

ADDRESS LENGTH
0 746
746

BINARY CONTROL CARDS.

IDENT PRINT
END

ENTRY POINTS.

PRINTING - 211 SEGERR - 633 LINE - 213 SLOPPY - 347
SLOPPY - 353 ENDRUN - 742 STATUS - 554

EXTERNAL SYMBOLS.

BAD.EVN	FUNNY.C	DESTOR	WRITEF	C.OPERC	DETHAD	CONVNUM	PDPBUFF
BUFFSIZ	F.RETUR	SAVE	BCOUNT	C.LINEA	ATTACH	TIMSG	MSGNAME
EBIT	GETPP	SIZE	BEAD	C.OWNFI	DETACH	PARAM	PRNMFLG
EJECTF	LBUFF	STOPE	CALLPP	C.PPCAL	DETHCT	TYPEIN	TYPEOUT
EVNTDAT	NBUFF	S.RETUR	FILEBUF	C.CPCAL	ATCHCT	TTYBUFF	PDPMSG
FADDR	NEXTBUF	TEMP	FLEN	C.PPBUF	EXIT2	LINECT	PDPFLG
FRMTSUR	NPOS	TITLECT	C.BEAD	C.FILE	EC:TIME	PDPRESP	MYID
FRMT.CL	PPBUFF	TITLEFM	C.SELF	C.FILE2	TITLE	PDPREST	LOG2SIZ
FTEMP	READF	WCNT	C.PRINT	ATCHAD	DISASCI	TIMEOUT	

```

1  DISKSYS  IDENT EQU PRINT
*
*
*  ASCII    MACRO    STRING
*            DIS      .,$$STRING,$$
*            ENDM
*
*  NM       MACRO    INDEX,CAP,X
*            MICRO    J,7,$$CAP,$$
*            EXT      *NM*
*            ENDM
*
*
*
*
*  EXT      BAD,EVN,BUFFSIZ,ERIT,EJECTFL,EVNTDAT,FADDR
*  EXT      FRMTSUP,FRMT,CL,FTMP,FUNNY,C,F,RETUR,GETBP
*  EXT      I,BUFF,NBUFF,NEXTBUF,NPOS,PPBUFF,READF,RESTOR,SAVE
*  EXT      SIZE,STORE,S,RETUR,TEMP,TITLECT,TITLEFM,WDCNT
*  EXT      WRITEF,BCOUNT,READ,CALLPP,FILEBUF,FLEN
*
*
*
*  C.READ   INDEX    1
*  C.SELF   INDEX    17
*  C.PRINT  INDEX    26
*  C.OPERCL INDEX
*  C.LINE$C INDEX
*  C.OWNFIL INDEX
*  C.PPCALL INDEX
*  C.CPCALL INDEX
*  C.PPBUFF INDEX
*  C.FILE   INDEX
*  C.FILE$  INDEX

