


```

0005' 000E' + .WORD ..EDI
0007' 0011' + .WORD ..EDA
0009' 4544495400 + .ASCIZ /EDIT/
+ ]

000E' CD 0000:05 ..EDI: CALL ARG
0011' 0A00 ..EDA: .BYTE $NAME,$TAF
0013' E5 PUSH H
0014' FD6E01 MOV L,1(Y)
0017' FD6602 MOV H,2(Y)
001A' 22 64A1 SHLD EDNAME
001D' 7E MOV A,M
001E' FE02 CPI $NULL
0020' 2020 JRNZ ..NOTN

; CREATE A NULL STRING AND MAKE THE NAME POINT AT IT
0022' 3E01 MVI A,1
0024' CD 0000:04 CALL ALSTS
0027' E5 PUSH H
0028' AF XRA A
0029' 12 STAX D
002A' 1B DCX D
002B' 12 STAX D ; GET EARLIER CHAR TOO
002C' 11 0006 LXI D,$SLEN
002F' 19 DAD D
0030' 77 MOV M,A
0031' 23 INX H
0032' 77 MOV M,A
0033' 2A 64A1 LHLD EDNAME
0036' D1 POP D
0037' E5 PUSH H
0038' 360A MVI M,$NAMADR
003A' 01 0005 LXI B,$NVALUE
003D' 09 DAD B
003E' 73 MOV M,E
003F' 23 INX H
0040' 72 MOV M,D
0041' E1 POP H

; MAKE DAMN SURE WE GOT A STRING
0042' CD 0000:11 ..NOTN: CALL NXTVAL
0045' 22 681A SHLD EDSTR
0048' 7E MOV A,M
0049' A7 ANA A
004A' FE08 CPI $STRADR
NZERROR ER.FNFI

004C' 2804 + JRZ ..0001
004E' CD 0000:0B + CALL ERRPGM
0051' 1C + .BYTE ER.FNF
0052' + ..0001: ]

; INITIALIZE OODLES OF POINTERS
0052' DD21 67CD LXI X,SCRWIN
0056' AF XRA A
0057' 32 681C STA EDMODE
005A' 01 000A LXI B,$SASCII
005D' 09 DAD B
005E' 22 6818 SHLD TOP
0061' 22 64A3 SHLD FIRST

```

```

0064' 22 64A5          SHLD    CURRENT
0067' 22 64A7          SHLD    POINTER
; GET ACCURATE COUNT ON STRING LENGTH
006A' 11 0000          LXI     D,0
006D' 7E               ..CNTL: MOV    A,M
006E' A7               ANA     A
006F' 2804            JRZ     ..GOTN
0071' 13               INX     D
0072' 23               INX     H
0073' 18F8            JMPR   ..CNTL
0075' ED53 6800       ..GOTN: SDED   STRSIZ
0079' CD 03AC'        CALL   REPAINT
007C' CD 0000:14      CALL   RESCX
007F' CD 0000:12      CALL   PUTCUR
; EDITOR CHARACTER GET AND DOIT LOOP
0082' CD 0000:07     ..LOOP: CALL  CNTCK
0085' CD 0000:0D      CALL  GETCH
0088' CD 008D'        CALL  EDDISP
008B' 18F5            JMPR   ..LOOP
008D'                ..EDX:
; ROUTINE TO DISPATCH ON INPUT CHARACTER OR CONTROL CHAR
ACTER
EDDISP:
008D' 281F            JRZ     ..BIZR
008F' 060A            MVI    B,10
0091' 21 00E3'        LXI    H,EDCUNT
0094' BE               ..GETA: CMP    M
0095' 23               INX     H
0096' 2009            JRNZ   ..NOEQ
0098' 5E               MOV    E,M
0099' 23               INX     H
009A' 56               MOV    D,M
009B' AF              XRA    A
009C' 32 681C         STA    EDMODE
009F' EB              XCHG
00A0' E9              PCHL
00A1' 23               ..NOEQ: INX     H
00A2' 23               INX     H
00A3' 10EF            DJNZ   ..GETA
00A5' A7              ANA    A
00A6' C2 014D'        JNZ    INSREP
00A9' 3E0A            MVI    A,NL
00AB' C3 014D'        JMP    INSREP
; CHECK TOMMYS WIERDO TABLE
00AE' 21 00C7'       ..BIZR: LXI    H,EDCLIT
00B1' 0607            MVI    B,7
00B3' 5E               ..GETZ: MOV    E,M
00B4' 23               INX     H
00B5' 56               MOV    D,M
00B6' 23               INX     H
00B7' 1A              LDAX   D          ; GET IT
00B8' A7              ANA    A          ; ZERO?
00B9' 2807            JRZ     ..NOTZ
00BB' AF              XRA    A
00BC' 12              STAX   D

```

```

00BD' 5E          MOV      E,M
00BE' 23          INX      H
00BF' 56          MOV      D,M
00C0' EB          XCHG
00C1' E9          PCHL
00C2' 23          ..NOTZ: INX      H
00C3' 23          INX      H
00C4' 10ED       DJNZ    ..GETZ
00C6' C9          RET

```

; OLD KEYBOARD TABLE

```

00C7'          EDCLIT:
00C7' 65D2 0101' WORD FCNTH,CURBAC
00CB' 65D4 0119' WORD FCNTJ,CURDWN
00CF' 65D3 0123' WORD FCNTI,INSERT
00D3' 65D5 0111' WORD FCNTK,CURUP
00D7' 65D6 0109' WORD FCNTL,CURFOR
00DB' 65E0 018F' WORD FCNTV,DELLIN
00DF' 65E3 012E' WORD FCNTY,DELCHR

```

; DISPATCH TABLE

```

00E3'          EDCUNT:
00E3' 1B          .BYTE   ESCAPE
00E4' 01CE'      .WORD   INSLIN
00E6' 80          .BYTE   80H
00E7' 0111'      .WORD   CURUP
00E9' 81          .BYTE   81H
00EA' 0119'      .WORD   CURDWN
00EC' 82          .BYTE   82H
00ED' 0101'      .WORD   CURBAC
00EF' 83          .BYTE   83H
00F0' 0109'      .WORD   CURFOR
00F2' 84          .BYTE   84H
00F3' 0123'      .WORD   INSERT
00F5' 85          .BYTE   85H
00F6' 01CE'      .WORD   INSLIN
00F8' 86          .BYTE   86H
00F9' 012E'      .WORD   DELCHR
00FB' 7F          .BYTE   RUBOUT
00FC' 0129'      .WORD   BACDEL
00FE' 15          .BYTE   CNTRLU
00FF' 018F'      .WORD   DELLIN

```

```

0101' CD 0000:12 CURBAC: CALL  PUTCUR
0104' CD 047A'   CALL  BACKC
0107' 1816      JMPR  PCRET
0109' CD 0000:12 CURFOR: CALL  PUTCUR
010C' CD 03E6'   CURF01: CALL  FORWC
010F' 180E      JMPR  PCRET
0111' CD 0000:12 CURUP:  CALL  PUTCUR
0114' CD 049B'   CALL  BACKL
0117' 1806      JMPR  PCRET
0119' CD 0000:12 CURDWN: CALL  PUTCUR
011C' CD 03FE'   CURDW1: CALL  FORWL
011F' CD 0000:12 PCRET:  CALL  PUTCUR
0122' C9          RET
0123' 3E01      INSERT: MVI   A,1
0125' 32 681C   STA   EDMODE

```

*.Byte 15FH
 word BACDEL*

```

0128'  C9          ENDINS: RET
                   ; BACKWARDS DELETE A CHARACTER
0129'  CD 012E'   BACDEL: CALL  DELCHR
012C'  18D3       JMPR    CURBAC
                   ; DELETE A CHARACTER!
012E'  2A 64A7   DELCHR: LHLD  POINTER
0131'  7E        MOV    A,M
0132'  A7        ANA   A
0133'  C8        RZ           ; QUIT IF AT NULL
0134'  CD 0000:12 CALL  PUTCUR           ; ERASE CURSE
0137'  3E02      MVI    A,2
0139'  32 6812   STA   EDLONG
013C'  01 FFFF   LXI   B,-1           ; -1 SIZE CHANGE
013F'  CD 01E8'   CALL  GNSTR
0142'  23        INX   H
0143'  CD 0261'   CALL  COPON
0146'  CD 0269'   CALL  SUPDAT
0149'  CD 0000:12 CALL  PUTCUR
014C'  C9        RET
                   ; INSERT - REPLACE CHARACTER
                   ;
                   ; IF POINTING AT NULL - INSERT
014D'  F5        INSREP: PUSH  PSW
014E'  CD 0000:12 CALL  PUTCUR
0151'  3E01      MVI    A,1
0153'  32 6812   STA   EDLONG
0156'  2A 64A7   LHLD  POINTER
0159'  7E        MOV    A,M
015A'  A7        ANA   A
015B'  281C      JRZ   ..INS
015D'  FE0A      CPI   NL           ; OR POINTING AT CR
015F'  2818      JRZ   ..INS
0161'  3A 681C   LDA   EDMODE
0164'  A7        ANA   A
0165'  2012      JRNZ  ..INS
                   ; REPLACE
                   ; TRANSLATE REPLACE OF CR INTO DOWN A LINE
0167'  F1        POP   PSW
0168'  FE0A      CPI   NL
016A'  CA 011C'   JZ    CURDW1
016D'  F5        PUSH  PSW
016E'  01 0000   LXI   B,0
0171'  CD 01E8'   CALL  GNSTR
0174'  F1        POP   PSW
0175'  12        STAX  D
0176'  23        INX   H
0177'  180C      JMPR  ..JOIN
                   ; INSERT
0179'  01 0001   ..INS: LXI   B,1
017C'  79        MOV    A,C           ; C=1 !!!
017D'  32 681C   STA   EDMODE           ; SET INSERTETH FLAG
0180'  CD 01E8'   CALL  GNSTR
0183'  F1        POP   PSW
0184'  12        STAX  D
0185'  13        ..JOIN: INX   D

```

```

0186'   CD 0261'           CALL   COPON
0189'   CD 0269'           CALL   SUPDAT
018C'   C3 010C'           JMP    CURFO1
018F'   2A 64A5           DELLIN: LHL   CURRENT
0192'   22 64A7           SHLD  POINTER
0195'   7E                 MOV   A,M
0196'   A7                 ANA   A
0197'   C8                 RZ
0198'   E5                 PUSH  H
0199'   CD 0000:12        CALL  PUTCUR
019C'   CD 0000:13        CALL  RESCX
019F'   01 0000           LXI  B,0
01A2'   E1                 POP   H
01A3'   7E                 ..LOOP: MOV  A,M
01A4'   A7                 ANA   A
01A5'   2808              JRZ   ..HIT
01A7'   03                 INX  B
01A8'   FE0A              CPI   NL
01AA'   2803              JRZ   ..HIT
01AC'   23                 INX  H
01AD'   18F4              JMPR  ..LOOP
01AF'   C5                 ..HIT: PUSH B
01B0'   CD 01C6'           CALL  NEGBC
01B3'   3E02              MVI  A,2
01B5'   32 6812           STA  EDLONG
01B8'   CD 01E8'           CALL  GNSTR
01BB'   C1                 POP   B
01BC'   09                 DAD  B
01BD'   CD 0261'           CALL  COPON
01C0'   CD 0269'           CALL  SUPDAT
01C3'   C3 011F'           JMP   PCRET

; NEGATE BC
01C6'   78                 NEGBC: MOV  A,B
01C7'   2F                 CMA
01C8'   47                 MOV  B,A
01C9'   79                 MOV  A,C
01CA'   2F                 CMA
01CB'   4F                 MOV  C,A
01CC'   03                 INX  B
01CD'   C9                 RET

; INSERT A LINE
01CE'   CD 0000:12        INSLIN: CALL  PUTCUR
01D1'   CD 0000:13        CALL  RESCX
01D4'   2A 64A5           LHL   CURRENT
01D7'   22 64A7           SHLD  POINTER
01DA'   CD 03FE'           CALL  FORWL

; INSERT A CR - TURN ON INSERT MODE
01DD'   CD 0123'           CALL  INSERT
01E0'   3E0A              MVI  A,NL
01E2'   CD 014D'           CALL  INSREP
01E5'   C3 0101'           JMP   CURBAC

; GET NEW STRING
; INPUT: BC=DESIRED SIZE DIFERENTIAL
01E8'   2A 6800           GNSTR: LHL   STRSIZ
01EB'   09                 DAD  B

```

01EC	22 6800	SHLD	STRSIZ
01EF	C5	PUSH	B
01F0	01 000C	LXI	B,\$SASCII+2
01F3	09	DAD	B
01F4	CD 0000:0A	CALL	DIV16
01F7	CD 0000:04	CALL	ALSTS
01FA	22 6802	SHLD	NSADDR
01FD	C1	POP	B
01FE	D5	PUSH	D
01FF	2A 6818	LHLD	TOP
0202	ED5B 64A3	LDED	FIRST
0206	A7	ANA	A
0207	ED52	DSBC	D
0209	D1	POP	D
020A	D5	PUSH	D
020B	19	DAD	D
020C	22 6818	SHLD	TOP
020F	2A 64A5	LHLD	CURRENT
0212	ED5B 64A3	LDED	FIRST
0216	A7	ANA	A
0217	ED52	DSBC	D
0219	D1	POP	D
021A	D5	PUSH	D
021B	19	DAD	D
021C	22 64A5	SHLD	CURRENT
021F	2A 64A9	LHLD	BOTTOM
0222	ED5B 64A3	LDED	FIRST
0226	A7	ANA	A
0227	ED52	DSBC	D
0229	D1	POP	D
022A	D5	PUSH	D
022B	19	DAD	D
022C	ED5B 64A5	LDED	CURRENT
0230	CD 0000:08	CALL	CPHLDE
0233	2801	JRZ	..SKUK
0235	09	DAD	B
0236	D1	POP	D
0237	22 64A9	SHLD	BOTTOM
023A	22 6810	SHLD	NEWBOT
023D	2A 64A3	LHLD	FIRST
0240	ED53 64A3	SDED	FIRST
0244	D5	..CLOP: PUSH	D
0245	ED5B 64A7	LDED	POINTER
0249	CD 0000:08	CALL	CPHLDE
024C	D1	POP	D
024D	2806	JRZ	..COPD
024F	7E	MOV	A,M
0250	12	STAX	D
0251	13	INX	D
0252	23	INX	H
0253	18EF	JMPR	..CLOP
0255	22 6804	..COPD: SHLD	EDPO
0258	ED53 6806	SDED	EDPN
025C	ED53 64A7	SDED	POINTER
0260	C9	RET	

```

; ROUTINE TO COPY OLD TO NEW UNTIL NULL HIT
; HL=OLD, DE=NEW
0261' 7E COPON: MOV A,M
0262' 12 STAX D
0263' A7 ANA A
0264' C8 RZ
0265' 13 INX D
0266' 23 INX H
0267' 18F8 JMPR COPON
; ROUTINE TO UPDATE THE SCREEN TO REFLECT AFFECTED CHANGE
E
0269' SUPDAT:
0269' CD 0000:0E CALL GETCX
026C' ED53 6814 SDED CURCX
0270' CD 0000:0F CALL GETCY
0273' ED53 6816 SDED CURCY
0277' CD 036F' CALL PNCXY
027A' CD 035E' CALL POCXY
027D' CD 02B6' CALL RETYPE
0280' ED5B 6814 LDED CURCX
0284' CD 0000:18 CALL SETCX
0287' ED5B 6816 LDED CURCY
028B' CD 0000:19 CALL SETCY
; AND FALL INTO ...
; MAKE NEW STRING THE OFFICIAL STRING
028E' OFFICIAL:
; SHITCAN THE OLD STUFF
028E' 2A 681A LHL D EDSTR
0291' CD 0000:17 CALL SDEL
; SET LENGTH
0294' 2A 6802 LHL D NSADDR
0297' E5 PUSH H
0298' 11 0006 LXI D, $SLEN
029B' 19 DAD D
029C' ED5B 6800 LDED STRSIZ
02A0' 73 MOV M, E
02A1' 23 INX H
02A2' 72 MOV M, D
02A3' D1 POP D
; UNCHAINIT
02A4' CD 0000:10 CALL INCUSE
; MAKE NAME POINT AT IT
02A7' 2A 64A1 LHL D EDNAME
02AA' 01 0005 LXI B, $NVALUE
02AD' 09 DAD B
02AE' ED53 681A SDED EDSTR
02B2' 73 MOV M, E
02B3' 23 INX H
02B4' 72 MOV M, D
02B5' C9 RET
; NEW SCREEN RETYPER
; THIS ROUTINE UPDATES THE TEXT DISPLAYED ON THE SCREEN
; TO SHOW THE LAST CHANGE
02B6' CD 0391' RETYPE: CALL GNCXY
02B9' 2A 6806 LHL D EDPN

