       ;SET NEW X
3E64 3EF5            MVI     A,0F5H    ;GO STEP Y
3E66 (3E65)          ORG     $-1       ;TRICK SKIP TO LL5
3E65 F5        LL4:  PUSH    PSW       ;SAVE COUNTER
3E66 3AF381    LL5:  LDA     TEMP1     ;GET YSTEP
3E69 80              ADD     B         ;ADD Y
3E6A 47              MOV     B,A       ;SET NEW Y
3E6B D5              PUSH    D         ;SAVE D&E
3E6C C5              PUSH    B         ;SAVE B&C
3E6D CDE53C          CALL    PUTYD-6   ;PLOT THIS POINT
3E70 C1              POP     B         ;RESTORE B&C
3E71 D1              POP     D         ;RESTORE D&E
3E72 3AF081          LDA     YZERO     ;GET END Y
3E75 B8              CMP     B         ;FINISHED ?
3E76 CAA43E          JZ      LL10      ;YES
3E79 F1              POP     PSW       ;RESTORE COUNTER
3E7A C3593E          JMP     LL3       ;LOOP
3E7D B7        LL6:  ORA     A         ;XCHANGE = 0 ?
3E7E C8              RZ                ;YES: RETURN TO CALLER
3E7F 1F              RAR               ;HALVE XCHANGE (C IS 0)
3E80 92        LL7:  SUB     D         ;SUBTRACT SMALL CHANGE
3E81 D28C3E          JNC     LL8       ;DON'T STEP Y THIS TIME
3E84 83              ADD     E         ;ADD LARGE CHANGE
3E85 F5              PUSH    PSW       ;SAVE COUNTER
3E86 3AF381          LDA     TEMP1     ;GET YSTEP
3E89 80              ADD     B         ;ADD Y
3E8A 47              MOV     B,A       ;SET NEW Y
3E8B 3EF5            MVI     A,0F5H    ;GO STEP X
3E8D (3E8C)          ORG     $-1       ;TRICK SKIP TO LL9
3E8C F5        LL8:  PUSH    PSW       ;SAVE COUNTER
3E8D 3AF281    LL9:  LDA     TEMPO     ;GET XSTEP
3E90 81              ADD     C         ;ADD X
3E91 4F              MOV     C,A       ;SET NEW X
3E92 D5              PUSH    D         ;SAVE D&E
3E93 C5              PUSH    B         ;SAVE B&C
3E94 CDE53C          CALL    PUTYD-6   ;PLOT THIS POINT
3E97 C1              POP     B         ;RESTORE B&C
```

```
3E98 D1              POP     D          ;RESTORE D&E
3E99 3AEF81          LDA     XZERO      ;GET END X
3E9C B9              CMP     C          ;FINISHED ?
3E9D CAA43E          JZ      LL10       ;YES
3EA0 F1              POP     PSW        ;RESTORE COUNTER
3EA1 C3803E          JMP     LL7        ;LOOP
3EA4 F1      LL10:   POP     PSW        ;CLEAN STACK
3EA5 C9              RET                ;RETURN TO CALLER

3EA6 CDD03F  TIM3X:  CALL    SAVE
3EA9 CDB33E          CALL    KEYBD
3EAC D8              RC
3EAD CA3B00          JZ      BREAK
3EB0 C32B00          JMP     KEYCO