```

12	↓0 EQU 12B
1	↓1 EQU 1B
2	↓2 EQU 2B
3	↓3 EQU 3B
4	↓4 EQU 4B
5	↓5 EQU 5B
6	↓6 EQU 6B
7	↓7 EQU 7B
10	↓8 EQU 10B
11	↓9 EQU 11B
0	↓COLON EQU 00B
13	↓EQUAL EQU 13B
14	↓NEQUAL EQU 14B
15	↓LESSEQU EQU 15B
16	↓PERCNT EQU 16B
17	↓LBRAK EQU 17B
60	↓PLUS EQU 60B
61	↓A EQU 61B
62	↓B EQU 62B
63	↓C EQU 63B
64	↓D EQU 64B
65	↓E EQU 65B
66	↓F EQU 66B
67	↓G EQU 67B
70	↓H EQU 70B
71	↓I EQU 71B
72	↓LESS EQU 72B
73	↓PERIOD EQU 73B
74	↓RPAREN EQU 74B
75	↓GTEQU EQU 75B
76	↓NOT EQU 76B
77	↓SEMICOL EQU 77B
40	↓MINUS EQU 40B
41	↓J EQU 41B
42	↓K EQU 42B
43	↓L EQU 43B
44	↓M EQU 44B
45	↓N EQU 45B
46	↓O EQU 46B
47	↓P EQU 47B
50	↓Q EQU 50B
51	↓R EQU 51B
52	↓OR EQU 52B
53	↓DLRSGN EQU 53B
54	↓STAR EQU 54B
55	↓UPAROW EQU 55B
56	↓DNAROW EQU 56B
57	↓GRTR EQU 57B
20	↓BLANK EQU 20B
21	↓SLASH EQU 21B
22	↓S EQU 22B
23	↓T EQU 23B
24	↓U EQU 24B

```

25  ↓V EQU 25B
26  ↓W EQU 26B
27  ↓X EQU 27B
30  ↓Y EQU 30B
31  ↓Z EQU 31B
32  ↓RBRK EQU 32B
33  ↓COMMA EQU 33B
34  ↓LPAREN EQU 34B
35  ↓RTAROW EQU 35B
36  ↓IDENT EQU 36B
37  ↓AND EQU 37B
    ASC2 MACRO A,B,C
      VFD 1/A,41/B,18/C
    ENDM
0  * ASCITAB EQU *L
0  0000000000000000000020 .BLANK ASC2 0,0,↓BLANK
1  0000000000000000000015 .EXCLAM ASC2 0,0,↓ESSEQU
2  0000000000000000000056 .DBQOT ASC2 0,0,↓DNAROW
3  0000000000000000000036 .NUMSGN ASC2 0,0,↓IDENT
4  0000000000000000000053 .DLRSGN ASC2 0,0,↓DIRSGN
5  0000000000000000000016 .PERCNT ASC2 0,0,↓PERCNT
6  0000000000000000000037 .AMPRSD ASC2 0,0,↓AND
7  0000000000000000000014 .SNQUOT ASC2 0,0,↓NEQUAL
10 0000000000000000000034 .LPAREN ASC2 0,0,↓LPAREN
11 0000000000000000000074 .RPAREN ASC2 0,0,↓RPAREN
12 0000000000000000000054 .STAR ASC2 0,0,↓STAR
13 4000000000000000000060 .PLUS ASC2 1,↑OVER,↓PLUS
14 0000000000000000000033 .COMMA ASC2 0,0,↓COMMA
15 4000000000000000000040 .MINUS ASC2 1,↑TRIDL,↓MINUS
16 0000000000000000000073 .DOT ASC2 0,0,↓PERIOD
17 0000000000000000000021 .SLASH ASC2 0,0,↓SLASH
20 4000000000000000000012 .0 ASC2 1,↑DOUBL,↓0
21 4000000000000000000001 .1 ASC2 1,↑EJECT,↓1
22 4000000000000000000002 .2 ASC2 1,↑HALF,↓2
23 4000000000000000000003 .3 ASC2 1,↑THIRD,↓3
24 4000000000000000000004 .4 ASC2 1,↑QTR,↓4
25 0000000000000000000005 .5 ASC2 0,0,↓5
26 4000000000000000000006 .6 ASC2 1,↑CONCAV,↓6
27 4000000000000000000007 .7 ASC2 1,↑CONVEX,↓7
30 4000000000000000000010 .8 ASC2 1,↑TOP,↓8
31 4000000000000000000011 .9 ASC2 1,↑BOTTOM,↓9
32 0000000000000000000000 .COLON ASC2 0,0,↓COLON
33 0000000000000000000077 .SEMICOL ASC2 0,0,↓SEMICOL
34 0000000000000000000072 .LDSS ASC2 0,0,↓LESS
35 0000000000000000000013 .EQUAL ASC2 0,0,↓EQUAL
36 0000000000000000000057 .FTRTHN ASC2 0,0,↓GRTR
37 0000000000000000000075 .QUEST ASC2 0,0,↓GREQU
40 0000000000000000000076 .AT ASC2 0,0,↓NOT
41 0000000000000000000061 .A ASC2 0,0,↓A
42 0000000000000000000062 .B ASC2 0,0,↓B
43 0000000000000000000063 .C ASC2 0,0,↓C
44 0000000000000000000064 .D ASC2 0,0,↓D
45 0000000000000000000065 .E ASC2 0,0,↓E