```



```

02BC' 7E          ..RT1:  MOV    A,M
02BD' A7          ANA    A
02BE' 2021        JRNZ   ..NNUL
; WE HIT A NULL - RESET BOTTOM
02C0' 2A 6810    LHLD  NEWBOT
02C3' 22 64A9    SHLD  BOTTOM
02C6' 3A 6812    LDA   EDLONG
02C9' 47          MOV   B,A
02CA' 3E0A        ..TNL:  MVI   A,NL
02CC' CD 05A8'   CALL  PRETYP
02CF' DDCB0266   BIT   SCNEED,C.ST(X)
02D3' 2029        JRNZ   ..FILB      ; YEP - GO DO FILLING
02D5' CD 0391'   CALL  GNCXY
02D8' CD 05CC'   CALL  POSTY1
02DB' CD 036F'   CALL  PNCXY
02DE' 10EA        DJNZ  ..TNL
02E0' C9          RET
; NEW NOT NULL - TRY IT
02E1' CD 05A8'   ..NNUL: CALL  PRETYP
02E4' 23          INX   H
02E5' 200B        JRNZ   ..YESNL
02E7' CD 0391'   CALL  GNCXY
02EA' CD 05CC'   CALL  POSTY1
02ED' CD 036F'   CALL  PNCXY
02F0' 18CA        JMPR  ..RT1
; WE GOT A NEW LINE - ARE WE DONE?
02F2' DDCB0266   ..YESN: BIT   SCNEED,C.ST(X)
02F6' 2814        JRZ   ..NOSC
; YES WE DID!
02F8' 2A 6810    LHLD  NEWBOT
02FB' 22 64A9    SHLD  BOTTOM
; FILL REST OF LINE FULL OF BLANKS
02FE' CD 0391'   ..FILB: CALL  GNCXY
0301' 3E20        ..FILS: MVI   A,' '
0303' CD 05A8'   CALL  PRETYP
0306' C0          RNZ
0307' CD 05C9'   CALL  POSTYP
030A' 18F5        JMPR  ..FILS
; NEWLINE BUT NO SCROLLING - TYPE THE NEWLINE FOR REAL
030C' CD 0391'   ..NOSC: CALL  GNCXY
030F' CD 05CC'   CALL  POSTY1
0312' 22 6806    SHLD  EDPN
0315' 2B          DCX  H
0316' CD 03A2'   CALL  NBUPD
0319' CD 036F'   CALL  PNCXY
; NOW PRETYPE THE OLD STRING UNTIL NULL OR NEWLINE
031C' CD 0380'   CALL  GOCXY
031F' 2A 6804    LHLD  EDPO
0322' 7E          ..OLDL: MOV   A,M
0323' A7          ANA   A
0324' 2806        JRZ   ..DONQ
0326' CD 05A8'   CALL  PRETYP
0329' 23          INX  H
032A' 28F6        JRZ   ..OLDL
032C' 22 6804    ..DONQ: SHLD  EDPO

```

```

032F'   CD 035E'           CALL   POCXY
                                ; ARE POSITIONS THE SAME?
0332'   2A 680C           LHL   EDNCX
0335'   ED5B 6808         LDEL  EDOCX
0339'   CD 0000:08       CALL  CPHLDE
033C'   C2 02B6'         JNZ   RETYPE
033F'   2A 680E           LHL   EDNCY
0342'   ED5B 680A         LDEL  EDOCY
0346'   CD 0000:08       CALL  CPHLDE
0349'   C2 02B6'         JNZ   RETYPE
                                ; NOW THE FINAL CHECK - DO STRINGS EQUAL EACH OTHER HERE
                                ON OUT?
034C'   2A 6806           LHL   EDPN
034F'   ED5B 6804         LDEL  EDPO
0353'   1A                ..FCHK: LDAX  D
0354'   BE                CMP   M
0355'   C2 02B6'         JNZ   RETYPE
0358'   13                INX  D
0359'   23                INX  H
035A'   A7                ANA  A
035B'   20F6              JRNZ ..FCHK
035D'   C9                RET

                                ; PUT OCXY
035E'   D5                POCXY: PUSH  D
035F'   CD 0000:0E       CALL  GETCX
0362'   ED53 6808         SDEL  EDOCX
0366'   CD 0000:0F       CALL  GETCY
0369'   ED53 680A         SDEL  EDOCY
036D'   D1                POP   D
036E'   C9                RET

                                ; PUT NCXY
036F'   D5                PNCXY: PUSH  D
0370'   CD 0000:0E       CALL  GETCX
0373'   ED53 680C         SDEL  EDNCX
0377'   CD 0000:0F       CALL  GETCY
037A'   ED53 680E         SDEL  EDNCY
037E'   D1                POP   D
037F'   C9                RET

                                ; GET OCXY
0380'   D5                GOCXY: PUSH  D
0381'   ED5B 6808         LDEL  EDOCX
0385'   CD 0000:18       CALL  SETCX
0388'   ED5B 680A         LDEL  EDOCY
038C'   CD 0000:19       CALL  SETCY
038F'   D1                POP   D
0390'   C9                RET

                                ; GET NCXY
0391'   D5                GNCXY: PUSH  D
0392'   ED5B 680C         LDEL  EDNCX
0396'   CD 0000:18       CALL  SETCX
0399'   ED5B 680E         LDEL  EDNCY
039D'   CD 0000:19       CALL  SETCY
03A0'   D1                POP   D
03A1'   C9                RET

                                ; SUBROUTINE TO UPDATE BOTTOM CANDIDATE AS PER CHAR TYPE

```

```

D
03A2' 7E          NBUPD:  MOV    A,M
03A3' FE0A        CPI     NL
03A5' 2001        JRNZ   ..NONL
03A7' 23          INX    H
03A8' 22 6810    ..NONL: SHLD  NEWBOT
03AB' C9          RET

; SUBROUTINE TO RETYPE THE ENTIRE SCREEN
; START WITH TOP AND TYPE AWAY UNTIL NULL HIT OR
; WE GO OFF THE BOTTOM.  REMEMBER THE NEWLINE PLACE
; JUST BEFORE AS BOTTOM
REPAINT:
03AC'           CALL   RESCX
03AC' CD 0000:14  LHLD  TOP
03AF' 2A 6818    MVI   A,CNTRL ; CLEAR AWAY THE SCREEN
03B2' 3E0C        CALL  POSTY1
03B4' CD 05CC'   MOV   B,H ; INIT NL POINT MEMORIZO
03B7' 44

R
03B8' 4D          MOV   C,L
03B9' 7E          ..LOOP: MOV  A,M ; A NULL?
03BA' A7          ANA   A
03BB' 2810        JRZ   ..NULL ; YEP - KICKOUT
03BD' CD 0000:0F CALL  GETCY
03C0' D5          PUSH  D
03C1' CD 05A8'   CALL  PRETYP ; PRETEND TO TYPE
03C4' 2813        JRZ   ..NONL ; JUMP IF NOT NEWLINE
03C6' DDCB0266   BIT   SCNEED,C.ST(X) ; DID WE SCROLL TOO?
03CA' 2806        JRZ   ..NL ; NO
03CC' D1          POP   D
03CD' ED43 64A9  ..NULL: SBCD  BOTTOM ; YES - WE IS DONE
03D1' C9          RET ; BYE BYE
03D2' 44          ..NL:  MOV  B,H ; REMEMBERIZE NEW NL POI
NT
03D3' 4D          MOV   C,L
03D4' FE0A        CPI     NL ; WAS THAT A NEW LINE?
03D6' 2001        JRNZ   ..NONL ; NO
03D8' 03          INX    B ; YES - FUZZ TO AFTER
03D9' CD 0000:18 ..NONL: CALL  SETCX
03DC' D1          POP   D
03DD' CD 0000:19 CALL  SETCY
03E0' CD 05CC'   CALL  POSTY1
03E3' 23          INX    H
03E4' 18D3        JMPR  ..LOOP

; SUBROUTINE TO ADVANCE THE CURSOR FORWARD A CHARACTER
FORWC:
03E6' 2A 64A7    LHLD  POINTER
03E9' 7E          MOV   A,M
03EA' A7          ANA   A
03EB' C8          RZ
03EC' CD 0000:0F CALL  GETCY ; SAVE CY
03EF' D5          PUSH  D
03F0' CD 05A8'   CALL  PRETYP
03F3' D1          POP   D
03F4' CD 0000:19 CALL  SETCY ; AND REST
03F7' 2005        JRNZ  FORWL ; IF FUNNY STUFF ADVANCE
  
```

```

03F9' 23          INX      H
03FA' 22 64A7    SHLD    POINTER
03FD' C9          RET
; ELSE FALL INTO...
;
; SUBROUTINE TO MOVE FORWARD A LINE
FORWL:
03FE' 2A 64A7    LHL    POINTER      ; IF ON NULL - FORGET IT
!
0401' 7E          MOV     A,M
0402' A7          ANA     A
0403' C8          RZ
0404' CD 0000:0E CALL    GETCX
0407' D5          PUSH   D
0408' 2A 64A5    LHL    CURRENT
040B' E5          PUSH   H
040C' CD 0000:0F CALL    GETCY
040F' D5          PUSH   D
0410' CD 0567'   CALL    FFNLP      ; FIND FOLLOWING NL
0413' D1          POP     D      ; DE=CY
0414' E3          XTHL           ; HL=OLD CURRENT
0415' CD 0000:19 CALL    SETCY
0418' D5          PUSH   D
0419' ED5B 64A9  LD     BOTTOM      ; ARE WE AT THE BOTTOM O
F THE FILE
041D' CD 0000:08 CALL    CPHLDE
0420' 280F       JRZ     ..ATBO
; WE ARE NOT YET AT THE BOTTOM
; WE THINK!!
0422' 3E0A       MVI     A,NL      ; MOVE CY DOWN 1 LINE
0424' CD 05A8'   CALL    PRETYP
0427' DDCB0266   BIT     SCNEED,C.ST(X)
042B' 2004       JRNZ    ..ATBO
042D' D1          POP     D
042E' E1          POP     H
042F' 1833       JMPR   ..JOIN
; WE ARE AT THE BOTTOM
0431' D1          ..ATBO: POP    D
0432' CD 0000:19 CALL    SETCY
0435' E1          POP     H
0436' 7E          MOV     A,M      ; NULL CHECK
0437' A7          ANA     A
0438' 200C       JRNZ    ..OK
043A' 2B          DCX     H      ; CHECK FOR NEW LINE
043B' 7E          MOV     A,M
043C' 23          INX     H
043D' FE0A       CPI     NL
043F' 2805       JRZ     ..OK
0441' D1          ..NOGO: POP    D      ; THERE AINT NOWHERE
0442' CD 0000:18 CALL    SETCX
0445' C9          RET      ; TO GO
; THANGS ARE OK
0446' 3E0A       ..OK:  MVI     A,NL      ; TYPE A REAL LIVE NEWLI
NE
0448' CD 05CC'   CALL    POSTY1

```

```

044B' E5          PUSH    H
044C' CD 0000:0F  CALL    GETCY
044F' D5          PUSH    D
0450' CD 0578'    CALL    THLUNL          ; THEN TYPE OUT THE NEXT
                                LINE
0453' 2A 6818    LHL    TOP          ; RESET TOP POINTER
0456' CD 0567'    CALL    FFNLP
0459' 22 6818    SHLD   TOP
045C' D1          ..NOSH: POP    D
045D' CD 0000:19 CALL    SETCY
0460' E1          POP    H
0461' 22 64A9    SHLD   BOTTOM        ; CHANGO THE BOTTOM POIN
                                TER TOO
0464' 22 64A5    ..JOIN: SHLD   CURRENT      ; RESET THE CURRENT LINE
0467' 44          MOV    B,H          ; BC=NEW CURRENT LINE
0468' 4D          MOV    C,L
0469' E1          POP    H          ; HL=TARGET CX
046A' CD 0000:0F  CALL    GETCY
046D' D5          PUSH    D
046E' CD 0585'    CALL    HORICX
0471' ED43 64A7  SBCD   POINTER
0475' D1          POP    D
0476' CD 0000:19 CALL    SETCY
0479' C9          RET
                                ; BACKUP A CHARACTER
047A' 2A 64A7    BACKC: LHL    POINTER
047D' ED5B 64A3  LDED   FIRST
0481' CD 0000:08  CALL    CPHLDE
0484' C8          RZ          ; KICKOUT IF AT BEGIN OF
                                FILE
0485' 2B          DCX    H
0486' 22 64A7    SHLD   POINTER
0489' CD 0000:0F  CALL    GETCY
048C' D5          PUSH    D
048D' 3E7F        MVI    A,RUBOUT      ; PRETEND TO TYPE A RUBO
                                UT
048F' CD 05A8'    CALL    PRETYP
0492' D1          POP    D
0493' CD 0000:19  CALL    SETCY
0496' DDCB026E    BIT    RUBACK,C,ST(X)
049A' C8          RZ          ; KICKOUT IF NOTHING FUN
                                NY
                                ; ELSE FALL INTO...
                                ; BACKUP A LINE
                                BACKL:
049B' 2A 64A5    LHL    CURRENT      ; IF ALREADY AT TOP
049E' ED5B 64A3  LDED   FIRST        ; IGNORE
04A2' CD 0000:08  CALL    CPHLDE
04A5' C8          RZ
04A6' CD 0000:0E  CALL    GETCX
04A9' D5          PUSH    D
04AA' CD 0000:0F  CALL    GETCY
04AD' D5          PUSH    D
04AE' E5          PUSH    H          ; SAVE CURRENT

```

```

04AF'   CD 0526'           CALL   FPNLP           ; SET BC AT PREV NL
; DID WE BACK UP PAST FIRST LINE ON SCREEN?
04B2'   E1                POP    H                ; HL=OLD CURRENT LINE
04B3'   ED5B 6818         LDED   TOP
04B7'   CD 0000:08       CALL   CPHLDE
04BA'   202D             JRNZ   ..NOSC           ; NO
; WE DID IT!? SCROLL DOWN A LINE
04BC'   C5                PUSH   B
04BD'   FDE5             PUSH   Y
04BF'   FD21 67AD        LXI    Y,TMPARG
04C3'   DD7E10           MOV    A,C.YF(X)
04C6'   2F                CMA
04C7'   3C                INR    A
04C8'   CD 0000:06       CALL   BLDBOX
04CB'   CD 0000:16       CALL   SCROLE
04CE'   CD 0000:14       CALL   RESCX
04D1'   FDE1             POP    Y
04D3'   E1                POP    H                ; HL=PREV NL POINT
04D4'   E5                PUSH   H
04D5'   CD 0578'         CALL   THLUNL          ; TYPE FROM THEIR ON
; RESET CURRENT AND TOP POINTERS
04D8'   E1                POP    H
04D9'   22 6818          SHLD  TOP
04DC'   22 64A5          SHLD  CURRENT
04DF'   CD 0507'         CALL   RESBOT
04E2'   ED43 64A9        SBCD  BOTTOM
04E6'   D1                POP    D                ; DE=CY
04E7'   180C             JMPR  ..SPOT
; WE DID NOT BACKUP BEFORE FILE START
04E9'   ED43 64A5        ..NOSC: SBCD  CURRENT   ; CHANGE CURRENT
04ED'   E1                POP    H                ; HL=NEW CY
04EE'   DD5E10           MOV    E,C.YF(X)
04F1'   1600             MVI   D,0
04F3'   19                DAD   D
04F4'   EB                XCHG
; COMPUTE NEW CURSOR POSITION AND RESET
04F5'   E1                ..SPOT: POP    H        ; HL=TARGET CX
04F6'   D5                PUSH   D
04F7'   ED4B 64A5        LBCD  CURRENT
04FB'   CD 0585'         CALL   HORICX
04FE'   ED43 64A7        SBCD  POINTER
0502'   D1                POP    D
0503'   CD 0000:19       CALL   SETCY
0506'   C9                RET
; IN HL=TOP
; OUT BC=BOTTOM
0507'   CD 0000:14       RESBOT: CALL   RESCX
050A'   44                ..PUZ1: MOV    B,H
050B'   4D                MOV    C,L
050C'   7E                ..PUZ2: MOV    A,M
050D'   A7                ANA   A
050E'   C8                RZ
050F'   CD 05A7'         CALL   PRETHL
0512'   280F             JRZ   ..FUZ3
0514'   DDCB0266        BIT   SCNEED,C.ST(X)