3EB3 3A1D00  KEYBD:  LDA     KEDEL
3EB6 D30B            OUT     TIME3
3EB8 01FF0F  Q01:    LXI     B,0FFFH    ;INIT. ROW COUNTER & 1ST KEYCODE
3EBB 59              MOV     E,C        ;INITIALIZE 2ND KEYCODE
3EBC 78      Q02:    MOV     A,B        ;GET COMPLEMENT OF NEXT ROW NUMBER
3EBD D307            OUT     EXTOT      ;SEND
3EBF DB01            IN      EXTIN      ;READ
3EC1 3C              INR     A          ;GOT ANY ?
3EC2 CAEF3E          JZ      Q06        ;NO !
3EC5 1C              INR     E          ;ALREADY SEEN TWO KEYS ?
3EC6 C2FE3E          JNZ     ROT        ;YES: MORE THAN TWO KEYS: IGNORE !
3EC9 3D      Q03:    DCR     A          ;RESTORE
3ECA 37              STC                ;SET <C> BIT
3ECB 1C      Q04:    INR     E
3ECC 1F              RAR                ;FOUND IT ?
3ECD DACB3E          JC      Q04        ;NO !
3ED0 F5              PUSH    PSW
3ED1 7B              MOV     A,E        ;GET GROUP CODE
3ED2 87              ADD     A          ;POSITION IT ...
3ED3 87              ADD     A
3ED4 87              ADD     A
3ED5 87              ADD     A
3ED6 B0              ORA     B          ;MERGE COMPLEMENT OF ROW NUMBER
3ED7 EE0F            XRI     0FH        ;MAKE ROW NUMBER RIGHT
3ED9 0C              INR     C          ;IS THIS 1ST KEY SEEN ON THIS ROW ?
3EDA CAE73E          JZ      Q05        ;YES !
3EDD 5F              MOV     E,A        ;COPY 2ND KEYCODE SEEN
3EDE F1              POP     PSW
3EDF 3C              INR     A          ;ANY MORE ON THIS ROW ?
3EE0 C2FE3E          JNZ     ROT        ;YES: MORE THAN TWO KEYS !
3EE3 0D              DCR     C          ;RESTORE 1ST KEYCODE SEEN
3EE4 C3EF3E          JMP     Q06
3EE7 4F      Q05:    MOV     C,A        ;SAVE 1ST KEYCODE SEEN
3EE8 F1              POP     PSW
3EE9 3C              INR     A          ;ANY MORE ON THIS ROW ?
3EEA C2C93E          JNZ     Q03        ;YES !
3EED 1EFF            MVI     E,-1       ;RE-INIT. 2ND KEYCODE
3EEF 05      Q06:    DCR     B          ;SCANNED ALL ROWS ?
3EF0 F2BC3E          JP      Q02        ;NO: READ NEXT ROW !
3EF3 0C              INR     C          ;SEE ANY KEYS ?
3EF4 C2073F          JNZ     Q08        ;YES !
3EF7 AF      Q07:    XRA     A
3EF8 32E481          STA     LKC        ;CLEAR "LAST" KEYCODE
```

```
3EFB 32E581              STA     NKC       ;CLEAR "NEW" KEYCODE
                         ;
3EFE 3AD681    ROT:      LDA     EXTBF
3F01 D307                OUT     EXTOT     ;RESTORE OUTPUT PORT
3F03 F601                ORI     1
3F05 37                  STC
3F06 C9                  RET               ;EXIT !
                         ;
3F07 0D        Q08:      DCR     C         ;RESTORE 1ST KEYCODE
3F08 1C                  INR     E         ;SEE TWO KEYS ?
3F09 C22C3F              JNZ     Q09       ;YES !
               ; SAW ONE KEY :
3F0C AF                  XRA     A
3F0D 32E581              STA     NKC       ;CLEAR "NEW" KEYCODE
3F10 3AE481              LDA     LKC       ;GET "LAST" KEYCODE
3F13 B9                  CMP     C         ;MATCH ?
3F14 C23F3F              JNZ     Q10       ;NO .!
3F17 3E80                MVI     A,80H
3F19 D307                OUT     EXTOT     ;SELECT 'B' HIBITS
3F1B DB01                IN      EXTIN     ;READ
3F1D E640                ANI     40H       ;REPEAT ?
3F1F C2FE3E              JNZ     ROT       ;NO !
3F22 21D781              LXI     H,SEC15   ;POINT TO DELAY COUNTER
3F25 35                  DCR     M         ;IS IT TIME FOR ANOTHER ?
3F26 C2FE3E              JNZ     ROT       ;NO: NOT YET !
3F29 C3433F              JMP     Q11
               ; SAW TWO KEYS :
3F2C 1D        Q09:      DCR     E         ;RESTORE 2ND KEYCODE
3F2D 3AE481              LDA     LKC       ;GET "LAST" KEYCODE
3F30 CDC23F              CALL    Q20       ;MATCH ?
3F33 CAFE3E              JZ      ROT       ;YES: DO NOTHING !
3F36 3AE581              LDA     NKC       ;GET "NEW" KEYCODE
3F39 CDC23F              CALL    Q20       ;MATCH ?
3F3C C2F73E              JNZ     Q07       ;NO !
3F3F 79        Q10:      MOV     A,C       ;GET KEYCODE
3F40 32E481              STA     LKC       ;SAVE "LAST" KEYCODE
3F43 3A1C00    Q11:      LDA     CHTIM     ;SCANS/CHAR. IN REPEAT MODE
3F46 32D781              STA     SEC15     ;SET REPEAT TIME COUNTER
3F49 79                  MOV     A,C       ;GET KEY CODE
3F4A FE50                CPI     50H       ;SPECIAL GROUP ?
3F4C DA543F              JC      Q12       ;NO: NORMAL !
3F4F FE56                CPI     56H       ;SPECIAL GROUP ?
3F51 DAAC3F              JC      Q16       ;YES !
3F54 0F        Q12:      RRC               ;ROTATE ...
3F55 0F                  RRC
3F56 0F                  RRC
3F57 0F                  RRC
3F58 E607                ANI     7         ;GET GROUP CODE
3F5A 5F                  MOV     E,A       ;COPY GROUP CODE
3F5B 3E80                MVI     A,80H
3F5D D307                OUT     EXTOT     ;SELECT 'B' HIBITS
3F5F DB01                IN      EXTIN     ;READ
3F61 E630                ANI     30H       ;GET CONTROL & SHIFT BITS
3F63 07                  RLC
3F64 07                  RLC
3F65 B3                  ORA     E         ;MERGE WITH GROUP CODE
3F66 07                  RLC
3F67 07                  RLC
```