```

46	0000000000000000000066	.F	ASC2	0,0,↑F
47	0000000000000000000067	.G	ASC2	0,0,↑G
50	0000000000000000000070	.H	ASC2	0,0,↑H
51	0000000000000000000071	.I	ASC2	0,0,↑I
52	0000000000000000000041	.J	ASC2	0,0,↑J
53	0000000000000000000042	.K	ASC2	0,0,↑K
54	0000000000000000000043	.L	ASC2	0,0,↑L
55	0000000000000000000044	.M	ASC2	0,0,↑M
56	0000000000000000000045	.N	ASC2	0,0,↑N
57	0000000000000000000046	.O	ASC2	0,0,↑O
60	0000000000000000000047	.P	ASC2	0,0,↑P
61	0000000000000000000050	.Q	ASC2	0,0,↑Q
62	0000000000000000000051	.R	ASC2	0,0,↑R
63	0000000000000000000022	.S	ASC2	0,0,↑S
64	0000000000000000000023	.T	ASC2	0,0,↑T
65	0000000000000000000024	.U	ASC2	0,0,↑U
66	0000000000000000000025	.V	ASC2	0,0,↑V
67	0000000000000000000026	.W	ASC2	0,0,↑W
70	400000000000013000027	.X	ASC2	1,↑ENTRSUP,↑X
71	400000000000014000030	.Y	ASC2	1,↑LEAVSUP,↑Y
72	400000000000015000031	.Z	ASC2	1,↑SUPRES1,↑Z
73	0000000000000000000017	.LBRAK	ASC2	0,0,↑LBRAK
74	0000000000000000000052	.BKSLASH	ASC2	0,0,↑BKSLASH
75	0000000000000000000032	.RBRAK	ASC2	0,0,↑RBRAK
76	0000000000000000000055	.UPAROW	ASC2	0,0,↑UPAROW
77	0000000000000000000035	.BKAROW	ASC2	0,0,↑BKAROW
100	0000000000000000000014	.APOSTR	ASC2	0,0,↑NEQUAL
101	0000000000000000000061	.LCASEA	ASC2	0,0,↑A
102	0000000000000000000062	.LCASEB	ASC2	0,0,↑B
103	0000000000000000000063	.LCASEC	ASC2	0,0,↑C
104	0000000000000000000064	.LCASED	ASC2	0,0,↑D
105	0000000000000000000065	.LCASEE	ASC2	0,0,↑E
106	0000000000000000000066	.LCASEF	ASC2	0,0,↑F
107	0000000000000000000067	.LCASEG	ASC2	0,0,↑G
110	0000000000000000000070	.LCASEH	ASC2	0,0,↑H
111	0000000000000000000071	.LCASEI	ASC2	0,0,↑I
112	0000000000000000000041	.LCASEJ	ASC2	0,0,↑J
113	0000000000000000000042	.LCASEK	ASC2	0,0,↑K
114	0000000000000000000043	.LCASEL	ASC2	0,0,↑L
115	0000000000000000000044	.LCASEM	ASC2	0,0,↑M
116	0000000000000000000045	.LCASEN	ASC2	0,0,↑N
117	0000000000000000000046	.LCASEO	ASC2	0,0,↑O
120	0000000000000000000047	.LCASEP	ASC2	0,0,↑P
121	0000000000000000000050	.LCASEQ	ASC2	0,0,↑Q
122	0000000000000000000051	.LCASER	ASC2	0,0,↑R
123	0000000000000000000022	.LCASES	ASC2	0,0,↑S
124	0000000000000000000023	.LCASET	ASC2	0,0,↑T
125	0000000000000000000024	.LCASEU	ASC2	0,0,↑U
126	0000000000000000000025	.LCASEV	ASC2	0,0,↑V
127	0000000000000000000026	.LCASEW	ASC2	0,0,↑W
130	400000000000013000027	.LCASEX	ASC2	1,↑ENTRSUP,↑X
131	400000000000014000030	.LCASEY	ASC2	1,↑LEAVSUP,↑Y
132	400000000000015000031	.LCASEZ	ASC2	1,↑SUPRES1,↑Z

PRINTER DRIVER MAIN LOOPS
ASCII - CDC TABLE

COMPASS - VER 2.

08/19/71 17.20.17.

PAGE 6

133	00000000000000000000000074	.RSQUIG	ASC2	0:0:*	R PAREN
134	00000000000000000000000052	.VERTBAR	ASC2	0:0:*	R OR
135	00000000000000000000000034	.LSQUIG	ASC2	0:0:*	R PAREN
		LIST			

```

*
*
*   FORMAT FUNCTION TABLE
*
*
0  *EJECT EQU 0
1  *DOUBL EQU 1
2  *TRIPL EQU 2
3  *OVER EQU 3
4  *HALF EQU 4
5  *THIRD EQU 5
6  *QTR EQU 6
7  *CONCAV EQU 7
10 *CONVEX EQU 8
11 *TOP EQU 9
12 *BOTTOM EQU 10
13 *ENTRSUP EQU 11
14 *LEAVSUP EQU 12
15 *SUPRES1 EQU 13
*
136 * FRMTABL EQU *L
*
*
FCNS      MACRO      A,B,C,D,E
          BSS
          VFD      12/A,12/B,12/C,12/D,12/E
FCNS      ENDM
*
*
136      FCNS      10B,20B,22B,10B      * EJECT
137      FCNS      10B,20B,23B,10B      * DOUBLE
140      FCNS      10B,20B,24B,10B      * TRIPLE
141      FCNS      10B,20B,25B,10B      * OVER DOESN'T WORK
142      FCNS      10B,20B,26B,10B      * HALF
143      FCNS      10B,20B,27B,10B      * THIRD
144      FCNS      10B,20B,28B,10B      * QUARTER
145      FCNS      10B,20B,29B,10B      * CONCAVE
146      FCNS      10B,20B,30B,10B      * CONVEX
147      FCNS      10B,20B,31B,10B      * TOP
150      FCNS      10B      * LAST
151      FCNS      10B      * SUPPRESS
152      FCNS      10B      * LEAVE SUPPRESS
153      FCNS      10B      * SUPPRESS ONE LINE
*
154      BSSZ      14=*L+FRMTABL