```

```

0518'  C0                RNZ
0519'  44                MOV     B,H
051A'  4D                MOV     C,L
051B'  7E                MOV     A,M
051C'  23                INX     H
051D'  FE0A             CPI     NL
051F'  28E9             JRZ     ..PUZ1
0521'  18E9             JMPR   ..PUZ2
0523'  23                ..PUZ3: INX   H
0524'  18E6             JMPR   ..PUZ2
                                ; SUBROUTINE TO FIND PREVIOUS NEWLINE POINT
                                ; IN: HL=CURRENT LINE POINTER
                                ; OUT: BC=PREVIOUS NEWLINE POINTER
                                ; EVERYTHING ELSE IS FUBAR
0526'  ED5B 64A3       FPNLP: LDED   FIRST      ; DE=FIRST CHAR IN STRIN
                                G
052A'  2B                DCX   H
052B'  44                MOV   B,H
052C'  4D                MOV   C,L
052D'  7E                MOV   A,M
052E'  FE0A             CPI   NL
0530'  2008             JRNZ  ..FNLO
0532'  CD 0000:08      CALL  CPHLDE
0535'  2004             JRNZ  ..FNL1
0537'  D5                PUSH  D
0538'  1812             JMPR  ..BONG
053A'  23                ..FNLO: INX  H
053B'  E5                ..FNL1: PUSH H      ; SAVE FORWARD SCAN LIMIT
                                T
053C'  2B                ..FNL:  DCX  H      ; LOOK BACKWARDS FOR A N
                                EWLIN
053D'  CD 0000:08      CALL  CPHLDE
0540'  7E                MOV   A,M
0541'  2004             JRNZ  ..BUFO
0543'  FE0A             CPI   NL
0545'  2005             JRNZ  ..BONG
0547'  FE0A             ..BUFO: CPI  NL
0549'  20F1             JRNZ  ..FNL
054B'  23                ..BANG: INX  H
054C'  44                ..BONG: MOV  B,H      ; REMEMBER WHERE NL IS
054D'  4D                MOV   C,L
054E'  CD 0000:13      CALL  RESCX
                                ; PRETEND TYPE FROM START OR PREV NEWLINE
                                ; UP UNTIL WE HIT THE SCANNING LIMIT WHERE WE CAME
                                ; FROM - THE PLACE WE WRAPPED AROUND OR STARTED FROM LAS
                                T
                                ; IS IT
0551'  7E                ..BON1: MOV  A,M
0552'  A7                ANA   A      ; NULL CHECK
0553'  2002             JRNZ  ..BON2
0555'  E1                POP   H
0556'  C9                RET
0557'  CD 05A7'       ..BON2: CALL  PRETHL
055A'  2808             JRZ   ..BON3      ; JUMP IF NOTHIN FUNNY
055C'  D1                POP   D      ; DE=LIMIT
  
```

```

055D'  CD 0000:08          CALL    CPHLDE          ; IS WE AT IT?
0560'  C8                  RZ                      ; WE GOT IT
0561'  D5                  PUSH    D                      ; ELSE
0562'  44                  MOV     B,H              ; REMEMBER THIS WRAPAROU

                                ND
0563'  4D                  MOV     C,L
0564'  23                  ..BON3: INX    H
0565'  18EA               JMPR    ..BON1
                                ; SUBROUTINE TO FIND FOLLOWING NEWLINE POINT
                                ; IN HL=CURRENT
                                ; OUT HL=FOLLOWING
0567'  CD 0000:13        FFNLP: CALL    RESCX
056A'  7E                  ..NCHK: MOV    A,M
056B'  A7                  ANA    A
056C'  C8                  RZ                      ; KICKOUT ON NULL
056D'  23                  INX    H
056E'  FE0A               CPI    NL              ; OR NL
0570'  C8                  RZ
0571'  CD 05A8'          CALL    PRETYP
0574'  28F4               JRZ    ..NCHK
0576'  2B                  DCX    H              ; CHAR WRAP AROUND - SO

                                FIX
0577'  C9                  RET
                                ; TYPEOUT STUFF UNTIL A NEWLINE WOULD BE TYPED
                                ; HL=ADDRESS TO TYPE FROM
0578'  7E                  THLUNL: MOV   A,M
0579'  A7                  ANA    A
057A'  C8                  RZ
057B'  CD 05A8'          CALL    PRETYP          ; PRE-TYPE IT
057E'  C0                  RNZ          ; QUIT IF WOULD BE NL
057F'  CD 05C9'          CALL    POSTYP         ; ELSE REALLY DO IT
0582'  23                  INX    H
0583'  18F3               JMPR    THLUNL
                                ; HORIZONTAL INDEX CX FROM THE CURRENT LINE POINTER
                                ;
                                ; WE PRETEND TO TYPE FROM THE LEFT HAND SIDE UNTIL WE
                                ; EITHER GET A CX > TARGET CX OR WE FORCE A NEWLINE
                                ; THE PREVIOUS CHARACTER BECOMES 'IT'
0585'  CD 0000:13        HORICX: CALL   RESCX
0588'  0A                  ..HORL: LDAX  B          ; GOOD OLD NULL CHECK
0589'  A7                  ANA    A
058A'  C8                  RZ
058B'  E5                  PUSH   H
058C'  CD 05A8'          CALL    PRETYP
058F'  DD6E0B             MOV    L,C,CX(X)       ; TOP=TARGET
0592'  DD660C             MOV    H,C,CX+1(X)    ; HL=NOW
0595'  200B               JRNZ   ..HOR1
0597'  EB                  XCHG
0598'  E3                  XTHL          ; DE=NOW,HL=TARGET, TOP=T

                                HEN
0599'  CD 0000:08          CALL    CPHLDE          ; IS NOW > TARGET?
059C'  D1                  POP    D
059D'  3804               JRC    ..HOR2
059F'  03                  INX    B
05A0'  18E6               JMPR   ..HORL

```



```

; DE=THEN
05A2' E1      ..HOR1: POP      H
05A3' CD 0000:18  ..HOR2: CALL    SETCX
05A6' C9      RET

; PRETEND TO TYPE THE CHARACTER POINTED AT BY HL
05A7' 7E      PRETHL: MOV     A,M
; PRETEND TO TYPE A CHARACTER IN A
05A8' CD 0000:0E  PRETYP: CALL   GETCX
05AB' D5      PUSH     D
05AC' DD5E0D    MOV     E,C.CY(X)
05AF' DDCB02A6  RES     SCNEED,C.ST(X)
05B3' DDCB02AE  RES     RUBACK,C.ST(X)
05B7' A7      ANA     A
05B8' CD 0000:09  CALL    DCHAR
05BB' DDCB0266  BIT     SCNEED,C.ST(X)
05BF' 2006     JRNZ    ..PRE1
05C1' 57      MOV     D,A
05C2' DD7E0D    MOV     A,C.CY(X)
05C5' BB      CMP     E
05C6' 7A      MOV     A,D
05C7' D1      ..PRE1: POP    D
05C8' C9      RET

; TYPE PREVIOUSLY TYPED CHARACTER AT SAME PLACE
05C9' CD 0000:18  POSTYP: CALL   SETCX
05CC' 37      POSTY1: STC
05CD' C3 0000:09  JMP     DCHAR
      .END

```

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

AASN	005F	ALSTS	0000:04 X	ARG	0000:05 X	ARGSTK	60CC
BACDEL	0129'	BACKC	047A'	BACKGR	65CC	BACKL	049B'
BLANK	0020	BLDBOX	0000:06 X	BOTRAM	6000	BOTTOM	64A9
CHARSL	65DD	CLEARC	60CA	CNTCK	0000:07 X	CNTRL	000C
CNTRLC	65CD	CNTRLO	65D9	CNTRLU	0015	CNTRLZ	65E4
COPON	0261'	CPHLDE	0000:08 X	CPLARE	611C	CPLSIZ	0140
CR	000D	CSBLOK	668D	CSFLAG	6260	CURBAC	0101'
CURCX	6814	CURCY	6816	CURDW1	011C'	CURDWN	0119'
CURF01	010C'	CURFOR	0109'	CURREN	64A5	CURUP	0111'
C.CO	0011	C.C1	0012	C.CX	000B	C.CY	000D
C.DP	0013	C.ST	0002	C.X	0003	C.XF	000F
C.XS	0007	C.Y	0005	C.YF	0010	C.YS	0009
DCHAR	0000:09 X	DDTON	65CE	DELCHR	012E'	DELLIN	018F'
DEVBL	6589	DEVCL0	6579	DEVCL1	657B	DEVCL2	657D
DEVCL3	657F	DEVCL4	6581	DEVCL5	6583	DEVCL6	6585
DEVCL7	6587	DEVFB	65CB	DEVHCB	658F	DEVMO	658B
DEVNM	65B7	DEVNT	658D	DEVTNA	65BB	DEVTNB	65BF
DEVTNC	65C3	DEVVA	65BD	DEVVAR	6579	DEVVB	65C1
DEVVBL	6591	DEVVC	65C5	DEVVD	65C9	DEVVN	65B9
DEVVS	65C7	DEVXCD	65B3	DEVYCD	65B5	DIV16	0000:0A X
DOLPLH	62E8	DOLPPT	62EA	DUMBST	6577	EDBCNT	64AD
EDCNT	64A7	EDCLIT	00C7'	EDCUNT	00E3'	EDDISP	008D'
EDIT	0000' I	EDLONG	6812	EDMODE	681C	EDNAME	64A1
EDNCX	680C	EDNCY	680E	EDNEWS	64A5	EDOCX	6808
EDOCY	680A	EDPN	6806	EDPO	6804	EDPTRC	64AB
EDPTRL	64A9	EDSTR	681A	ENDINS	0128'	ERABIT	0002
ERRPGM	0000:0B X	ER.ARA	002F	ER.ARG	0034	ER.ASN	0015
ER.BOX	001A	ER.CHN	0002	ER.CMD	001F	ER.CNV	0016
ER.COR	001B	ER.CTL	0036	ER.DEL	0026	ER.DIM	0030
ER.DIV	0018	ER.DP	0037	ER.DSK	0019	ER.EDT	0035
ER.FMT	0038	ER.FNF	001C	ER.FOR	0028	ER.IMP	0003
ER.LAB	0025	ER.MAC	0022	ER.NAE	002D	ER.NAM	0029
ER.NEG	003A	ER.NOT	0023	ER.NUL	0039	ER.NUM	002B
ER.NXT	001D	ER.OFL	0017	ER.OPN	0014	ER.OVE	001E
ER.PAR	002A	ER.REN	002C	ER.RET	0024	ER.SEP	0021
ER.SNP	0031	ER.SPC	002E	ER.STK	0004	ER.SW	0032
ER.TER	0020	ER.UFL	0033	ER.UNF	0027	ESCAPE	001B
EXP	0000:0C X	EXTDEL	002E	E.HVAL	0002	E.LVAL	0001
E.SIZ	0005	E.TYP	0000	E.VAL	0001	FCNTH	65D2
FCNTI	65D3	FCNTJ	65D4	FCNTK	65D5	FCNTL	65D6
FCNTV	65E0	FCNTY	65E3	FFNLP	0567'	FIRST	64A3
FLAGS	65CB	FOREGR	65D0	FORWC	03E6'	FORWL	03FE'
FPNLP	0526'	FRAGSI	0400	FREELS	65E5	FSTDOL	648C
FSTINT	62EC	FWDPTR	65E7	GETCH	0000:0D X	GETCX	0000:0E X
GETCY	0000:0F X	GNCXY	0391'	GNSTR	01E8'	GOCXY	0380'
HCAREA	6593	HORICX	0585'	INCRO	65E9	INCUSE	0000:10 X
INSERT	0123'	INSLIN	01CE'	INSREP	014D'	JUNK	6542
KBLOCK	65F1	KEYFLG	67FF	KEYPTK	6533	KEYTRK	67F7
LF	000A	LISTON	65E2	MACSTU	6536	MACTOP	625E
MAXFRG	0040	MNMX	65EB	NBLKB	0000	NBLKM	0001
NBUPD	03A2'	NEGBC	01C6'	NEWBOT	6810	NL	000A
NSADDR	6802	NUMBUF	64A3	NXTVAL	0000:11 X	OFFICI	028E'
OLDCHR	64A0	OLDCUR	649F	OLDKEY	649D	OLDXY	65EF
ONEBUF	6729	OPRL	0014	OPRSP	655B	OPRSTK	6547
OPRSZ	655D	OUTOFF	62E7	PCNT	655E	PCRET	011F'

EDITOR -

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

PIXVAL 65ED	PNCXY 036F'	POCXY 035E'	POINTE 64A7
PONOFF 65DA	POSTY1 05CC'	POSTYP 05C9'	PRETHL 05A7'
PRETYP 05A8'	PRINTR 6540	PUTCUR 0000:12 X	RAMEND 7FFF
RAMSTR 6900	RANSHT 655F	REPAIR 03AC'	RESBOT 0507'
RESCX 0000:13 X	RESCXY 0000:14 X	RESCY 0000:15 X	RETYPE 02B6'
RMDTMP 6538	RUBACK 0005	RUBOUT 007F	SAVESP 625C
SCNEED 0004	SCROLE 0000:16 X	SCRWIN 67CD	SDEL 0000:17 X
SETCX 0000:18 X	SETCY 0000:19 X	SOPRSP 653D	SOPRSZ 653F
STACK 60C8	STAKTO 6000	STRSIZ 6800	SUBSTU 6534
SUPDAT 0269'	TAB 0009	TAPBUF 64B3	TAPCON 64AF
TAPPRO 64B1	TBFEND 6533	TEMPHD 60CA	TEMPS 62E5
THLUNL 0578'	TMPARG 67AD	TOP 6818	TTYBEG 6261
TTYEND 62E1	TTYINT 62E3	TTYPTR 62E1	TXTWIN 67E2
UARTFL 649C	USREND 65F1	V3PTR 6573	VDCHAR 649E
VDNLF 65CF	VIPLH 6545	VOICE0 6563	WRMODE 65EE
ZGIM2 0001	ZGREND 681D	\$AADR 0010	\$ADDRF 0007
\$ADDRI 0005	\$ADDRS 0009	\$ANY 001A	\$ANYNA FFFC
\$ANYVA FFFE	\$ARGPT 0011	\$BGPTR 000F	\$BNDL 0007
\$CALLE 000D	\$CMDAD 0018	\$CPLBL 002A	\$CSBLD 0028
\$DATAP 0007	\$DOLDE 0001	\$DOL00 0002	\$DVAL 0000
\$END 001C	\$FADR 000E	\$FLAGS 0002	\$FORBL 0024
\$FORPT 000B	\$FVAL 0006	\$GOSUB 001A	\$IADR 000C
\$INPBU 0018	\$INPPT 0016	\$IVAL 0004	\$KEYBL 0026
\$LENGT 0001	\$LINPT 0005	\$LOCPT 0009	\$MIBBL 0022
\$MIBEN 001B	\$NAMAD 000A	\$NAME 000A	\$NASCI 0009
\$NDEL 0080	\$NLINK 0003	\$NULL 0002	\$NVALU 0005
\$REPEA 001E	\$RVSTU 0013	\$SAME 0020	\$SASCI 000A
\$SLEN 0006	\$STRAD 0008	\$STRPT 0003	\$TAF 0000
\$TYPE 0000	\$USE 0005	.BLNK. 0000:03 X	.DATA. 0000* X
.PROG. 05D0' X			


```

0020' 47          MOV      B,A
0021' 1A          LDAX    D          ; DIFERENT FROM LAST?
0022' A8          XRA      B
0023' C8          RZ          ; NO
0024' 78          MOV      A,B
0025' 12          STAX    D
0026' C9          RET

; EQUATES:
0080          SLBITM=80H      ; SHIFT LOCK BIT IN KEYFLG AND LEDS
0040          MDBITM=40H     ; MODE BIT SAME STORY
0003          CK1RAM=3       ; LOCATION OF CONTROL KEYS
0003          CK1BIT=3
0007          CK2RAM=7
0004          CK2BIT=4
0000          SK1RAM=0       ; LOCATION OF SHIFT KEYS
0000          SK1BIT=0
0007          SK2RAM=7
0002          SK2BIT=2
0000          TOKRAM=0       ; LOCATION OF TOKEN KEY
0002          TOKBIT=2
0001          SHYLOK=1      ; KEYCODE NUMBER FOR SHIFT LOCK KEY
0004          ESCKEY=4      ; KEYCODE NUMBER OF ESCAPE KEY
;
; KEYBOARD SCANNER ROUTINE
; IF A KEY HAS BEEN PRESSED, CY IS SET AND THE CHARACTER
; IS RETURNED
; IN A.
0027' 01 FE2C    KEYSCHN: LXI    B,0FE2CH          ; B=COL SEL BIT,
; C=PORT #
002A' 11 0000    LXI      D,0          ; DE=COLUMN #
002D' ED41      ..SCAN: OUTP   B
002F' ED78      INP      A          ; GET ROW DATA
0031' 2F        CMA          ; MAKE RIGHT SIDE UP
0032' 21 67F7   LXI      H,KEYTRK      ; UPDATE TRACKING MEMORY

0035' 19        DAD      D
0036' 77        MOV      M,A
0037' 21 00CC'   LXI      H,KEYMES          ; DO WE HAVE A LIVE ONE?

003A' 19        DAD      D
003B' A6        ANA      M
003C' 200A     JRNZ    ..LIVE          ; JUMP OUT IF SO
003E' 1C        INR      E          ; NO - BUMP COL
003F' CB00     RLCR      B          ; SHOVE OVER THE MASK
0041' 38EA     JRC      ..SCAN      ; LOOP TILL FALLS OFF EN
;
; D
0043' AF        XRA      A          ; FAILURE-NAIL OLDKEY ME
; M
0044' 32 649D   STA      OLDKEY
0047' C9        RET          ; HOME TO MAMMA
; A KEY IS DOWN - CONVERT TO INTERMEDIATE KEYCODE
; FIRST CONVERT TO BIT NUMBER
0048' 0F        ..LIVE: RRC
0049' 3803     JRC      ..BITF
004B' 14        INR      D
    
```

```

004C' 18FA          JMPR    ..LIVE
004E' 7B          ..BITF: MOV    A,E      ; COLUMN #
004F' A7          ANA     A
0050' 07          RLC
0051' 07          RLC
0052' 07          RLC
0053' 82          ADD     D      ; + BIT #
0054' 4F          MOV    C,A      ; KEYCODE TO C
; IS THE KEYCODE THE SAME AS THAT FOUND ON PREVIOUS SCAN
??

0055' 21 649D     LXI    H,OLDKEY
0058' BE          CMP    M
0059' C8          RZ      ; QUIT IF SO
005A' 77          MOV    M,A      ; ELSE UPDATE THANGS
;

005B' 21 67FF     LXI    H,KEYFLG ; POINT AT SHIFTLOCK/MOD
E FLAGS
; CHECK FOR SHIFT LOCK KEYPRESS

005E' FE01       CPI    SHYLOK
;
0060' 2005       JMR    ..NOSL
0062' 7E          JRNZ   ..NOSL
0063' EE80       MOV    A,M      ; YEP - TOGGLE SHIFT LOC
K BIT
0065' 1816       XRI    SLBITM
; CHECK FOR CONTROL KEY
0067' 3A 67FA     JMPR   ..ULED ; JUMP TO UPDATE
..NOSL: LDA    KEYTRK+CK1RAM
006A' CB5F       BIT    CK1BIT,A
006C' 2007       JRNZ   ..CKDN
006E' 3A 67FE     LDA    KEYTRK+CK2RAM
0071' CB67       BIT    CK2BIT,A
0073' 2819       JRZ    ..NOCK
; WE GOT A CONTROL KEY - DO WE HAVE ESCAPE AS WELL?
0075' 79          ..CKDN: MOV   A,C
0076' FE04       CPI    ESCKEY
0078' 200F       JRNZ   ..NOES
007A' 7E          MOV    A,M      ; YEP - TOGGLE MODE BIT
007B' EE40       XRI    MDBITM
007D' 77          ..ULED: MOV   M,A      ; SHIFTLOCK JOINS HERE T
00

007E' 3A 6589     LDA    DEVBL
0081' E63F       ANI    3FH
0083' B6          ORA    M
0084' 32 6589     STA    DEVBL
0087' A7          ANA    A
0088' C9          RET
; NO ESCAPE - NORMAL CONTROL KEY
0089' 21 0154'    ..NOES: LXI    H,CKTBL
008C' 181B       JMPR   ..LOOK
; HOW ABOUT SHIFT KEY?
008E' 3A 67F7     ..NOCK: LDA    KEYTRK+SK1RAM
0091' CB47       BIT    SK1BIT,A
0093' 200C       JRNZ   ..SKDN
0095' 3A 67FE     LDA    KEYTRK+SK2RAM
0098' CB57       BIT    SK2BIT,A