```
3F68 5F              MOV     E,A
3F69 1600            MVI     D,0
3F6B 21DA3F          LXI     H,KTAB-4 ;POINT TO TRANSLATE TABLE
3F6E 19              DAD     D        ;INDEX
3F6F 3E0F            MVI     A,0FH
3F71 A1              ANA     C        ;GET ROW NUMBER
3F72 B6              ORA     M        ;MERGE WITH TABLE ENTRY
3F73 5F              MOV     E,A      ;SAVE
3F74 79              MOV     A,C      ;GET KEYCODE
3F75 FE7C            CPI     7CH
3F77 D2813F          JNC     Q13      ;WEIRD !
3F7A D61C            SUI     1CH
3F7C FE04            CPI     20H-1CH
3F7E D2853F          JNC     Q14      ;NOT WEIRD !
3F81 3E10    Q13:    MVI     A,10H
3F83 AB              XRA     E        ;FIX WEIRD
3F84 5F              MOV     E,A
3F85 DB01    Q14:    IN      EXTIN    ;READ 'B' HIBITS
3F87 A7              ANA     A        ;CAPS LOCK ?
3F88 F29E3F          JP      Q15      ;YES: O.K. !
3F8B 7B              MOV     A,E      ;GET ASCII CODE
3F8C D641            SUI     'A'
3F8E FE3A            CPI     'Z'+20H+1-'A'
3F90 D29E3F          JNC     Q15      ;NOT ALPHA !
3F93 D61A            SUI     'Z'+1-'A'
3F95 FE06            CPI     'A'+20H-'Z'-1
3F97 DA9E3F          JC      Q15      ;NOT ALPHA !
3F9A 3E20            MVI     A,20H
3F9C AB              XRA     E        ;FLIP SHIFT
3F9D 5F              MOV     E,A
3F9E CDFE3E  Q15:    CALL    ROT      ;RESTORE OUTPUT PORT
3FA1 3E80            MVI     A,80H    ;SET UP READY FLAG
3FA3 32FF81          STA     READY
3FA6 B7              ORA     A        ;SET SIGN BIT = GOOD KEY
3FA7 7B              MOV     A,E      ;GET CHAR.
3FA8 32FE81          STA     KBCHA    ;STORE IT
3FAB C9              RET              ; AND EXIT
;
3FAC 32FF81  Q16:    STA     READY    ;SET READY = 50H IF BREAK
3FAF D650            SUI     50H      ;ADJUST CODE
3FB1 37              STC              ; CARRY AND ZERO IS BREAK
3FB2 C8              RZ               ;CODE IS BREAK !
3FB3 1E7F            MVI     E,127    ;MAY BE DELETE
3FB5 FE04            CPI     4        ;IS IT DELETE ?
3FB7 CA9E3F          JZ      Q15      ;YES: GO DO IT !
3FBA 5F              MOV     E,A
3FBB 3E06            MVI     A,6      ;MAP 5==>1,3==>3,2==>4,1==>5
3FBD 93              SUB     E
3FBE 5F              MOV     E,A
3FBF C39E3F          JMP     Q15      ;PROCESS IT !
;
3FC2 B9      Q20:    CMP     C        ;MATCH THIS ONE ?
3FC3 CACB3F          JZ      Q21      ;YES !
3FC6 69              MOV     L,C      ;SWITCH THEM ...
3FC7 4B              MOV     C,E
3FC8 5D              MOV     E,L
3FC9 B9              CMP     C        ;MATCH THIS ONE ?
3FCA C0              RNZ              ;NO MATCH: EXIT <NZ> !
```