```

```

154 414143345511114240522 ER.INTRJ ASCII (6681 INTERNAL REJECT)
157 41414334550530240522 ER.EXTRJ ASCII (6681 EXTERNAL REJECT)
162 41414334552422011623 ER.PAR ASCII (6681 TRANSMISSION PARITY ERROR)
166 20012005225517252455 ER.PAPER ASCII (PAPER OUT ON PRINTER)
171 20221116240522551617 ER.NTRDY ASCII (PRINTER NOT READY)
173 10012204270122055505 ER.UNKNO ASCII (HARDWARE ERROR)
175 05260516245503100116 ER.HOWBD ASCII (EVENT CHANNELS SCREWED UP)
200 05260516245522051205 ER.BRUCE ASCII (EVENT REJECTED $$$)
202 22050317260522315516 CANTFIX ASCII (RECOVERY NOT POSSIBLE)
205 010217222400000000000 ABORT ASCII (ABORT)
206 20221116240522551417 SEGERRM ASCII (PRINTER LOCK ERROR, SORRY$$$)
211 20221116241116075557 PRNTING ASCII (PRINTING ...)
      ENTRY BRNTING
*
*
MESOUT MACRO WHAT
      SX2 WHAT
      EQ STATMES
MESOUT ENDM
*
TESTCHAR MACRO CHAR, MASK, USE, FOUND, NTFFOUND
      MX, MASK 0-7
      LX, MASK 8
      SA, USE 1 BUFF
      BX, USE 1-X, MASK*X, USE
      AX, USE 9
      SX, USE Y, USE=ASCII*CHAR
      IF NE, **FOUND*, 1
      ZR Y, USE, FOUND
      IFC NE, **NTFOUND*, 1
      NZ Y, USE, NTFFOUND
TESTCHAR ENDM
*
*
*
LINEIN MACRO W
      SX7 1
      SA7 1 INCALL*1
      IF 1 DEF, LINCALL
      EXT 1 INCALL
      ENDF
      CEJ 1 INCALL..., W-1, 208
LINEIN ENDM
*
LINEOUT MACRO W
      SX7 100
      SA7 W-1
      SX7 W-1
      SX6 208
      LX6 30
      BX7 X6+X7
      SA7 1 INCALL*2
      SX7 2

```



```

SA7      LINCALL+1
IF        DEF,LINCALL
EXT       LINCALL
ENDIF
CEJ       LINCALL
ENDM
LINEOUT   JIMGREY
MACRO     A,WHAT
LOCAL    RET
CEJ       SAVE
SB7       WHAT
IN        IF
          EQ,*A*IN*,1
          WHAT
          IF
          EQ,$ASOUTS
          RET
          PDPTST
          EQ,XDPMSG,1
          EQ,XTYPEOUT
          ENDIF
          RET
          CEJ
          ENDM
*
PDPTST    MACRO
          SA,REG
          SA,REG
          NZ
          ENDM
*
CEJ       MACRO
          WT,CT,JP,DATA,DCT,CAPS,CCT
*
          XJ
          WT
*
DAT       IF
          NE,**DATA*
          RTLST
          CT,JP,DATA,DCT,CAPS,CCT
          ENDF
*
CAP       IF
          NE,**CAPS*
          IF
          EQ,**DATA**,1
          RTLST
          CT,JP,DATA,DCT,CAPS,CCT
*
CAP       ELSE
CAP       IF
          EQ,**DATA*
          VFD
          30/CT
          IF
          NE,**JP*,2
          EQ
          JP
          BSS
          0
*
CAP       ENDIF
*
CEJ       ENDM
*
*
*

```

```

RTLST  MACRO      CT,JP,DATA,DCT,CAPS,CCT
CT      IFC      EQ,**CT*
-      VFD      30/1000003B
CT      ELSE
-      VFD      30/1000000B+CT
CT      ENDIF
*
JP      IFC      NE,**JP*
+      EQ      JP
JP      ELSE
+      EQ      **3
JP      ENDIF
*
-      VFD      30/**+1
*
DT      IFC      NE,**DATA*
DC      IFC      NE,**DCT*
+      VFD      30/DCT
DC      ELSE
+      VFD      30/1
DC      ENDIF
-      VFD      30/DATA
DT      ELSE
+      VFD      30/0
DT      ENDIF
*
*
CP      IFC      NE,**CAPS*
CTC     IFC      NE,**CCT*
+      VFD      30/CCT
CTC     ELSE
+      VFD      30/1
CTC     ENDIF
-      VFD      30/CAPS
CP      ELSE
+      VFD      30/0
CP      ENDIF
*
RTLST  ENDM
*
*
*      MACROS
*
PUTCHAR MACRO      A,L
LOCAL   Z
*
LX3     B3,X,A      * SHIFT CHARACTER
BX7     X7+X3      * ADD
SB3     B3-6        * DECREMENT SHIFT COUNT
SB5     B5+1        * INCREMENT CHARACTER COUNT
GE      B3,B0,Z     * JUMP IF ROOM LEFT IN OUTPUT WORD
SB3     B0-6        * RESET SHIFT COUNT