```

```

009A' 2005          JRNZ  ..SKDN
009C' 7E           MOV   A,M           ; IS SHIFT LOCKED?
009D' E680        ANI   SLBITM
009F' 2805        JRZ   ..NOSK
; YEP - USE SHIFT LOOKUP TABLE
00A1' 21 0114'    ..SKDN: LXI   H,SKTBL
00A4' 1803        JMPR  ..LOOK
00A6' 21 00D4'    ..NOSK: LXI   H,NORTBL      ; ASSUME NOT
00A9' 0600        ..LOOK: MVI   B,0           ; DO TABLE LOOKUP
00AB' 09          DAD   B
00AC' 7E           MOV   A,M           ; GET ASCII
00AD' 4F           MOV   C,A           ; SAVE CHARACTER
00AE' A7           ANA   A           ; VALID KEY?
00AF' C8          RZ           ; ZERO MEANS NOT SO
; IS UPPER/LOWER ALPHA REVERSE WANTED?
00B0' 3A 67FF    LDA   KEYFLG
00B3' E640        ANI   MDBITM
00B5' 79          MOV   A,C           ; STAGE CHAR FOR WHATEVE
R
00B6' 2812        JRZ   ..NORM
; REVERSE MODE IS SET/IS CHARACTER IN RANGE FOR CONFUSIO
N
00B8' FE41        CPI   'A'
00BA' 380E        JRC   ..NORM           ; SKIP IF < UPPER A
00BC' FE7B        CPI   7BH           ; OR IF ABOVE LOWER Z
00BE' 300A        JRNC  ..NORM
00C0' FE61        CPI   61H           ; COOL IF >= LOWER A
00C2' 3004        JRNC  ..BIZR
00C4' FE5B        CPI   5BH           ; BAD IF ABOVE UPPER Z
00C6' 3002        JRNC  ..NORM
00C8' EE20        ..BIZR: XRI   20H           ; DO REVERSAL
00CA' 37          ..NORM: STC
00CB' C9          RET
; KEYBOARD SCANNER TABLES
; TABLE OF LIVE KEYS - ORDERED AS THE SCAN MATRIX IS
KEYMES:
00CC' FA          .BYTE  11111010B
00CD' DF          .BYTE  11011111B
00CE' FF          .BYTE  11111111B
00CF' F7          .BYTE  11110111B
00D0' FF          .BYTE  11111111B
00D1' FF          .BYTE  11111111B
00D2' FF          .BYTE  11111111B
00D3' E0          .BYTE  11100000B
; CHARACTER LOOKUP TABLES
;
; UNSHIFTED CHARACTERS
; ORDERED BY ROW, THEN COLUMN STARTING WITH BIT ZERO
NORTBL:
00D4' 000000091B51 .BYTE  0,0,0,09H,1BH,'Q','I','1'
00DC' 204247465900 .BYTE  ' ','B','G','F','Y',0,'6','7'
00E4' 564344535254 .BYTE  'V','C','D','S','R','T','4','5'
00EC' 585A41005745 .BYTE  'X','Z','A',0,'W','E','2','3'
00F4' 4E4D4A484955 .BYTE  'N','M','J','H','I','U','8','9'
00FC' 2C2E4C4B504F .BYTE  ',','.',',','L','K','P','O','0','-'
    
```


KEYSCN -

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

AASN	005F	ARGSTK	60CC	AUARTD	002A	AUARTS	002B
BACKGR	65CC	BAUDSE	0023	BLANK	0020	BOTRAM	6000
BOTTOM	64A9	BSAUD	0022	CHARSL	65DD	CK1BIT	0003
CK1RAM	0003	CK2BIT	0004	CK2RAM	0007	CKTBL	0154
CLEAR	60CA	CNTRL	000C	CNTRLC	65CD	CNTRLO	65D9
CNTRLU	0015	CNTRLZ	65E4	CPLARE	611C	CPLSIZ	0140
CR	000D	CSBLOK	668D	CSFLAG	6260	CURCX	6814
CURCY	6816	CURREN	64A5	C.CO	0011	C.C1	0012
C.CX	000B	C.CY	000D	C.DP	0013	C.ST	0002
C.X	0003	C.XF	000F	C.XS	0007	C.Y	0005
C.YF	0010	C.YS	0009	DDTON	65CE	DEVBL	6589
DEVCL0	6579	DEVCL1	657B	DEVCL2	657D	DEVCL3	657F
DEVCL4	6581	DEVCL5	6583	DEVCL6	6585	DEVCL7	6587
DEVFB	65CB	DEVHCB	658F	DEVMO	658B	DEVNM	65B7
DEVNT	658D	DEVTNA	65BB	DEVTNB	65BF	DEVTNC	65C3
DEVVA	65BD	DEVVAR	6579	DEVVB	65C1	DEVVBL	6591
DEVVC	65C5	DEVVD	65C9	DEVVN	65B9	DEVVS	65C7
DEVXCD	65B3	DEVYCD	65B5	DOLPLH	62E8	DOLPPT	62EA
DUMBST	6577	EDBCNT	64AD	EDCNT	64A7	EDLONG	6812
EDMODE	681C	EDNAME	64A1	EDNCX	680C	EDNCY	680E
EDNEWS	64A5	EDOCX	6808	EDOCY	680A	EDPN	6806
EDPO	6804	EDPTRC	64AB	EDPTL	64A9	EDSTR	681A
ERABIT	0002	ER.ARA	002F	ER.ARG	0034	ER.ASN	0015
ER.BOX	001A	ER.CHN	0002	ER.CMD	001F	ER.CNV	0016
ER.COR	001B	ER.CTL	0036	ER.DEL	0026	ER.DIM	0030
ER.DIV	0018	ER.DP	0037	ER.DSK	0019	ER.EDT	0035
ER.FMT	0038	ER.FNF	001C	ER.FOR	0028	ER.IMP	0003
ER.LAB	0025	ER.MAC	0022	ER.NAE	002D	ER.NAM	0029
ER.NEG	003A	ER.NOT	0023	ER.NUL	0039	ER.NUM	002B
ER.NXT	001D	ER.OFL	0017	ER.OPN	0014	ER.OVE	001E
ER.PAR	002A	ER.REN	002C	ER.RET	0024	ER.SEP	0021
ER.SNF	0031	ER.SPC	002E	ER.STK	0004	ER.SW	0032
ER.TER	0020	ER.UFL	0033	ER.UNF	0027	ESCKEY	0004
EXTDEL	002E	E.HVAL	0002	E.LVAL	0001	E.SIZ	0005
E.TYP	0000	E.VAL	0001	FCNTH	65D2	FCNTI	65D3
FCNTJ	65D4	FCNTK	65D5	FCNTL	65D6	FCNTV	65E0
FCNTY	65E3	FIRST	64A3	FLAGS	65CB	FOREGR	65D0
FRAGSI	0400	FREELS	65E5	FSTDOL	648C	FSTINT	62EC
FWDPTR	65E7	HCAREA	6593	HORCB	0009	INCRO	65E9
INFBK	000D	INLIN	000F	INMOD	000E	JUNK	6542
KBLOCK	65F1	KEYFLG	67FF	KEYMES	00CC	KEYPSN	0000
KEYPTK	6533	KEYSCN	0027	KEYTRK	67F7	LEDS	0024
LF	000A	LISTON	65E2	MACSTU	6536	MACTOP	625E
MAGIC	000C	MAXFRG	0040	MDBITM	0040	MNMX	65EB
MPBAV	0016	MPMO	0010	MPNCV	0015	MPNV	0017
MPTNA	0011	MPTNB	0012	MPTNC	0013	MPVIB	0014
NBLKB	0000	NBLKM	0001	NEWBOT	6810	NL	000A
NORML	0020	NORTBL	00D4	NSADDR	6802	NUMBUF	64A3
OLDCHR	64A0	OLDCUR	649F	OLDKEY	649D	OLDXY	65EF
ONEBUF	6729	OPRL	0014	OPRSP	655B	OPRSTK	6547
OPRSZ	655D	OUTOFF	62E7	PCNT	655E	PIXVAL	65ED
POINTE	64A7	PONOFF	65DA	PRINTR	6540	RAMEND	7FFF
RAMSTR	6900	RANSHT	655F	RMDTMP	6538	RUBACK	0005
RUBOUT	007F	SAVESP	625C	SCNEED	0004	SCRWIN	67CD
SHYLOK	0001	SK1BIT	0000	SK1RAM	0000	SK2BIT	0002

I

I

KEYSCN -

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

SK2RAM 0007	SKTBL 0114	SLBITM 0080	SOPRSP 653D
SOPRSZ 653F	STACK 60C8	STAKTD 6000	STRSIZ 6800
SUBSTU 6534	TAB 0009	TAPBUF 64B3	TAPCON 64AF
TAPPRO 64B1	TBFEND 6533	TEMPHD 60CA	TEMPS 62E5
TMPARG 67AD	TOKBIT 0002	TOKRAM 0000	TOP 6818
TTYBEG 6261	TTYEND 62E1	TTYINT 62E3	TTYPTR 62E1
TXTWIN 67E2	UARTD 00E0	UARTFL 649C	UARTS 00E1
USREND 65F1	V3PTR 6573	VLANK 000A	VCTR 0021
VDCHAR 649E	VDNLF 65CF	VIPLH 6545	VOICE0 6563
WRMODE 65EE	XPAND 0019	ZGIM2 0001	ZGREND 681D
\$ADDR 0010	\$ADDRF 0007	\$ADDRI 0005	\$ADDRS 0009
\$ANY 001A	\$ANYNA FFFC	\$ANYVA FFFE	\$ARGPT 0011
\$BGPTR 000F	\$BN DL 0007	\$CALLE 000D	\$CMDAD 0018
\$CPLBL 002A	\$CSBLO 0028	\$DATAP 0007	\$DOLDE 0001
\$DOL00 0002	\$DVAL 0000	\$END 001C	\$FADR 000E
\$FLAGS 0002	\$FORBL 0024	\$FORPT 000B	\$FVAL 0006
\$GOSUB 001A	\$IADR 000C	\$INPBU 0018	\$INPPT 0016
\$IVAL 0004	\$KEYBL 0026	\$LENGT 0001	\$LINPT 0005
\$LOCPT 0009	\$MIBBL 0022	\$MIBEN 001B	\$NAMAD 000A
\$NAME 000A	\$NASCI 0009	\$NDEL 0080	\$NLINK 0003
\$NULL 0002	\$NVALU 0005	\$REPEA 001E	\$RVSTU 0013
\$SAME 0020	\$SASCI 000A	\$SLEN 0006	\$STRAD 0008
\$STRPT 0003	\$TAF 0000	\$TYPE 0000	\$USE 0005
.BLNK. 0000:03 X	.DATA. 0000* X	.PRG. 0194	X

```
.INSERT A:S.ASM
@.REMARK /
@           *
@           * * *
@           ***
@           *****
@           *
@           *
@           *
@           *
@           *
@           *****
@
@           "WHEN YOU CARE ENOUGH TO PROGRAM
@           THE VERY BEST"
@
@           ZGRASS V2.00000000
@BY JAY FENTON, NOLA DONATO, AND TOM DEFANTI
@           (C) 1978
@
@/
```

```
.INSERT A:ZRAM.ASM
.PREL
.LINK
.IDENT CHARS
.INTERN BLDBOX
.INTERN CHARWIN
.INTERN DCHAR
.INTERN DVDE2
.INTERN GETCX
.INTERN GETCY
.INTERN INIVDA
.INTERN INIVDM
.INTERN INIVDR
.INTERN LARGE
.INTERN PUTCUR
.INTERN RESCX
.INTERN RESCY
.INTERN RESCX
.INTERN SETCX
.INTERN SETCY
.INTERN SMALL
.INTERN TEXT
.INTERN ZOUTCH
.EXTERN ARG
.EXTERN COSINE
.EXTERN CPHLDE
.EXTERN DOBOX
.EXTERN ERRPGM
.EXTERN INC5IY
.EXTERN ISNL
.EXTERN KISNL
.EXTERN OUTCH
.EXTERN FNXT
.EXTERN POUTCH
```

```

                                .EXTERN R2A
                                .EXTERN SCROLE
                                .EXTERN SEREE
                                .EXTERN THEN
0001      BX.X=1
0006      BX.Y=6
000B      BX.XS=11
0010      BX.YS=16
0015      BX.MOD=21
000C      MAGIC=0CH
0019      XPAND=19H
                                ; TEXT COMMAND
                                ; VERY SIMILAR TO PRINT
                                M.CMDTEXT,TEXTX,0,TS,TEXTA,TEXTSII
0000'    18      +TEXT:  .BYTE $CMDADR
0001'    04      +      .BYTE (TEXTX-TEXT)/16+1
0002'    00      +      .BYTE 0
0003'    0035'   +      .WORD TS
0005'    0029'   +      .WORD TEXTS
0007'    000F'   +      .WORD TEXTA
0009'    5445585400 +      .ASCIZ /TEXT/
                                +]

000E'    00      .BYTE 0
000F'    63      TEXTA:  .BYTE 99
0010'    582C592C4D4F .ASCIZ /X,Y,MODE,THING,THING,.../
0029'    E5      TEXTS:  PUSH  H
002A'    21 0806   LXI    H,0806H
002D'    22 67F1   SHLD   TXTWIN+C.XF
0030'    21 0111'   LXI    H,DIS57
0033'    1817     JMPR   JOINTS
0035'

                                TEXTX:
                                ; TS (TEXT SMALL ENTRY)
                                M.CMDTETS,TSX,0,THEN,TEXTA,TSSII
0035'    18      +TS:  .BYTE $CMDADR
0036'    06      +      .BYTE (TSX-TS)/16+1
0037'    00      +      .BYTE 0
0038'    0000:12 +      .WORD THEN
003A'    0042'   +      .WORD TSS
003C'    000F'   +      .WORD TEXTA
003E'    545300 +      .ASCIZ /TS/
                                +]

0041'    00      .BYTE 0
0042'    E5      TSS:  PUSH  H
0043'    21 0604   LXI    H,0604H
0046'    22 67F1   SHLD   TXTWIN+C.XF
0049'    21 0435'   LXI    H,DIS35
004C'    22 67F5   JOINTS: SHLD   TXTWIN+C.DP
004F'    E1      POP    H
0050'    CD 0000:04 CALL   ARG
0053'    040404081E .BYTE  $IVAL,$IVAL,$IVAL,$STRADR,$REPEAT
0058'    FDE5     PUSH  Y
005A'    E5      PUSH  H
005B'    21 01C5'   LXI    H,TOUTCH
005E'    22 6540   SHLD   PRINTR
0061'    DD21 67E2 LXI    X,TXTWIN