```
3FCB 7B         G21:    MOV     A,E     ;GET KEYCODE THAT DOES NOT MATCH
3FCC 32E581             STA     NKC     ;STORE IT AS "NEW" KEYCODE
3FCF C9                 RET             ;MATCH: EXIT <Z> !

3FD0 E3         SAVE:   XTHL    ;SAVE HL WHILE LOADING ADDRESS
3FD1 D5                 PUSH    D       ; TO RESUME EXECUTION AT,
3FD2 C5                 PUSH    B       ;THEN SAVE ALL REGISTERS
3FD3 F5                 PUSH    PSW     ; AND STATUS
3FD4 CDDD3F             CALL    JMPHL   ;PUSH $+3 AS RETURN ADDRESS
3FD7 F1                 POP     PSW     ; AND RESUME EXECUTION OF CALLER
3FD8 C1                 POP     B       ;THEN RESTORE ALL REGISTERS
3FD9 D1                 POP     D       ; AND STATUS
3FDA E1                 POP     H
3FDB FB                 EI              ;REENABLE INTERUPTS
3FDC C9                 RET             ; AND EXIT FROM INTERUPT

3FDD E9         JMPHL:  PCHL            ; JUMP THRU HERE TO ROUTINE

3FDE A0         KTAB:   DB      0A0H    ; 1 CS
3FDF 20                 DB      020H    ; 1 -S
3FE0 B0                 DB      0B0H    ; 1 C-
3FE1 30                 DB      030H    ; 1 --
3FE2 80                 DB      080H    ; 2 CS
3FE3 60                 DB      060H    ; 2 -S
3FE4 00                 DB      000H    ; 2 C-
3FE5 40                 DB      040H    ; 2 --
3FE6 90                 DB      090H    ; 3 CS
3FE7 70                 DB      070H    ; 3 -S
3FE8 10                 DB      010H    ; 3 C-
3FE9 50                 DB      050H    ; 3 --
3FEA E0                 DB      0E0H    ; 4 CS
3FEB D0                 DB      0D0H    ; 4 -S
3FEC C0                 DB      0C0H    ; 4 C-
3FED F0                 DB      0F0H    ; 4 --
3FEE 80                 DB      080H    ; 5 CS
3FEF 80                 DB      080H    ; 5 -S
3FF0 00                 DB      000H    ; 5 C-
3FF1 00                 DB      000H    ; 5 --
3FF2 90                 DB      090H    ; 6 CS
3FF3 90                 DB      090H    ; 6 -S
3FF4 10                 DB      010H    ; 6 C-
3FF5 10                 DB      010H    ; 6 --
3FF6 20                 DB      020H    ; 7 CS
3FF7 20                 DB      020H    ; 7 -S
3FF8 20                 DB      020H    ; 7 C-
3FF9 20                 DB      020H    ; 7 --

E 3FFA (FFF4)   ZZZZZZ: END OF SYSTEM ***********************************

1 ERRORS
FCS>
```

* UAT ENDS AT 3FFF