```

```

SA7      A7+1      * STORE WORD
SX7      0
SB7      1+2      * MAX CHAR COUNT
LE       B5,B7,Z  * LINE NOT FULL, JUMP
CALL     FBUFF
BSS      0
Z
*
PUTCHAR  ENDM
*
*
GETCHAR  MACRO     A,L
LOCAL   Z
*
BX,A     =X0*X1    * GET CHARACTER
LX1      7         * MOVE CHARACTERS
SB4      B4-1      * DECREMENT CHAR COUNT
NE       B4,B0,Z   * WORD NOT EMPTY, JUMP
CALL     NWORD     * GET NEW WORD
BSS      0         * DONE
Z
*
GETCHAR  ENDM
*
*
CALL     MACRO     WHAT,RETURN
LOCAL   X
IFC      EQ,**RETURN*
SB7      X
ELSE
SB7      RETURN
ENDIF
EQ       WHAT
BSS      0
X
CALL     ENDM
*
*
SAVEN    MACRO     L
LOCAL   Z,Y
*
SB3      B3-12     * LEAVE SPOT FOR N
GE       B3,B0,Y   * JUMP IF ROOM LEFT
SA7      A7+1      * STORE WORD
SB3      60-12
SX7      0
Y
SX6      B3        * SAVE SHIFT COUNT
SX2      A7+1      * NEXT LOCATION
LX2      18        * POSITION LOCATION
BX6      X6+X2     * POSITION AND SHIFT SAVE
SA6      NPOS      * SAVE INFO
SB3      B1-6      * NEXT THING IS A CHARACTER
GE       B3,B0,Z   * JUMP IF ROOM LEFT
SA7      A7+1      * STORE WORD
SB3      60-6      * RESET SHIFT COUNT
SX7      0         * ZERO OUTPUT WORD

```

```

Z      BSS      0      * DONE
*
SAVEN  ENDM
*
*
INSERTN MACRO      REG,L
*
      SA2      NPOS      * GET INFO
      SB7      X2        * SHIFT COUNT
      AX2      10        * ADDRESS
      SA2      X2        * WORD TO INSERT IN
      SX3      REG      * N ITSELF
      AX3      1         * CHAR COUNT / 2
      LX3      B7+X3     * POSITION N
      BX6      X2+X2     * N INSERTED
      SA6      A2        * STORE
*
INSERTN ENDM
*
*
*
GETBUFF MACRO      RETURN
      LOCAL
      SB2      B2+20B    * TRY NEXT LINE
      SA2      NBUFF     * START OF THIS HALF
      SX2      X2+SIZE   * END OF THIS HALF
      BX2      -X2
      SX2      X2+B2     * COMPARE
      NZ      X2,X      * JUMP IF ROOM LEFT
      CALL    CENDEV,RETURN * GET NEXT BUFFER HALF
*
X      BSS
GETBUFF ENDM
*
*
BREAKPT MACRO      ON
      IFC      NE,**ON*
      RJ
      ELSE
      NO
      NO
      NO
      NO
      ENDIF
      ENDM
*
*
CALLINE MACRO      WHAT,RET
      SB4      0
      SA1      WHAT
      SB7      RET
      LX1      40-49
      EQ
      LINE
CALLINE ENDM

```

```
*  
*  
*  
SEQCHECK MACRO REG  
MX6 60-2  
BX,REG X6*X.REG  
BX2 X6*X7  
SA3 XMYID  
BX3 X3-X2  
NZ X3,SEQERR  
ENDM
```

213 76770
5170000000 X
13777
214 5172777776

LINE

LINE IS THE MAIN TRANSLATION LOOP

SX7 B7 * RETURN
SA7 S RETUR * SAVE RETURN
BX7 X7-X7 * ZERO OUTPUT WORD
SA7 B2-1 * A7+1 = START OF NEXT LINE BUFFER

NOTE THAT THE ABOVE INSTRUCTION MAY CLOBBER SOMETHING

215 6130000060
6150000000
6110000000
216 5150000000 Y
0315000220 +
217 0400000263 +
220 6133777763
230 15210
233 7262777642
0326000244 +
234 5222000000 +
73220
22332
243 0400000230 +

LINE1
LINE2

LINE3

SB5 0 * ZERO CHARACTER COUNT
SB3 A0-12
SB1 0 * ZERO BLANK COUNT
SA5 FRMTSUP * PICK UP FORMAT FLAG
NZ X5+LINE1 * FORMAT CONTROL ON ZERO
EQ FRMT * CALL FRMT
SAVEN
GETCHAR 2 * GET CHARACTER
SX6 X2-135B * CHECK FOR LEGAL CHARACTER
PL X6+SPECIAL * JUMP IF SPECIAL CHARACTER
SA2 X2+ASCITAB * TABLE ENTRY
SX2 X2 * CHARACTER ONLY IN TABLE
PUTCHAR 2 * PUT CHARACTER IN OUTPUT BUFFER
EQ LINE2 * LOOP

244 7262777622
0306000666 +
245 7262777604
0306000247 +
246 7120000000
0406000234 +

SPECIAL

SX6 X2-155B
ZR X6+EOLCR * CHECK FOR CARRIAGE RETURN
SX6 X2-173B
ZR X6+BLNKS * CHECK FOR ESCAPE CHARACTER
SX2 0 * ELSE CHARACTER IS BLANK
EQ LINE3 * AND GO BACK

BLNKS EXPANDS COMPRESSED BLNKS

247 15210
252 63120
7120000020
253 0410000230 +
6111777776
254 22332
262 0400000253 +