```

Tangent

Tangent

```

0065'  FD5E01          MOV      E,1(Y)
0068'  FD5602          MOV      D,2(Y)
006B'  CD 0391'        CALL     SETCX
006E'  FD5E06          MOV      E,6(Y)
0071'  FD5607          MOV      D,7(Y)
0074'  CD 0398'        CALL     SETCY
0077'  FD7E0B          MOV      A,11(Y)
007A'  DD7712          MOV      C,C1(X),A
007D'  11 000F          LXI      D,15
0080'  FD19             DADY     D
0082'  CD 0000:0D       CALL     PNXT
0085'  3E0A             MVI      A,NL
0087'  CD 0000:0E       CALL     POUTCH
008A'  21 0000:0C       LXI      H,OUTCH
008D'  22 6540          SHLD     PRINTR
0090'  E1              POP      H
0091'  FDE1            POP      Y
0093'  C9              RET
0094'

TSX:
; SETUP LARGE CHARACTER SET
0094'  E5              LARGE:  PUSH H
0095'  21 0111'        LXI      H,DIS57
0098'  22 67E0          SHLD     SCRWIN+C.DP
009B'  21 0806          LXI      H,0806H
009E'  22 67DC          SHLD     SCRWIN+C.XF
00A1'  180D            JMPR     CHOUTC
00A3'  E5              SMALL:  PUSH H
00A4'  21 0435'        LXI      H,DIS35
00A7'  22 67E0          SHLD     SCRWIN+C.DP
00AA'  21 0604          LXI      H,0604H
00AD'  22 67DC          SHLD     SCRWIN+C.XF
00B0'  3E0C            CHOUTC: MVI      A,CNTRL
00B2'  CD 01AF'        CALL     ZOUTCH
00B5'  AF              XRA     A          ; ENABLE DISPLAY
00B6'  32 65CF          STA     VDNLF
00B9'  E1              ..DONE: POP H
00BA'  C9              RET

; MODIFY WINDOW PARAMETERS
M.CMD[CHARWINDOW,..CHWX,0,COSINE,..CHWA,..CHWS][
00BB'  18              +CHARWINDOW: .BYTE $CMDADR
00BC'  06              +          .BYTE (..CHWX-CHARWINDOW)/16+1
00BD'  00              +          .BYTE 0
00BE'  0000:05        +          .WORD COSINE
00C0'  00CF'          +          .WORD ..CHWS
00C2'  00D2'          +          .WORD ..CHWA
00C4'  434841525749+ .ASCIZ /CHARWINDOW/
+ ]

00CF'  CD 0000:04      ..CHWS: CALL     ARG
00D2'  040404040404  ..CHWA: .BYTE  $IVAL,$IVAL,$IVAL,$IVAL,$IVAL,$IVAL,$TAF

00D9'  E5              PUSH     H
00DA'  FD6E01          MOV      L,BX.X(Y)
00DD'  FD6602          MOV      H,BX.X+1(Y)
00E0'  22 67D0          SHLD     SCRWIN+C.X
00E3'  FD6E06          MOV      L,BX.Y(Y)

```

```

00E6'   FD6607           MOV     H,BX.Y+1(Y)
00E9'   22 67D2        SHLD   SCRWIN+C.Y
00EC'   FD6E0B         MOV     L,BX.XS(Y)
00EF'   FD660C         MOV     H,BX.XS+1(Y)
00F2'   22 67D4        SHLD   SCRWIN+C.XS
00F5'   FD6E10         MOV     L,BX.YS(Y)
00F8'   FD6611         MOV     H,BX.YS+1(Y)
00FB'   22 67D6        SHLD   SCRWIN+C.YS
00FE'   FD7E15         MOV     A,BX.MOD(Y)
0101'   32 67DF        STA    SCRWIN+C.C1
0104'   FD7E1A         MOV     A,BX.MOD+5(Y)
0107'   32 67DE        STA    SCRWIN+C.CO
010A'   3E0C           MVI    A,CNTRL
010C'   CD 01AF'       CALL   ZOUTCH
010F'   E1             POP    H
0110'   C9             RET
0111'
      ..CHWX:
      ; DISPLAY 5 X 7 CHARACTER
0111'   FE20           DIS57: CPI     20H
0113'   3002           JRNC   ..LOK
0115'   3E3F           ..UBAD: MVI   A,'?'
0117'   FE7B           ..LOK: CPI     7BH
0119'   30FA           JRNC   ..UBAD
      ; SET FC AND BC
011B'   4F             MOV     C,A
011C'   DD7E12        MOV     A,C.C1(X)
011F'   07             RLC
0120'   07             RLC
0121'   DDAE11        XRA    C.CO(X)
0124'   E60C           ANI    0CH
0126'   DDAE11        XRA    C.CO(X)
0129'   D319           OUT    XPAND
      ; LOAD BASE OF APPROPRIATE CHAR SET
012B'   79             MOV     A,C
012C'   2A 020B        LHLD   20BH ; LOAD THRU DOPE VECTOR
012F'   FE61           CPI     61H
0131'   3807           JRC    ..UPER
0133'   21 05CE'       LXI    H,LC57 ; WRONG
0136'   D661           SUI    61H
0138'   1802           JMPR   ..JOIN
013A'   D620           ..UPER: SUI   20H
013C'   4F             ..JOIN: MOV   C,A
013D'   0600           MVI    B,0
013F'   3E07           MVI    A,7
0141'   09             ..ADL: DAD   B
0142'   3D             DCR    A
0143'   20FC           JRNZ   ..ADL
      ; HL=CHAR START, DE=COORD
0145'   7A             MOV     A,D
0146'   C604           ADI    4
0148'   57             MOV     D,A
0149'   1D             DCR    E
014A'   1D             DCR    E
014B'   E5             PUSH   H
014C'   CD 0000:0F     CALL   R2A

```

```

014F'  CBB4          RES      6,H
0151'  F628          ORI      28H
0153'  D30C          OUT      MAGIC
0155'  D1            POP      D
0156'  0607          MVI      B,7
0158'  1A            ..LOOP: LDAX   D
0159'  77            MOV      M,A
015A'  23            INX     H
015B'  77            MOV      M,A
015C'  D5            PUSH   D
015D'  11 0027       LXI     D,39
0160'  19            DAD     D
0161'  D1            POP      D
0162'  13            INX     D
0163'  10F3          DJNZ   ..LOOP
0165'  C9            RET

; INITIALIZE WONDERFULL CONTROL PACKET
0166'  21 0435'     INIVDR: LXI H,DIS35
0169'  22 67E0       SHLD  SCRWIN+C.DP      ;SMALL WINDOW
016C'  21 0604       LXI H,0604H      ;STUFF
016F'  22 67DC       SHLD  SCRWIN+C.XF
0172'  1811         JMPR  INIVDA
0174'  21 0066       INIVDM: LXI H,102
0177'  22 67EB       SHLD  TXTWIN+C.YS
017A'  22 67D6       SHLD  SCRWIN+C.YS
017D'  21 0000       LXI H,0          ;CENTER Y
0180'  22 67D2       SHLD  SCRWIN+C.Y
0183'  1812         JMPR  INIVDX
0185'  21 004E       INIVDA: LXI H,78      ;TWO LINES
0188'  22 67EB       SHLD  TXTWIN+C.YS
018B'  21 0018       LXI H,24
018E'  22 67D6       SHLD  SCRWIN+C.YS
0191'  21 FFD9       LXI H,-39
0194'  22 67D2       SHLD  SCRWIN+C.Y
0197'  21 00A0       INIVDX: LXI      H,160
019A'  22 67E9       SHLD  TXTWIN+C.XS
019D'  22 67D4       SHLD  SCRWIN+C.XS
01A0'  3E05          MVI A,5
01A2'  32 67DF       STA   SCRWIN+C.C1
01A5'  3E01          MVI A,NBLKM
01A7'  32 67CF       STA   SCRWIN+C.ST
01AA'  3E0C          MVI A,CNTRL      ;CLEAN SCREEN
01AC'  C3 01AF'     JMP  ZOUTCH

; OUTPUT CHARACTER ON SCREEN
01AF'  DDE5         ZOUTCH: PUSH   X
01B1'  E5           PUSH   H
01B2'  21 65CF       LXI   H,VDNLF
01B5'  CB46          BIT   0,M
01B7'  E1           POP   H
01B8'  2008          JRNZ  ..NOZ
01BA'  DD21 67CD     LXI   X,SCRWIN      ; ** I THINK **
01BE'  37           STC           ; REALLY DO IT
01BF'  CD 01C6'     CALL  DCHAR
01C2'  DDE1         ..NOZ: POP   X
01C4'  C9           RET

```

```

; ENTRY TO ALWAYS TYPE
01C5' 37 TOUCH: STC ; PLEASE TYPE
; DRAW CHARACTER IN WINDOW
; ENTRY: IX=WINDOW DESCRIBER
; A=CHARACTER
; IF CY SET - ACTUALLY DRAW THE CHARACTER, ELSE
; UPDATE CX,CY AS IF CHARACTER WERE DRAWN
01C6' FDE5 DCHAR: PUSH Y
01C8' E5 PUSH H
01C9' D5 PUSH D
01CA' C5 PUSH B
01CB' F5 PUSH PSW
01CC' FD21 67AD LXI Y,TMPARG
01D0' FE0A CPI NL ; NEW LINE CHAR?
01D2' 2064 JRNZ ..NONL
; IF REAL DRAWING WANTED - PLOP SPACES OVER REMAINING SL
; OTS
; ON THIS LINE
01D4' F1 POP PSW
01D5' F5 PUSH PSW
01D6' 3018 JRNC ..KBND
01D8' DDCB0246 BIT NBLKB,C.ST(X)
01DC' 2812 JRZ ..KBND
01DE' CD 0331' ..KILB: CALL COMURL
01E1' CD 0000:06 CALL CPHLDE ; A SPACE TO NAIL?
01E4' 2802 JRZ ..KBTR
01E6' 3008 JRNC ..KBND
01E8' CD 03AB' ..KBTR: CALL BOXXYF
01EB' CD 030C' CALL ADVCX
01EE' 18EE JMPR ..KILB
01F0' ..KBND:
01F0' CD 034F' CALL SCXLLL ; SET LEFT LOWER LIMIT
; IS CY LT -YS/2+YF+(YF-1)/2?
01F3' DD5E10 MOV E,C.YF(X) ; D ALREADY=0
01F6' 62 MOV H,D
01F7' 6B MOV L,E
01F8' CD 0421' CALL DVDEM2
01FB' 19 DAD D
01FC' DD5E09 MOV E,C.YS(X)
01FF' DD560A MOV D,C.YS+1(X)
0202' 1B DCX D
0203' CD 042A' CALL DVDE2N
0206' 19 DAD D
0207' CD 038A' CALL GETCY
020A' CD 0000:06 CALL CPHLDE ; HL=LL, DE=CY
020D' 3817 JRC ..NOSC ; SKIP SCROLLING IF OK
020F' 2815 JRZ ..NOSC
; CALL SCROLL ROUTINE
0211' DDCB02E6 SET SCNEED,C.ST(X) ; SET SCROLLING WAS NEED
ED FLAG
0215' F1 POP PSW
0216' F5 PUSH PSW
0217' D2 0305' JNC ..DONE ; SKIP IT IF DRAW OFF
021A' DD7E10 MOV A,C.YF(X)
021D' CD 03ED' CALL BLDBOX

```



```

0220'   CD 0000:10           CALL   SCROLE
0223'   C3 0305'           JMP    ..DONE
                                ; SCROLLING NOT NEEDED - ADVANCE CY
0226'   EB                 ..NOSC: XCHG                    ; CY TO HL
0227'   DD5E10             MOV    E,C.YF(X)
022A'   1600               MVI    D,0
022C'   A7                 ANA    A
022D'   ED52               DSBC   D
022F'   DD750D             MOV    C,CY(X),L
0232'   DD740E             MOV    C,CY+1(X),H
0235'   C3 0305'           JMP    ..DONE
0238'   FE7F               ..NONL: CPI    RUBOUT
023A'   2066               JRNZ   ..NORB
                                ; ZONK OUT THE CURSOR
023C'   F1                 POP    PSW
023D'   F5                 PUSH   PSW
023E'   3003               JRNC   ..NZNK
0240'   CD 03AB'           CALL   BOXXYF                    ; KILL CURSOR
0243'                               ..NZNK:
                                ; RUBOUT CHARACTER - BACKUP CX
0243'   DD5E0F             MOV    E,C.XF(X)
0246'   1600               MVI    D,0
0248'   DD6E0B             MOV    L,C.CX(X)
024B'   DD660C             MOV    H,C.CX+1(X)
024E'   A7                 ANA    A
024F'   ED52               DSBC   D
                                ; DID WE BACKUP TO BEFORE THE BEGINNING OF THIS LINE?
0251'   DD5E07             MOV    E,C.XS(X)
0254'   DD5608             MOV    D,C.XS+1(X)
0257'   1B                 DCX    D
0258'   CD 042A'           CALL   DVDE2N                    ; DE--XS/2
025B'   CD 0000:06         CALL   CPHLDE                    ; HL= NEW CX
025E'   3031               JRNC   ..NOWR                    ; JUMP TO STORE GOODIE
                                ; WRAP AROUND TO PREVIOUS LINE
0260'   DDCB02EE           SET    RUBACK,C.ST(X)           ; YES - SET STATUS BIT
0264'   CD 038A'           CALL   GETCY
0267'   DD6E10             MOV    L,C.YF(X)
026A'   2600               MVI    H,0
026C'   19                 DAD    D
026D'   E5                 PUSH   H                        ; CHECK FOR BACKUP PAST
                                START
026E'   CD 031C'           CALL   COMUTL
0271'   D1                 POP    D
0272'   CD 0000:06         CALL   CPHLDE
0275'   DA 0305'           JC     ..DONE                    ; SKIPOUT IF YES
0278'   CD 0398'           CALL   SETCY
                                ; SET CX TO LAST VALID POSITION ON PREVIOUS LINE
027B'   CD 034F'           CALL   SCXLLL                    ; HL=CX OF LLL
027E'   CD 0331'           CALL   COMURL                    ; DE=URL
0281'   DD4E0F             MOV    C,C.XF(X)
0284'   0600               MVI    B,0
0286'   09                 ..GROK: DAD   B                    ; KEEP ADDING TILL
0287'   CD 0000:06         CALL   CPHLDE                    ; WE HIT THE LIMIT
028A'   38FA               JRC    ..GROK
028C'   28F8               JRZ    ..GROK

```

```

028E'  A7          ANA      A          ; SUBTRACT TO LEGALIZE
028F'  ED42       DSBC     B
0291'  DD750B     ..NOWR: MOV     C,CX(X),L
0294'  DD740C     MOV     C,CX+1(X),H
0297'  F1        POP     PSW          ; CHECK CY STATUS
0298'  F5        PUSH    PSW
0299'  D2 0305'   JNC     ..DONE
; DRAW ERASING BOX TO SNUFF RUBBED OUT CHAR
029C'  CD 03AB'   CALL    BOXXYF
029F'  C3 0305'   JMP     ..DONE
02A2'  FE0C     ..NORB: CPI     CNTRL   ; SCREEN ERASE??
02A4'  2015     JRNZ    ..NOCL
; SCREEN ERASE CHARACTER
02A6'  F1        POP     PSW          ; CHECK FOR CY STATUS
02A7'  F5        PUSH    PSW
02A8'  300B     JRNC    ..FUZZ
02AA'  DD7E11     MOV     A,C.CO(X)   ; GET BC
02AD'  C604     ADI     4
02AF'  CD 03ED'   CALL    BLDBOX
02B2'  CD 0000:07 CALL    DOBOX       ; DRAW BOX TO ERASE
02B5'  CD 0372'   ..FUZZ: CALL    RESCX Y
02B8'  C3 0305'   JMP     ..DONE
; CHECK FOR CR, IF FOUND IGNORETH IT
02BB'  FE0D     ..NOCL: CPI     ODH
02BD'  2846     JRZ     ..DONE
; NORMAL GOOD GUY CHARACTER
; DO WE HAVE LINE WRAPAROUND??
02BF'  CD 0331'   CALL    COMURL     ; GET UPPER RIGHT LIMIT
02C2'  CD 0000:06 CALL    CPHLDE
02C5'  3809     JRC     ..CXOK
02C7'  2807     JRZ     ..CXOK
; WRAP AROUND - FORCE NEWLINE
02C9'  F1        POP     PSW
02CA'  F5        PUSH    PSW          ; PASS SAME CY STATUS TO
NEWLINE
02CB'  3E0A     MVI     A,NL
02CD'  CD 01C6'   CALL    DCHAR
; DRAW A BOX TO ERASE BENEATH
02D0'  F1        ..CXOK: POP    PSW
02D1'  F5        PUSH    PSW
02D2'  302E     JRNC    ..CLIP
02D4'  CD 03A3'   CALL    BOXXY1
; DID THAT BOX GET CLIPPED??
02D7'  FD7E0B     MOV     A,BX.XS(Y)
02DA'  DDBE0F     CMP     C.XF(X)
02DD'  2023     JRNZ    ..CLIP
02DF'  FD7E10     MOV     A,BX.YS(Y)
02E2'  DDBE10     CMP     C.YF(X)
02E5'  201B     JRNZ    ..CLIP
; WE HAVE A CLEAR SHOT
02E7'  DD7E0B     MOV     A,C.CX(X)   ; SET DE=TRANSLATED COOR
DINATES
02EA'  DD8603     ADD     C.X(X)
02ED'  5F        MOV     E,A
02EE'  DD7E0D     MOV     A,C.CY(X)

```

```

02F1' DD8605 ADD C,Y(X)
02F4' 57 MOV D,A
02F5' F1 POP PSW ; RESTORE CHAR
02F6' F5 PUSH PSW
02F7' 21 0302' LXI H,..COME ; PUT RETURN POINT ON ST
                                ACK
02FA' E5 PUSH H
02FB' DD6E13 MOV L,C.DP(X) ; HL=ROUTINE ADDR
02FE' DD6614 MOV H,C.DP+1(X)
0301' E9 PCHL ; BRANCHEROONIE
0302' ..COME:
; UPDATE CX
0302' CD 030C' ..CLIP: CALL ADVCX
; ITS ALL OVER WITH
0305' F1 ..DONE: POP PSW
0306' C1 POP B
0307' D1 POP D
0308' E1 POP H
0309' FDE1 POP Y
030B' C9 RET
; SUBROUTINE TO ADVANCE CX BY ONE SLOT
030C' CD 0383' ADVCX: CALL GETCX
030F' DD6E0F MOV L,C.XF(X)
0312' 2600 MVI H,0
0314' 19 DAD D
0315' DD750B MOV C,CX(X),L
0318' DD740C MOV C,CX+1(X),H
031B' C9 RET
; SUB TO COMPUTE UPPER - TOP LIMIT OF WINDOW
; THIS LIMIT IS RETURNED IN HL
; DE IS CLOBBBERED
031C' DD5E09 COMUTL: MOV E,C.YS(X)
031F' DD560A MOV D,C.YS+1(X)
0322' 13 INX D
0323' CD 0422' CALL DVDE2
0326' EB XCHG
0327' DD5E10 MOV E,C.YF(X)
032A' 1600 MVI D,0
032C' CD 042A' CALL DVDE2N
032F' 19 DAD D
0330' C9 RET
; SUB TO COMPUTE UPPER RIGHT WINDOW LIMIT
; RETURNS: DE=URL, HL=CX
0331' DD5E07 COMURL: MOV E,C.XS(X)
0334' DD5608 MOV D,C.XS+1(X)
0337' 13 INX D
0338' CD 0422' CALL DVDE2 ; DE=XS/2
033B' EB XCHG
033C' DD5E0F MOV E,C.XF(X)
033F' 1600 MVI D,0
0341' CD 0421' CALL DVDEM2
0344' A7 ANA A
0345' ED52 DSBC D
0347' EB XCHG
0348' DD6E0B MOV L,C.CX(X) ; HL=CX

```

```

034B' DD660C          MOV     H,C.CX+1(X)
034E' C9             RET
; SUB TO COMPUTE LEFT LOWER LIMIT OF WINDOW
; AND SET IT
; LLL RETURNED IN HL, DE IS CLOBBED
034F' DD5E07          SCXLLL: MOV     E,C.XS(X)
0352' DD5608          MOV     D,C.XS+1(X)
0355' 1B             DCX     D
0356' CD 042A'        CALL    DVDE2N
0359' EB             XCHG
035A' DD5E0F          MOV     E,C.XF(X)
035D' 1600           MVI     D,0
035F' CD 0422'        CALL    DVDE2
0362' 19             DAD     D
0363' DD750B          MOV     C.CX(X),L
0366' DD740C          MOV     C.CX+1(X),H
0369' C9             RET
; SUBROUTINE TO RESET CX TO LEFT SIDE OF SCREEN
036A' D5             RESCX:  PUSH   D
036B' E5             PUSH   H
036C' CD 034F'        CALL    SCXLLL
036F' E1             POP    H
0370' D1             POP    D
0371' C9             RET
; SUBROUTINE TO RESET CX AND CY
0372' CD 036A'        RESCXY: CALL   RESCX           ; DO CX
; AND FALL INTO ...
; SUBROUTINE TO RESET CY TO TOP OF WINDOW
0375' D5             RESCY:  PUSH   D
0376' E5             PUSH   H
0377' CD 031C'        CALL    COMUTL
037A' DD750D          MOV     C.CY(X),L
037D' DD740E          MOV     C.CY+1(X),H
0380' E1             POP    H
0381' D1             POP    D
0382' C9             RET
; ROUTINE TO LOAD DE WITH CX
0383' DD5E0B          GETCX:  MOV     E,C.CX(X)
0386' DD560C          MOV     D,C.CX+1(X)
0389' C9             RET
; SAME FOR CY
038A' DD5E0D          GETCY:  MOV     E,C.CY(X)
038D' DD560E          MOV     D,C.CY+1(X)
0390' C9             RET
; ROUTINE TO SET CX
0391' DD730B          SETCX:  MOV     C.CX(X),E
0394' DD720C          MOV     C.CX+1(X),D
0397' C9             RET
; SAME FOR CY
0398' DD730D          SETCY:  MOV     C.CY(X),E
039B' DD720E          MOV     C.CY+1(X),D
039E' C9             RET
; SUBROUTINE TO DRAW A BOX OF SIZE XF,YF AT
; X+CX, Y+CY OF COLOR 3
039F' 3E03           PUTCUR: MVI     A,3

```

```

03A1' 180F JMPR BOXXY2
03A3' DDCB1256 BOXXY1: BIT ERABIT,C,C1(X)
03A7' 3E00 MVI A,0
03A9' 2807 JRZ BOXXY2
03AB' DD7E11 BOXXYF: MOV A,C,C0(X)
03AE' E603 JOINXY: ANI 3
03B0' C604 ADI 4
03B2' FD7715 BOXXY2: MOV BX,MOD(Y),A
03B5' DD6E03 MOV L,C,X(X)
03B8' DD6604 MOV H,C,X+1(X)
03BB' CD 0383' CALL GETCX
03BE' 19 DAD D
03BF' 2B DCX H
03C0' FD7501 MOV BX,X(Y),L
03C3' FD7402 MOV BX,X+1(Y),H
;
03C6' DD6E05 MOV L,C,Y(X)
03C9' DD6606 MOV H,C,Y+1(X)
03CC' CD 038A' CALL GETCY
03CF' 19 DAD D
03D0' FD7506 MOV BX,Y(Y),L
03D3' FD7407 MOV BX,Y+1(Y),H
;
03D6' DD5610 MOV D,C,YF(X)
03D9' DD5E0F MOV E,C,XF(X)
03DC' AF XRA A
03DD' FD730B MOV BX,XS(Y),E
03E0' FD770C MOV BX,XS+1(Y),A
03E3' FD7210 MOV BX,YS(Y),D
03E6' FD7711 MOV BX,YS+1(Y),A
;
03E9' CD 0000:07 CALL DOBOX
03EC' C9 RET
; SUBROUTINE TO PREPARE STACK FOR CALL TO BOX-LIKE SUBRO
UTINE
; A=MODE PARAMETER TO STUFF
03ED' FD7715 BLDBOX: MOV BX,MOD(Y),A ; STUFF MODE
03F0' DD5E03 MOV E,C,X(X) ; XFER X
03F3' DD5604 MOV D,C,X+1(X)
03F6' FD7301 MOV BX,X(Y),E
03F9' FD7202 MOV BX,X+1(Y),D
03FC' DD5E05 MOV E,C,Y(X) ; XFER Y
03FF' DD5606 MOV D,C,Y+1(X)
0402' FD7306 MOV BX,Y(Y),E
0405' FD7207 MOV BX,Y+1(Y),D
;
0408' DD5E07 MOV E,C,XS(X) ; X SIZE
040B' DD5608 MOV D,C,XS+1(X)
040E' FD730B MOV BX,XS(Y),E
0411' FD720C MOV BX,XS+1(Y),D
0414' DD5E09 MOV E,C,YS(X) ; Y SIZE
0417' DD560A MOV D,C,YS+1(X)
041A' FD7310 MOV BX,YS(Y),E
041D' FD7211 MOV BX,YS+1(Y),D
0420' C9 RET

```

```

; DIVIDE DE-1 BY 2
0421' 1B DVDEM2: DCX D ; DE=DE-1
; FALL INTO...
; DIVIDE DE BY 2
0422' A7 DVDE2: ANA A
0423' 7A MOV A,D
0424' 1F RAR
0425' 57 MOV D,A
0426' 7B MOV A,E
0427' 1F RAR
0428' 5F MOV E,A
0429' C9 RET
; DIVIDE DE BY 2 THAN NEGATE
042A' CD 0422' DVDE2N: CALL DVDE2
042D' 7A MOV A,D
042E' 2F CMA
042F' 57 MOV D,A
0430' 7B MOV A,E
0431' 2F CMA
0432' 5F MOV E,A
0433' 13 INX D
0434' C9 RET
; TRANSFER 3 X 5 CHARACTER TO THE SCREEN
; CHAR PASSED IN A
; IX=WINDOW DESC
; D=Y, E=X
0435' FE20 DIS35: CPI / / ; CHAR IN RANGE?
0437' 3002 JRNC ..LOK
0439' 3E3F ..UBAD: MVI A,'?'
043B' FE7B ..LOK: CPI 7BH
043D' 30FA JRNC ..UBAD
; LOWER CASE CHARACTER????
043F' 47 MOV B,A
0440' DD4E12 MOV C,C.C1(X)
0443' FE61 CPI 61H
0445' 380B JRC ..UPER
0447' D620 SUI 20H
0449' 47 MOV B,A
044A' 79 MOV A,C
044B' 3C INR A
044C' E603 ANI 3
044E' 2001 JRNZ ..NOTZ
0450' 3C INR A
0451' 4F ..NOTZ: MOV C,A
0452' 79 ..UPER: MOV A,C ; SET COLORS
0453' 07 RLC
0454' 07 RLC
0455' DD4E11 MOV C,C.CO(X)
0458' A9 XRA C
0459' E60C ANI 0CH
045B' A9 XRA C
045C' D319 OUT XPAND
; LOOKUP CHARACTER
045E' 78 MOV A,B
045F' D620 SUI 20H

```

```

0461' 4F          MOV     C,A
0462' 0600       MVI     B,0
0464' 60        MOV     H,B
0465' 69        MOV     L,C
0466' 29        DAD     H
0467' 29        DAD     H
0468' 09        DAD     B
0469' 01 048E'   LXI     B, FONT35
046C' 09        DAD     B
046D' E5        PUSH    H
046E' 7A        MOV     A,D
046F' C603      ADI     3
0471' 57        MOV     D,A
0472' 1D        DCR     E
0473' CD 0000:0F CALL    R2A
0476' D1        POP     D
0477' CBB4      RES     6,H
0479' F628      ORI     28H
047B' D30C      OUT    MAGIC
047D' 0605      MVI     B,5      ; SET LOOP CTR
047F' 1A        ..LOOP: LDAX   D
0480' 77        MOV     M,A
0481' 23        INX     H
0482' 3600      MVI     M,0
0484' D5        PUSH    D
0485' 11 0027   LXI     D,39
0488' 19        DAD     D
0489' D1        POP     D
048A' 13        INX     D
048B' 10F2     DJNZ   ..LOOP
048D' C9        RET
; 3 X 5 CHAR SET

```

CHARS -
 +++++ SYMBOL TABLE +++++

AASN 005F	ADVCX 030C'	ARG 0000:04 X	ARGSTK 60CC
BACKGR 65CC	BLANK 0020	BLDBOX 03ED' I	BOTRAM 6000
BOTTOM 64A9	BOXXY1 03A3'	BOXXY2 03B2'	BOXXYF 03AB'
BX.MOD 0015	BX.X 0001	BX.XS 000B	BX.Y 0006
BX.YS 0010	CHARSL 65DD	CHARWI 00BB' I	CHOUTC 00B0'
CLEAR5 60CA	CNTRL 000C	CNTRLC 65CD	CNTRL0 65D9
CNTRLU 0015	CNTRLZ 65E4	COMURL 0331'	COMUTL 031C'
COSINE 0000:05 X	CPHLDE 0000:06 X	CPLARE 611C	CPLSIZ 0140
CR 000D	CSBLOK 668D	CSFLAG 6260	CURCX 6814
CURCY 6816	CURREN 64A5	C.CO 0011	C.C1 0012
C.CX 000B	C.CY 000D	C.DP 0013	C.ST 0002
C.X 0003	C.XF 000F	C.XS 0007	C.Y 0005
C.YF 0010	C.YS 0009	DCHAR 01C6' I	DDTON 65CE
DEVBL 6589	DEVCL0 6579	DEVCL1 657B	DEVCL2 657D
DEVCL3 657F	DEVCL4 6581	DEVCL5 6583	DEVCL6 6585
DEVCL7 6587	DEVFB 65CB	DEVHCB 658F	DEVMO 658B
DEVNM 65B7	DEVNT 658D	DEVTNA 65BB	DEVTNB 65BF
DEVTNC 65C3	DEVVA 65BD	DEVVAR 6579	DEVVB 65C1
DEVVBL 6591	DEVVC 65C5	DEVVD 65C9	DEVVN 65B9
DEVVS 65C7	DEVXCD 65B3	DEVYCD 65B5	DIS35 0435'
DIS57 0111'	DOBOX 0000:07 X	DOLPLH 62E8	DOLPPT 62EA
DUMBST 6577	DVDE2 0422' I	DVDE2N 042A'	DVDEM2 0421'
EDBCNT 64AD	EDCCNT 64A7	EDLONG 6812	EDMODE 681C
EDNAME 64A1	EDNCX 680C	EDNCY 680E	EDNEWS 64A5
EDOCX 6808	EDOCY 680A	EDPN 6806	EDPO 6804
EDPTRC 64AB	EDPTRL 64A9	EDSTR 681A	ERABIT 0002
ERRPGM 0000:08 X	ER.ARA 002F	ER.ARG 0034	ER.ASN 0015
ER.BOX 001A	ER.CHN 0002	ER.CMD 001F	ER.CNV 0016
ER.COR 001B	ER.CTL 0036	ER.DEL 0026	ER.DIM 0030
ER.DIV 0018	ER.DP 0037	ER.DSK 0019	ER.EDT 0035
ER.FMT 0038	ER.FNF 001C	ER.FOR 0028	ER.IMP 0003
ER.LAB 0025	ER.MAC 0022	ER.NAE 002D	ER.NAM 0029
ER.NEG 003A	ER.NOT 0023	ER.NUL 0039	ER.NUM 002B
ER.NXT 001D	ER.OFL 0017	ER.OPN 0014	ER.OVE 001E
ER.PAR 002A	ER.REN 002C	ER.RET 0024	ER.SEP 0021
ER.SNP 0031	ER.SPC 002E	ER.STK 0004	ER.SW 0032
ER.TER 0020	ER.UFL 0033	ER.UNF 0027	EXTDEL 002E
E.HVAL 0002	E.LVAL 0001	E.SIZ 0005	E.TYP 0000
E.VAL 0001	FCNTH 65D2	FCNTI 65D3	FCNTJ 65D4
FCNTK 65D5	FCNTL 65D6	FCNTV 65E0	FCNTY 65E3
FIRST 64A3	FLAGS 65CB	FONT35 048E'	FOREGR 65D0
FRAGSI 0400	FREELS 65E5	FSTDOL 648C	FSTINT 62EC
FWDPTR 65E7	GETCX 0383' I	GETCY 038A' I	HCAREA 6593
INC5IY 0000:09 X	INCRO 65E9	INIVDA 0185' I	INIVDM 0174' I
INIVDR 0166' I	INIVDX 0197'	ISNL 0000:0A X	JOINTS 004C'
JOINXY 03AE'	JUNK 6542	KBLOCK 65F1	KEYFLG 67FF
KEYPTK 6533	KEYTRK 67F7	KISNL 0000:0B X	LARGE 0094' I
LC57 05CE'	LF 000A	LISTON 65E2	MACSTU 6536
MACTOP 625E	MAGIC 000C	MAXFRG 0040	MNMX 65EB
NBLKB 0000	NBLKM 0001	NEWBOT 6810	NL 000A
NSADDR 6802	NUMBUF 64A3	OLDCHR 64A0	OLDCUR 649F
OLDKEY 649D	OLDXY 65EF	ONEBUF 6729	OPRL 0014
OPRSP 655B	OPRSTK 6547	OPRSZ 655D	OUTCH 0000:0C X
OUTOFF 62E7	PCNT 655E	PIXVAL 65ED	PNXT 0000:0D X
POINTE 64A7	PONOFF 65DA	POUTCH 0000:0E X	PRINTR 6540