BLNKS

BLNKS1

GETCHAR 2 * GET COUNT
SB1 X5 * COUNT IN R1
SX2 2AB * EXT BCD BLANK
EQ B1+B0,LINE2 * RETURN IF DONE
SB1 B1-1 * DECREMENT BLANK COUNT
PUTCHAR 2 * OUTPUT A BLANK
EQ BLNKS1 * LOOP

```

*
*
*
263 15210          FRMT  GETCHAR 2      * GET FORMAT CHARACTER
266 7262777642    SX6      X2=135B
                0336000275 + NG      X4,FRMT1 * GOOD CHA
267 7262777622    SX6      X2=155B
                0306000651 + ZR      X4,FOLF * CHECK FOR NULL LINE
270 7262777604    SX6      X2=173B * CHECK FOR ESCAPE
                7120000000 SX2      0      * CHAR IS BLANK ANYWAY
271 0316000275 +   NZ      X4,FRMT1 * NO COMPRESSED BLANKS
                15210      GETCHAR 2      * GET BLANK COUNT
274 6212777776    SB1      X2=1      * DECREMENT IT
                7120000000 SX2      0      * REPLACE BLANK YOU JUST CLOBBERED
275 5130000000 X   FRMT1  SA3      FRMT.CL * GET WAITING FUNCTION CODE
                0303000300 + ZR      X3,FRMT2 * NO CLEAR NEEDED
276 73430          SX4      X3      * SEND FUNCTION CODE
                13666      BX6      X4=X6
                54630      SA6      A3      * RESET CLEAR FLAG
277 6170000300 +   CALL      PUTBYTE * OUTPUT THE FUNCTION CODE
300 5222000000 +   FRMT2  SA2      X2+ASCITAB * TABLE ENTRY
                0322000332 + PL      X2,FDONE * HIGH BIT IS ACTION REQUIRED FLAG
301 43360          MX3      6A=12
                7160000000 SX6      0
302 5160000000 X   SA6      FRMT.CL * CLEAR WAITING FUNCTION CODE
                20252      LX2      6A=18 * SHIFT TABLE ENTRY TO GET FORMAT TABLE ENT
303 7242777762    SX4      Y2=+SUPRES1 * CHECK FOR SPECIAL CASE
                0314000305 + NZ      Y4,FRMT3
304 7160000005    SX6      * SELECT AUTO PAGE EJECT NEXT TIME
                5160000000 X SA6      FRMT.CL
305          FRMT3  BSS      *
306 7262000000    SX6      Y2+0
                5160000000 X SA6      FTEMP * SAVE TYPE OF CONTROL
306 5222000136 +   SA2      X2+FRMTABL * GET FORMAT TABLE ENTRY
                000005   FCNOUT DUP      5      * OUTPUT UP TO FIVE FUNCTION CODES
                LX2      12      * POSITION NEXT FUNCTION CODE
                BX4      -Y3*X2 * GET FUNCTION CODE IN X4
                ZR      X4,FRMT5 * QUIT ON ZERO CODE
                CALL      PUTBYTE * OUTPUT THE CODE
                ENDD
321 5120000000 X   FRMT5  SA2      FTEMP * GET TYPE
                7222777764 SX2      X2=+ENTRSUP * TRY FOR SUPPRESS EJECT MODE
322 0322000326 +   PL      Y2,FRMT4 * JUMP IF NOT NORMAL TYPE
                5140000000 X SA4      EJECTFL * GET MODE SWITCH
323 0304000332 +   ZR      X4,FDONE * QUIT IF NO SELECT EJECT NEEDED
                6170000325 + CALL      PUTBYTE * ELSE OUTPUT THE FUNCTION
325 0400000332 +   EQ      FDONE * AND QUIT
*
326 7160000000    SX6      0
                5160000000 X SA6      EJECTFL * CLEAR FLAG
327 5120000000 X   SA2      FTEMP
                7222777764 SX2      X2=+ENTRSUP
330 0302000332 +   ZR      X2,FDONE * QUIT: YOU WERE RIGHT

```

PRINTER DRIVER MAIN LOOPS
FORMAT PROCESSOR

COMPASS - VER 2.

08/19/71 17.20.21.