CHARS -

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

PUTCUR 039F'	I	R2A 0000:0F	X	RAMEND 7FFF		RAMSTR 6900	
RANSHT 655F		RESCX 036A'	I	RESCXY 0372'	I	RESCY 0375'	I
RMDTMP 6538		RUBACK 0005		RUBOUT 007F		SAVESP 625C	
SCNEED 0004		SCROLE 0000:10	X	SCRWIN 67CD		SCXLLL 034F'	
SETCX 0391'	I	SETCY 0398'	I	SFREEL:11	X	SMALL 00A3'	I
SOPRSP 653D		SOPRSZ 653F		STACK 60C8		STAKTD 6000	
STRSIZ 6800		SUBSTU 6534		TAB 0009		TAPBUF 64B3	
TAPCON 64AF		TAPPRO 64B1		TBFEND 6533		TEMPHD 60CA	
TEMPS 62E5		TEXT 0000'	I	TEXTA 000F'		TEXTS 0029'	
TEXTX 0035'		THEN 0000:12	X	TMPARG 67AD		TOP 6818	
TOUTCH 01C5'		TS 0035'		TSS 0042'		TSX 0094'	
TTYBEG 6261		TTYEND 62E1		TTYINT 62E3		TTYPTR 62E1	
TXTWIN 67E2		UARTFL 649C		USREND 65F1		V3PTR 6573	
VDCHAR 649E		VDNLF 65CF		VIPLH 6545		VOICE0 6563	
WRMODE 65EE		XPAND 0019		ZGIM2 0001		ZGREND 681D	
ZOUTCH 01AF'	I	\$AADR 0010		\$ADDRF 0007		\$ADDRI 0005	
\$ADDRS 0009		\$ANY 001A		\$ANYNA FFFC		\$ANYVA FFFE	
\$ARGPT 0011		\$BGPTR 000F		\$BNDL 0007		\$CALLE 000D	
\$CMDAD 0018		\$CPLBL 002A		\$CSBLO 0028		\$DATAP 0007	
\$DOLDE 0001		\$DOLDO 0002		\$DVAL 0000		\$END 001C	
\$FADR 000E		\$FLAGS 0002		\$FORBL 0024		\$FORPT 000B	
\$FVAL 0006		\$GOSUB 001A		\$IADR 000C		\$INPBU 0018	
\$INPPT 0016		\$IVAL 0004		\$KEYBL 0026		\$LENGT 0001	
\$LINPT 0005		\$LOCPT 0009		\$MIBBL 0022		\$MIBEN 001B	
\$NAMAD 000A		\$NAME 000A		\$NASCI 0009		\$NDEL 0080	
\$NLINK 0003		\$NULL 0002		\$NVALU 0005		\$REPEA 001E	
\$RVSTU 0013		\$SAME 0020		\$SASCI 000A		\$SLEN 0006	
\$STRAD 0008		\$STRPT 0003		\$TAF 0000		\$TYPE 0000	
\$USE 0005		.BLNK. 0000:03	X	.DATA. 0000*	X	.PRG. 05CE'	X

```
      ;DSKSHIT=1
      .INSERT A:S.ASM
@.REMARK /
@
@           *
@           * * *
@           ***
@           *****
@           *
@           *
@           *
@           *
@           *
@           *****
@
@           "WHEN YOU CARE ENOUGH TO PROGRAM
@           THE VERY BEST"
@
@           ZGRASS V2.00000000
@BY JAY FENTON, NOLA DONATO, AND TOM DEFANTI
@           (C) 1978
@
@/
```

```
      .INSERT A:ZRAM.ASM
      .INSERT A:IOEQU.ASM
      .PREL
      .IDENT TAPEIO
      .EXTERN ALLOC
      .EXTERN ALSTR
      .EXTERN ARG
      .EXTERN CNTCK
      .EXTERN ENDSTR
      .EXTERN ERRPGM
      .EXTERN INC5IY
      .EXTERN INCUSE
      .EXTERN NXTVAL
      .EXTERN PI
      .INTERN GETDSK
      .INTERN PUTTAPE
      .INTERN UARTINT
```

; EQUATES:

; CPM ADDRESSES:

```
E106      BDOS=0E106H      ; BDOS ENTRY POINT
005C      FCBADR=5CH      ; DEFAULT FCB
0028      DBUFPT=40      ; ADDR OF NEXT BYTE IN DISC BUFFER
      ;
```

; CPM FUNCTION CODES:

```
0013      CPDELETE=19      ; DELETE
0014      CPREAD=20      ; READ
0016      CPCREATE=22
0015      CPWRITE=21
000F      CPOPEN=15      ; OPEN
0010      CPCLOSE=16      ; GOT IT YET?
      ;
```

```
00A5      FNCHAR=0A5H
0001      TXRDY=1
```

```

;
; .IFNDEF DSKSHIT,[
; OUTPUT TO TAPE COMMAND
M.CMD[PUTTAPE,..PUTX,0,PI,..PUTA,..PUTS][
0000' 18 +PUTTAPE: .BYTE $CMDADR
0001' 06 + .BYTE (..PUTX-PUTTAPE)/16+1
0002' 00 + .BYTE 0
0003' 0000:0D + .WORD PI
0005' 0012' + .WORD ..PUTS
0007' 0015' + .WORD ..PUTA
0009' 505554544150+ .ASCIZ /PUTTAPE/
+ ]

0011' 00 .BYTE 0
0012' CD 0000:06 ..PUTS: CALL ARG
0015' 040A1E ..PUTA: .BYTE $IVAL,$NAME,$REPEAT
0018' FDE5 PUSH Y
001A' E5 PUSH H
001B' CD 01BD' CALL TOUTEST
001E' CD 0000:0A CALL INC5IY
0021' FD7E00 ..PUTL: MOV A,0(Y) ; GET TYPE
0024' FE0A CPI $NAMADR ; IS IT A NAME?
0026' 2025 JRNZ ..SKIP ; NOPE
0028' FD6602 MOV H,2(Y)
002B' FD6E01 MOV L,1(Y)
002E' E5 PUSH H
002F' CD 0000:0C CALL NXTVAL
; WRITE A LEADER
0032' CD 01FE' CALL WRTLDR
0035' E3 XTHL ; WRITE OUT FILENAME
0036' CD 0216' CALL OUTNAM
0039' E1 POP H
003A' 23 INX H
003B' 7E MOV A,M
003C' 2B DCX H
003D' CD 0157' CALL SIZBC
0040' 7E ..OUTL: MOV A,M ; GET CHAR
0041' CD 020B' CALL OUTTAP ; WRITE IT OUT
0044' 23 INX H ; WE DO IT TO A
0045' 0B DCX B
0046' 78 MOV A,B
0047' B1 ORA C
0048' 20F6 JRNZ ..OUTL
004A' CD 01FE' CALL WRTLDR ; WRITE TRAILER
004D' CD 0000:0A ..SKIP: CALL INC5IY
0050' FD7E00 MOV A,0(Y) ; AT END OF LIST?
0053' FE00 CPI $TAF
0055' 20CA JRNZ ..PUTL
0057' CD 01E5' CALL SHUTOF ; TURN OFF MOTORS
005A' E1 POP H
005B' FDE1 POP Y
005D' C9 RET
005E' ..PUTX:
]
; INPUT FROM TAPE COMMAND
; .IFNDEF DSKSHIT,[
    
```

```

005E'          GETDSK:
                M.CMD[IGOTTAPE,..GETX,0,GETTAPE,..GETA,..GETS]I
005E'  18      +GOTTAPE:      .BYTE $CMDADR
005F'  07      +              .BYTE (..GETX-GOTTAPE)/16+1
0060'  00      +              .BYTE 0
0061'  00C7'   +              .WORD GETTAPE
0063'  0070'   +              .WORD ..GETS
0065'  0073'   +              .WORD ..GETA
0067'  474F54544150+
                .ASCIZ /GOTTAPE/
                +J

006F'  00              .BYTE 0
0070'  CD 0000:06     ..GETS: CALL ARG
0073'  040A1E         ..GETA: .BYTE $IVAL,$NAME,$REPEAT
0076'  E5             PUSH H
0077'  FDE5           PUSH Y
0079'  CD 01AC'       CALL TINPEST
007C'  CD 0000:0A     CALL INC5IY
                ; IS THIS NAME ANY GOOD?
007F'  FD6602         ..LOOP: MOV H,2(Y)
0082'  FD6E01         MOV L,1(Y)
0085'  7E             MOV A,M
0086'  FE02           CPI $NULL
0088'  202C           JRNZ ..SKIP
                ; AWAIT APPEARANCE OF FILENAME
008A'  CD 0164'       CALL NAMATH
                ; ALLOCATE MEMORY
008D'  CD 0000:05     CALL ALSTR
                ; COPY UNTIL NULL READ IN
0090'  CD 0188'       ..CLOP: CALL TAPGET
0093'  A7             ANA A                ; DID WE GET A NULL?
0094'  2807           JRZ ..GEOF
0096'  12             STAX D
0097'  13             INX D
0098'  0B             DCX B
0099'  78             MOV A,B
009A'  B1             ORA C
009B'  20F3           JRNZ ..CLOP
009D'  CD 0000:08     ..GEOF: CALL ENDSTR
00A0'  CD 0000:0B     CALL INCUSE
00A3'  FD6602         MOV H,2(Y)
00A6'  FD6E01         MOV L,1(Y)
00A9'  360A           MVI M,$NAMADR
00AB'  23             INX H
00AC'  23             INX H
00AD'  3600           MVI M,0
00AF'  01 0003        LXI B,$NVALUE-2
00B2'  09             DAD B
00B3'  73             MOV M,E
00B4'  23             INX H
00B5'  72             MOV M,D
00B6'  CD 0000:0A     ..SKIP: CALL INC5IY
00B9'  FD7E00         MOV A,0(Y)
00BC'  FE00           CPI $TAF
00BE'  20BF           JRNZ ..LOOP
                ; STOP THE MOTOR
    
```

```

00C0'   CD 01E5'           CALL    SHUTDF
00C3'   FDE1             POP     Y
00C5'   E1              POP     H
00C6'   C9             RET
00C7'
    ..GETX:
    M.CMD[GETTAPE,..GETX,0,AFIX[G],..GETA,..GETS][
00C7'   18             +GETTAPE: .BYTE  $CMDADR
00C8'   0A             + .BYTE  (..GETX-GETTAPE)/16+1
00C9'   00             + .BYTE  0
00CA'   634C           + .WORD  AFIX[G][("G"-101Q)*16+FSTINT]
00CC'   00D9'         + .WORD  ..GETS
00CE'   00DC'         + .WORD  ..GETA
00D0'   474554544150+ .ASCIZ /GETTAPE/
    +]

00D8'   00             .BYTE  0
00D9'   CD 0000:06     ..GETS: CALL   ARG
00DC'   040A1E        ..GETA: .BYTE  $IVAL,$NAME,$REPEAT
00DF'   E5            PUSH   H
00E0'   FDE5         PUSH   Y
00E2'   CD 01AC'     CALL   TINPEST
00E5'   CD 0000:0A   CALL   INC5IY
00E8'   FD6602       ..LOOP: MOV    H,2(Y)
00EB'   FD6E01       MOV    L,1(Y)
00EE'   7E           MOV    A,M
00EF'   FE02         CPI    $NULL
00F1'   2053         JRNZ  ..SKIP
00F3'   CD 0164'     CALL   NAMATH
00F6'   CD 0188'     CALL   TAPGET
00F9'   57           MOV    D,A      ; D=TYPE
00FA'   CD 0188'     CALL   TAPGET
00FD'   6F           MOV    L,A      ; L=LEN
00FE'   CD 0188'     CALL   TAPGET
0101'   5F           MOV    E,A      ; E=FLAGS
    ; IGNORE NEXT 3 BYTES

0102'   CD 0188'     CALL   TAPGET
0105'   CD 0188'     CALL   TAPGET
0108'   CD 0188'     CALL   TAPGET
010B'   7D           MOV    A,L
010C'   F5           PUSH   PSW
010D'   CD 0000:04   CALL   ALLOC
0110'   F1           POP    PSW
0111'   E5           PUSH   H
0112'   72           MOV    M,D      ; SET TYPE
0113'   23           INX   H
0114'   23           INX   H
0115'   73           MOV    M,E      ; AND FLAGS
0116'   23           INX   H
0117'   23           INX   H
0118'   23           INX   H      ; TO USE COUNT
0119'   3600        MVI    M,0      ; ZERO THAT OUT
011B'   23           INX   H
011C'   CD 0157'     CALL   SIZEBC
011F'   EB           XCHG      ; FUDGE SIZE TO SHOW GUYS ALREA

    DY GOT
0120'   21 FFFA      LXI    H,-6
    
```

```

0123' 09          DAD      B
0124' EB          XCHG                    ; DE=BYTES LEFT TO READ
0125' CD 0188'    ..CLOP: CALL     TAPGET
0128' 77          MOV      M,A
0129' 23          INX      H
012A' 1B          DCX      D
012B' 7A          MOV      A,D
012C' B3          ORA      E
012D' 20F6        JRNZ     ..CLOP
012F' D1          POP      D
0130' CD 0000:0B CALL     INCUSE
0133' FD6602      MOV      H,2(Y)
0136' FD6E01      MOV      L,1(Y)
0139' 360A        MVI      M,$NAMADR
013B' 23          INX      H
013C' 23          INX      H
013D' 3600        MVI      M,0
013F' 01 0003    LXI      B,$NVALUE-2
0142' 09          DAD      B
0143' 73          MOV      M,E
0144' 23          INX      H
0145' 72          MOV      M,D
0146' CD 0000:0A ..SKIP: CALL     INC5IY
0149' FD7E00      MOV      A,0(Y)
014C' FE00        CPI      $TAF
014E' 2098        JRNZ     ..LOOP
0150' CD 01E5'    CALL     SHUTOF
0153' FDE1        POP      Y
0155' E1          POP      H
0156' C9          RET
0157'              ..GETX:
                    ; SUBROUTINE TO CONVERT ALLOCATOR SIZE CODE IN BYTE COUN
                    ; T IN BC
                    ; IN A=ALLOC SIZE CODE
                    ; OUT BC=# BYTES IN BLOCK
0157' E5          SIZBC: PUSH     H
0158' 6F          MOV      L,A
0159' 2600        MVI      H,0
015B' 23          INX      H
015C' 29          DAD      H
015D' 29          DAD      H
015E' 29          DAD      H
015F' 29          DAD      H
0160' 44          MOV      B,H
0161' 4D          MOV      C,L
0162' E1          POP      H
0163' C9          RET
                    ; NAME MATCH SUBROUTINE
0164' CD 0188'    NAMATH: CALL     TAPGET
0167' FE A5       CPI      FNCHAR
0169' 20F9        JRNZ     NAMATH
                    ; WE GOT A NAME - POINT AT FIRST CHAR OF RAM NAME
016B' FD6602      MOV      H,2(Y)
016E' FD6E01      MOV      L,1(Y)
0171' 01 0009    LXI      B,$NASCII
    
```

```

0174' 09          DAD      B
0175' CD 0188'    ..MATC: CALL   TARGET      ; GOODIE
0178' BE          CMP      M
0179' 20E9       JRNZ     NAMATH      ; NILCHO
017B' 23          INX      H
017C' 7E          MOV      A,M
017D' A7          ANA      A
017E' 20F5       JRNZ     ..MATC
; IGNORE FURTHER SHIT
0180' CD 0188'    ..IGNO: CALL   TARGET
0183' FE0A       CPI      NL
0185' 20F9       JRNZ     ..IGNO
0187' C9          RET
; SUBROUTINE TO RETRIEVE CHARACTER
; FROM TAPE INPUT BUFFER
; PERFORM CONTROL C CHECK
0188' E5          TARGET: PUSH   H
0189' D5          PUSH   D
018A' CD 0000:07 ..CNTC: CALL   CNTCK
018D' ED5B 64B1   LDED     TAPPRO
0191' 2A 64AF     LHLD    TAPCON
0194' 7D          MOV      A,L
0195' BB          CMP      E
0196' 28F2       JRZ      ..CNTC
0198' 56          MOV      D,M
0199' CD 024B'    CALL    BUMPTR
019C' 22 64AF     SHLD    TAPCON
019F' 3A 67FF     LDA     KEYFLG
01A2' AA          XRA     D
01A3' E6C0       ANI     OCOH
01A5' AA          XRA     D
01A6' D324       OUT     LEDS
01A8' 7A          MOV      A,D
01A9' D1          POP     D
01AA' E1          POP     H
01AB' C9          RET
; SUBROUTINE TO ESTABLISH TAPE INPUT
01AC' F3          TINPES: DI
01AD' 21 64B3     LXI     H,TAPBUF
01B0' 22 64AF     SHLD    TAPCON
01B3' 22 64B1     SHLD    TAPPRO
; RESET THE UART
01B6' CD 01CF'    CALL    RESUART
01B9' 3E73       MVI     A,01110011B
01BB' 1806       JMPR    SETBS
; SUBROUTINE TO ESTABLISH TAPE OUTPUT
01BD' F3          TOUTEST: DI
01BE' CD 01CF'    CALL    RESUART
01C1' 3E72       MVI     A,01110010B
01C3' FDCB0146   SETBS: BIT    0,1(Y)
01C7' 2002       JRNZ     ..NOFL
01C9' EE07       XRI     00000111B
01CB' D323       ..NOFL: OUT    BAUDSEL
01CD' FB          EI
01CE' C9          RET
    
```

```

; ROUTINE TO RESET THE UART
RESUART:
01CF'      3EAD          MVI      A,0ADH
01CF'      D32B          OUT      AUARTS
01D1'      3E40          MVI      A,40H
01D3'      D32B          OUT      AUARTS
01D5'      D32B          OUT      AUARTS
01D7'      3ECD          MVI      A,0CDH
01D9'      D32B          OUT      AUARTS
01DB'      3E37          MVI      A,37H
01DD'      D32B          OUT      AUARTS
01DF'      3E01          MVI      A,1
01E1'      32 649C      STA      UARTFL
01E4'      C9           RET

; SUBROUTINE TO SHUT OFF MOTORS
SHUTOFF:
01E5'      3E38          MVI      A,00111000B
01E7'      D323          OUT      BAUDSEL
01E9'      AF           XRA      A
01EA'      32 649C      STA      UARTFL
01ED'      3EAE          MVI      A,0AEH
01EF'      D32B          OUT      AUARTS
01F1'      3E40          MVI      A,40H
01F3'      D32B          OUT      AUARTS
01F5'      3ECE          MVI      A,0CEH
01F7'      D32B          OUT      AUARTS
01F9'      3E27          MVI      A,27H
01FB'      D32B          OUT      AUARTS
01FD'      C9           RET

; SUBROUTINE TO WRITE LEADER/TRAILER
WRTLDR:
01FE'      E5           PUSH     H
01FF'      F5           PUSH     PSW
0200'      21 7530      LXI     H,30000
0203'      2B          ..WAIT: DCX     H
0204'      7C          MOV     A,H
0205'      B5          ORA     L
0206'      20FB        JRNZ    ..WAIT
0208'      F1          POP     PSW
0209'      E1          POP     H
020A'      C9           RET

; SUBROUTINE TO OUTPUT CHARACTER THRU UART
OUTTAP:
020B'      F5           PUSH     PSW
020C'      DB2B        ..OUT1: IN      AUARTS
020E'      E601        ANI     TXRDY
0210'      28FA        JRZ     ..OUT1
0212'      F1          POP     PSW
0213'      D32A        OUT     AUARTD
0215'      C9           RET

; SUBROUTINE TO OUTPUT FILENAME HEADER
OUTNAM:
0216'      3EA5        MVI     A, FNCHAR
0218'      CD 020B'    CALL    OUTTAP
021B'      01 0009      LXI     B, $NASCII
021E'      09          DAD     B
021F'      7E          ..OUTL: MOV     A, M
0220'      A7          ANA     A
0221'      2806        JRZ     ..DONE
0223'      CD 020B'    CALL    OUTTAP
    
```



```

0226' 23          INX      H
0227' 18F6        JMPR     ..OUTL
0229' 3E0A        ..DONE: MVI     A,NL
022B' CD 020B'    CALL    OUTTAP
022E' C9          RET

; UART CHARACTER INPUT ROUTINE
022F' 08          UARTINT: EXAF
0230' D9          EXX
0231' 2A 64B1     LHLD    TAPPRO
0234' 44          MOV     B,H
0235' 4D          MOV     C,L
0236' CD 024B'    CALL    BUMPTR
0239' ED5B 64AF   LD     TAPCON
023D' 7D          MOV     A,L
023E' BB          CMP     E
023F' DB2A        IN      AUARTD
0241' 2804        JRZ     ..FULL
0243' 02          STAX   B
0244' 22 64B1     SHLD   TAPPRO
0247' D9          ..FULL: EXX
0248' 08          EXAF
0249' FB          EI
024A' C9          RET

; SUBROUTINE TO INCREMENT POINTER
024B' 23          BUMPTR: INX     H
024C' 7D          MOV     A,L
024D' FE33        CPI     TBFEND&OFFH
024F' C0          RNZ
0250' 21 64B3     LXI     H,TAPBUF
0253' C9          RET

]

; .IFDEF DSKSHIT,C
; GET COMMAND
; THIS COMMAND RETRIEVES A STRING FROM DISC
;
; FOR EACH NAME:
; BUILD FCB AND OPEN
; GET LARGEST POSSIBLE MEMORY BLOCK
; READ UNTIL EOF
; RETURN REMAINING MEMORY
M.CMD[GETDSK,..GETX,0,AFIX[G],..GETA,..GET1]
; .BYTE 0
..GET1: CALL     ARG
..GETA: .BYTE   $NAME,$REPEAT ; GET STRING,STRING,...
        PUSH    H
        PUSH    Y
        CALL    CPMON
; BUILD FCB AND OPEN FILE
..LOOP: CALL    FCBBUILD
        CALL    FCBOPEN
; ALLOCATE MEMORY FOR STRING
        CALL    ALSTR ; HL=TOP,DE=FIRST,BC=#B'
TES
; LOOP TO COPY FROM INPUT BUFFER TO ALLOCATED STRING
..GETB: CALL    DINBYT
    
```

```

        CPI        0DH                ; FILTER OUT CARRIAGE RE
TURNS   JRZ        ..GETB
        CPI        'Z'-40H           ; END OF FILE
        JRZ        ..GEOF
        STAX       D
        DCX       B                ; DECREMENT BYTES LEFT
        INX       D                ; ADVANCE POINTER
        MOV       A,B                ; OUT OF MEM?
        ORA       C
        JRNZ      ..GETB            ; NO
        JMP       DSKERR
; DONE WITH FILE - STUFF NULL AND FREE REMAINING BYTES
..GEOF: CALL       ENDSTR
        CALL      INCUSE
        MOV       H,2(Y)            ; POINT NAME AT US
        MOV       L,1(Y)
        MVI      M,$NAMADR         ; MAKE IT A NAME
        INX      H
        INX      H
        MVI      M,0                ; CLEAR FLAGS
        LXI      B,$NVALUE-2
        DAD      B
        MOV      M,E
        INX      H
        MOV      M,D
        CALL     FCBCLOSE           ; CLOSE OUT THE FILE
        CALL     INC5IY
        MOV      A,0(Y)            ; MORE NAMES?
        CPI      $TAF
        JRNZ     ..LOOP            ; YES
        CALL     CPMOFF
        POP      Y
        POP      H
        RET

..GETX:
; PUTDSK COMMAND
; THIS ROUTINE PUTS OUT STRINGS ON THE DISC
;
PUTTAPE:
M.CMD[PUTDSK,..PUTX,0,PI,..PUTA,..PUTS]
        .BYTE    0
..PUTS: CALL     ARG
..PUTA: .BYTE    $NAME,$REPEAT
        PUSH     H
        PUSH     Y
        CALL     CPMON
..LOOP:
        CALL     FCBBUILD
        CALL     FCBDELETE
        CALL     FCBCREATE
        MOV     L,1(Y)
        MOV     H,2(Y)
        MOV     A,M
        CPI     $NULL
    
```

```

        JRZ     ..FUBB
        CALL    NXTVAL
        MOV     A,M
        CPI     $STRADR
        JRNZ    ..FUBB
; OK - DO IT
        LXI     B,$SASCII
        DAD     B
..CLOP: MOV     A,M
        ANA     A
        JRZ     ..CDON
        CALL    DOUTBYTE
        INX     H
        JMPR    ..CLOP
..CDON:
..FUBB: MVI     A,'Z'-40H
        CALL    DOUTBYTE
        CALL    FCBFLUSH
        CALL    FCBCLOSE
        CALL    INCSIY
        MOV     A,0(Y)
        CPI     $TAF
        JRNZ    ..LOOP
        CALL    CPMOFF
        POP     Y
        POP     H
        RET
..PUTX:
; DELETE ROUTINE
FCBDELETE:
        MVI     C,CPDELETE
        JMP     BDFCB
FCBCREATE:
        MVI     C,CPCREATE
        CALL    BDFCB
        INR     A
        JZ      DSKERR
        JMPR    RESBUF
; FLUSHER
FCBFLUSH:
        LDA     DBUFPT
        CPI     80H
        RZ
; SUBROUTINE TO WRITE OUT A SECTOR
OUTSEC: MVI     C,CPWRITE
        CALL    BDFCB
        ANA     A
        JNZ     DSKERR
RESBUF: MVI     A,80H
        STA     DBUFPT
        RET
; ROUTINE TO OUTPUT BYTE
DOUTBYTE:
        CPI     NL
        JRNZ    ..DOUT
; STICK IN CR CHARS
    
```

```

        MVI    A,0DH
        CALL   ..DOUT
        MVI    A,NL
..DOUT: PUSH   H
        MOV   H,A
        LDA   DBUFPT
        MOV   L,A
        MOV   A,H
        MVI   H,0
        MOV   M,A
        MOV   A,L
        INR   A
        STA   DBUFPT
        CZ    OUTSEC
        POP   H
        RET

; TURN ON CPM
CPMON:  DI
        MVI   A,10H
        OUT   0F8H
        MVI   A,2
        OUT   0FFH
        EI
        RET

;
CPMOFF: DI
        MVI   A,11H
        OUT   0F8H
        MVI   A,3
        OUT   0FFH
        NOP
        NOP
        NOP
        NOP
        NOP

UARTINT: EI
        RET

; ERROR OUT!
DSKERR: MVI   A,55H
        OUT   LEDS
        CALL  CPMOFF
        ERROR ER.DSK
        RET

; SUBROUTINE TO FETCH CHARACTER FROM DISC
; USES BUFFER AT 80H
DINBYT: PUSH   H
        LDA   DBUFPT
        ANA   A ; NONZERO?
        JRNZ  ..NOTZ
; POINTER IS ZERO - GET ANOTHER DISC BUFFER
        PUSH  B
        PUSH  D
    
```