PAGE 16

331 5160000000 X 7160000005

SX6
SA6

EJECTFL . RESTORE, YOU WERE WRONG

332 6133777763

*
FDONE

SAVEN

SAVE POSITION FOR BYTE COUNT

342 7120000020

SX2

20B

343 0400000230 * 0701000253 *

GT
EQ

B1,B0,BLNKS1
LINE2


```

*
*
*      NWORD          GETS THE NEXT WORD OUT OF THE INPUT BUFFER
*
344  5011000001      NWORD   SA1       A1+1      * GET NEXT WORD
                                OR        X1,E0I    * CHECK FOR END OF FILE
345  0371000347 +     ID       X1,SLOPPY * CHECK FOR SLOPPY BLOCK
                                SB4      8         * RESET COUNT
346  20113           6140000010 LX1     60-49   * POSITIN CHARS
                                JP        B7        * RETURN
                                0270000000
*
347  SLOPPY          BSS      0
                                EXT      ATACHAD,DETCHAD,ATTACH,DETACH,DETCCT,ATATCHCT
*
*      DECIDE IF NEXT WORD IS IN BUFFER. IF SO, GET IT; ELSE READ
*      NEW BUFFER LOAD
*
347  74610           SX6      A1          . ADDRESS
                                SX1      X1       . SLOPPY LENGTH
350  36661           SA4      WDCNT      . GET OLD WORDCOUNT
                                IX6      X6+X1    . DESIRED CM ADDRESS
                                SX1      FILEBUF   . BUFFER ORIGIN
                                IX6      X6=X1     . WORDS USED IN BUFFER
351  37164           IX1      X6-X4      . USED - READ
                                PL        X1,SLOPPY . JUMP IF MUST READ
352  5216777776 X    SA1      FILEBUF-1+X6 . NEXT =1
                                EQ        NWORD    . AND CONTINUE
                                0400000344 +
*
*      NOW WE MUST COMPUTE HOW MUCH TO DETACH
*
*
ENTRY      SLOPPY1
*
353  5140000000 X    SLOPPY1 SA4      FADDR      . FILE ADDRESS
                                IX6      X6+X4      . ADD OFFSET FOR NEXT READ
                                SA6      A4         . UPDATE ADDRESS
354  5110000000 X    SA1      =XLOG2SIZ . GET LOG 2 OF BLOCKSIZE =1
                                SB4      X1         .
                                AX6      R4,X6      . ROUND DOWN TO BLOCK ORGION
355  22646           LX6      R4,X6
                                IX6      X6-X4      . POSSIBLE DETACH COUNT
                                37664
*
*      DETACH MIN OF NEXT READ BLOCK LESS LAST READ ADDRESS
*      OR TOTAL OF ATTACHED WORDS AT LAST ADDRESS
*
356  5110000000 X    SA3      ATACHAD   . GET LAST ATTACH ADDRESS
                                SA1      ATATCHCT  . AND COUNT
                                IX1      X1+X3     . LAST ATTACHED WORD + 1
                                IX1      X1-X4     . LESS READ ADDRESS IS TOTAL ATTACHED COUNT
357  37316           IX3      X1-X6     . COMPARE
                                PL        X3,**1    . JUMP, X6 IS LESS
                                0323000360 +

```

```

10611
360 0306000365 5160000000 x
361 5110000000 x
      13114
      23141
362 0301000365 10644
363 5160000000 x
364 0130000000 x
365 5140000000 x
      5130000000 x
366 36634
      5130000000 x
      37463
367 0324000370 10633
370 5160000000 x
371 20301
      0334000372 +
      37334
372 10633
      5160000000 x
373 0130000000 x
375 0130000000 x
377 5110777776 x
      0400000344 +

```

```

BX6      x1      . NOPE, X1 WAS LESS
SKIP DETACH IF MIN WAS 0
ZR        X6, NODETACH      . SKIP ON 0 COUNT
SA6       DETCHCT      . SET DETACH COUNT
SKIP DETACH IF DETACH IS IN SAME BLOCK AS READ
SA1       FADDR      . GET FILE ADDRESS FOR READ
BX1       X1-X4      . COMPARE ADDRESSES
AX1       R4,X1      . CLEAR BLOCK ADDRESS BITS
ZR        X1, NODETACH      . SKIP IF SAME BLOCK
BX6       X4      . OLD READ ADDRESS
SA6       DETCHAD      . IS DETACH ADDRESS
CEJ       DETACH      . DO THE DETACH

```

NOW COMPUTE ATTACH ADDRESS AND COUNT TO STAY ONE BUFFER LOAD AHEAD

GET THE ADDRESS FIRST

```

NODETACH SA4       ATACHAD      . OLD ATTACH ADDRESS
SA3       ATACHCT      . AND COUNT
IX6       X3+X4      . LAST ATTACHED WORD + 1
SA3       FADDR      . NEW READ ADDRESS
IX4       X6-X3      . LAST + 1 - READ

```

MUST KNOW IF ANY OF PRESENT READ IS ALREADY ATTACHED

```

PL        X4, **1      . JUMP IF OVERLAP EXISTS
BX6       X3      . ELSE ATTACH NEXT AT READ ADDRESS
SA6       ATACHAD      . SET ATTACH ADDRESS

```

THE AMOUNT OF OVERLAY IS IN X4 IF X4 IS POSITIVE
THEREFORE, WE SHOULD TRY TO ATTACH 2*WORDCOUNT-X4 IF X4 >= 0
OR JUST 2*WORDCOUNT IF X4 < 0

```

SA3       WDCNT      . GET OLD READ COUNT
LX3       1      . *2
NG        X4, **1      . SKIP ON X4 < 0
IX3       X3-X4
BX6       X3
SA6       ATACHCT      . SET ATTACH COUNT
CEJ       ATTACH, 1, FIXREAD      . DO THE ATTACH