TAPEID -

```

        LXI    D,80H
        MVI    C,26
        CALL   BDOSFU
        LXI    D,FCBADR
        MVI    C,CPREAD
        CALL   BDOSFU
        ANA    A
        JNZ    DSKERR
        POP    D
        POP    B
        MVI    A,80H
..NOTZ: MOV    L,A                ; POINT AT BYTE IN BUFFE
R                                           ;
        INR    A                ; UPDATE BUFFER POINTER
        STA    DBUFPT
        MVI    H,0
        MOV    A,M                ; A = CHAR
        POP    H
        RET
; FCBBUILD
; CONSTRUCT FCB GRABBING FILENAME FROM NAME ON IY STACK
; EXTENSION USED IS .ZGR
FCBBUILD: MOV    H,2(Y)          ; HL=PTR TO NAME

        MOV    L,1(Y)
        LXI    B,$NASCII        ; HL=FIRST CHAR OF NAME
        DAD    B
        LXI    D,FCBADR        ; DE = FCB
        XRA    A
        STAX   D
        INX    D
        MVI    B,8
; COPY FILE NAME OVER
..MORE: MOV    A,M                ; GET CHAR
        ANA    A                ; IS IT A NULL?
        JRZ    ..NULL
        STAX   D                ; NO - STUFF IT
        INX    D
        INX    H
        DJNZ   ..MORE
        JMPR   ..FULL
; PARTIAL NAME - PAD WITH BLANKS
..NULL: MVI    A,BLANK
..BLNK: STAX   D
        INX    D
        DJNZ   ..BLNK
;
..FULL: MVI    A,'Z'
        STAX   D
        INX    D
        MVI    A,'G'
        STAX   D
        INX    D
        MVI    A,'R'
        STAX   D

```

TAPEIO -

```

        INX      D
; FILL REMAINING AREA WITH ZEROS
        MVI     B,21
        XRA     A
..CLR:  STAX    D
        INX     D
        DJNZ   ..CLR
        RET

; SUBROUTINE TO OPEN EXISTING FILE FOR INPUT
FCBOPEN:  PUSH   B
        PUSH   D
        PUSH   H
        LXI   D,FCBADR
        MVI   C,CPOPEN
        CALL  BDOSFU
        INR   A
        JZ    DSKERR
; OPEN WAS GOOD - RESET BUFFER SHIT
        XRA   A
        STA  DBUFPT
        POP  H
        POP  D
        POP  B
        RET

; SUBROUTINE TO CLOSE EXISTING FILE
FCBCLOSE:  PUSH  H
        PUSH  D
        PUSH  B
        LXI  D,FCBADR
        MVI  C,CPCLOS
        CALL BDOSFU
        POP  B
        POP  D
        POP  H
        RET

BDFCB:
        LXI  D,FCBADR

BDOSFU:
        DI
        IMO
        MOV  A,C
        OUT  LEDS
        XRA  A
        OUT  OFFH
        MVI  A,4
        OUT  OF1H
        MVI  A,05H
        OUT  INMOD
..NSEX:  EI
        CALL BDOS
        DI
        STA  POINTER      ; ** TEST **
        XRA  A
        OUT  OF1H
        MVI  A,8

```

```
        OUT      INMOD
        IM2
        MVI      A,2
        OUT      OFFH
        LDA      POINTER
        OUT      LEDS
..ASEX: EI
..BSEX:
        RET
]
        .END
```

TAPEIO -

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

AASN	005F	ALLOC	0000:04 X	ALSTR	0000:05 X	ARG	0000:06 X
ARGSTK	60CC	AUARDT	002A	AUARTS	002B	BACKGR	65CC
BAUDSE	0023	BDOS	E106	BLANK	0020	BOTRAM	6000
BOTTOM	64A9	BSAUD	0022	BUMPTR	024B	CHARSL	65DD
CLEAR5	60CA	CNTCK	0000:07 X	CNTRL	000C	CNTRLC	65CD
CNTRLO	65D9	CNTRLU	0015	CNTRLZ	65E4	CPCLOSE	0010
CPCREA	0016	CPDELE	0013	CPLARE	611C	CPLSIZ	0140
CPOPEN	000F	CPREAD	0014	CPWRIT	0015	CR	000D
CSBLOK	668D	CSFLAG	6260	CURCX	6814	CURCY	6816
CURREN	64A5	C.CO	0011	C.C1	0012	C.CX	000B
C.CY	000D	C.DP	0013	C.ST	0002	C.X	0003
C.XF	000F	C.XS	0007	C.Y	0005	C.YF	0010
C.YS	0009	DBUFPT	0028	DDTON	65CE	DEVBL	6589
DEVCL0	6579	DEVCL1	657B	DEVCL2	657D	DEVCL3	657F
DEVCL4	6581	DEVCL5	6583	DEVCL6	6585	DEVCL7	6587
DEVFB	65CB	DEVHCB	658F	DEVMO	658B	DEVNM	65B7
DEVNT	658D	DEVTNA	65BB	DEVTNB	65BF	DEVTNC	65C3
DEVVA	65BD	DEVVAR	6579	DEVVB	65C1	DEVVBL	6591
DEVVC	65C5	DEVVD	65C9	DEVVN	65B9	DEVVS	65C7
DEVXCD	65B3	DEVYCD	65B5	DOLPLH	62E8	DOLPPT	62EA
DUMBST	6577	EDBCNT	64AD	EDCCNT	64A7	EDLONG	6812
EDMODE	681C	EDNAME	64A1	EDNCX	680C	EDNCY	600E
EDNEWS	64A5	EDOCX	6808	EDOCY	680A	EDPN	6806
EDPO	6804	EDPTRC	64AB	EDPTRL	64A9	EDSTR	681A
ENDSTR	0000:08 X	ERABIT	0002	ERRPGM	0000:09 X	ER.ARA	002F
ER.ARG	0034	ER.ASN	0015	ER.BOX	001A	ER.CHN	0002
ER.CMD	001F	ER.CNV	0016	ER.COR	001B	ER.CTL	0036
ER.DEL	0026	ER.DIM	0030	ER.DIV	0018	ER.DP	0037
ER.DSK	0019	ER.EDT	0035	ER.FMT	0038	ER.FNF	001C
ER.FOR	0028	ER.IMP	0003	ER.LAB	0025	ER.MAC	0022
ER.NAE	002D	ER.NAM	0029	ER.NEG	003A	ER.NOT	0023
ER.NUL	0039	ER.NUM	002B	ER.NXT	001D	ER.OFL	0017
ER.OPN	0014	ER.OVE	001E	ER.PAR	002A	ER.REN	002C
ER.RET	0024	ER.SEP	0021	ER.SNP	0031	ER.SPC	002E
ER.STK	0004	ER.SW	0032	ER.TER	0020	ER.UFL	0033
ER.UNF	0027	EXTDEL	002E	E.HVAL	0002	E.LVAL	0001
E.SIZ	0005	E.TYP	0000	E.VAL	0001	FCBADR	005C
FCNTH	65D2	FCNTI	65D3	FCNTJ	65D4	FCNTK	65D5
FCNTL	65D6	FCNTV	65E0	FCNTY	65E3	FIRST	64A3
FLAGS	65CB	FNCHAR	00A5	FOREGR	65D0	FRAGSI	0400
FREELS	65E5	FSTDOL	648C	FSTINT	62EC	FWDPTR	65E7
GETDSK	005E	GETTAP	00C7	GOTTAP	005E	HCAREA	6593
HORCB	0009	INC5IY	0000:0A X	INCRO	65E9	INCUSE	0000:0B X
INFBK	000D	INLIN	000F	INMOD	000E	JUNK	6542
KBLOCK	65F1	KEYFLG	67FF	KEYPTK	6533	KEYTRK	67F7
LEDS	0024	LF	000A	LISTON	65E2	MACSTU	6536
MACTOP	625E	MAGIC	000C	MAXFRG	0040	MNMX	65EB
MPBAV	0016	MPMO	0010	MPNCV	0015	MPNV	0017
MPTNA	0011	MPTNB	0012	MPTNC	0013	MPVIB	0014
NAMATH	0164	NBLKB	0000	NBLKM	0001	NEWBOT	6810
NL	000A	NORML	0020	NSADDR	6802	NUMBUF	64A3
NXTVAL	0000:0C X	OLDCHR	64A0	OLDCUR	649F	OLDKEY	649D
OLDXY	65EF	ONEBUF	6729	OPRL	0014	OPRSP	655B
OPRSTK	6547	OPRSZ	655D	OUTNAM	0216	OUTOFF	62E7
OUTTAP	020B	PCNT	655E	PI	0000:0D X	PIXVAL	65ED