```

FIX UP READ COUNT IF ATTACH FREturns 1

NOW READ, BUT QUIT ON FREturn

```

CEJ       READ, 1, EOI
SA1       FILEBUF=1
EQ        NWORD      . AND GO

```

HERE WE FIX UP THE ATTACH COUNT AFTER AN FRETURN TO REFLECT
WHAT WAS REALLY ATTACHED. THE READ IS CHECKED AND ITS COUNT
ADJUSTED IF NECESSARY

```

400 5110000000 Y
      5130000000 X
401 37161
      0301000705 +
      37663
402 5160000000 Y
      5130000000 X
403 37313
      0323000375 +
      10611
404 54630
      43601
      5261000000 X
405 0400000375 +

```

FIXREAD

```

SA1  FADDR      . READ ADDRESS
SA3  ATACHAD     . ATTACH ADDRESS
IX1  Y6=X1      . TOTAL CURRENTLY ATTACHED
ZR   X1=EOI     . DONE IF NONE LEFT
IX6  Y6=X3      . PROPER ATTACH COUNT
SA6  ATACHCT     . RESET IN CORE
SA3  WDCNT       . GET READ WORDCOUNT
IX3  X1=X3      . TOTAL ATTACHED - READ COUNT
PL   Y3,READ    . JUMP IF OK TO READ
BX6  X1          . ELSE SET READ COUNT TO
SA6  A3          . TOTAL AVAILABLE
MX6  1           . AND STORE AN EOI
SA6  X1+FILEBUF . AT END OF BUFFER AREA
EQ   READ       . AND THEN GO READ

```

```

*
*
*      PUTBYTE  OUTPUTS A BYTE FROM X4
*
*
406  22434      12774      6133777763
*      PUTBYTE  LX4      R3,X4      * SHIFT X4
*      BX7      X7,X4      * AND IT IN
*      SB3      R3-12      * DECREMENT SHIFT COUNT
407  0730000410      0270000000
*      LT      R3,R0,PUTBYTE1      * NEED MORE ROOM
*      JP      R7      * RETURN
410  5077000001      6130000060
*      PUTBYTE1 SA7      A7+1      * STORE OUTPUT WORD
*      SB3      R0-12      * RESET SHIFT COUNT
411  7170000000      0270000000
*      SX7      C      * CLEAR OUTPUT REGISTER
*      JP      R7      * RETURN
*
*
*      EBUFF  CLEANS UP LINES > 132 CHARACTERS LONG
*
*
412  001000000000020202020      30
*      EBUFWO  VFD      12/10B,24/0,24/20202020B      * SPECIAL HEADER
*      EBUFSHT EQU      34      * SHIFT COUNT FOR N IN SPECIAL HEADER
*      EBUFCHRS EQU      132      * NUMBER OF BLANKS PREFIXED
*      EBUFF  INSERTN      132      * N = MAX CHARS PER LINE
*      SX7      A7
413  5120000000 X      74770
*      SA7      F,RETUR      * SAVE LAST WORD LOCATION
416  5170000000 X      6122000020
*      GETBUFF      * GET FREE BUFFER
422  5120000412      10722
*      SA2      EBUFWO      * GET SPECIAL HEADER
*      BX7      Y2
423  7120000030      74370      20322
*      SA7      R2      * STORE IT, PREFIXING 4 BLANKS
*      SX2      EBUFSHT      * SHIFT COUNT FOR INSERTING N
*      SX3      A7      * LOCATION FOR SAME
424  14632      5160000000 X      43301
*      SA6      X3,X2
425  7125777573      6252000004
*      MX      NPOS      * POSITION FOR N SAVED
*      SX2      T      * BEGIN TO GENERATE MASK
426  20201      10622      20601      36226
*      SB5      R5-132      * NUMBER OF CHARACTERS TO SAVE
*      LX2      X2+EBUFCHRS      * NEW CHARACTER COUNT
*      BX6      T
427  14222      6272000073      23373
*      LX6      T
*      IX2      Y2,X6      * 6*COUNT=BIT COUNT
*      BX2      -X2
*      SB7      Y2+59
430  6177000001      6133000006
*      AX3      R7,X3      * -MASK IS CHARACTERS TO BE SAVED
*      SB7      R7+1      * 60-BIT COUNT
*      SB3      R3+6      * LAST CHARACTER POSITION
431  0707000432      66700      13333
*      GT      R7,R0,++1      * SHIFT WASNT NEGATIVE
*      SB7      R0
432  7163777703      0336000434
*      BX3      X3-X3      * ELSE SAVE EVERYTHING
*      SX6      R3-60      * NEED R3 MOD 60
433  6236000000      46000
*      NG      X6,*+2
*      SB3      X6+0
*      NO

```

NO
LX3
SA2
SA2
BX7
SB7
LX7
SB3
SB3
GE
SA7
SX7
SB3
LT
EQ
SX2
EQ

SX4
LX4
BX7
SA7
EQ

```

B3,X3      * HOLE IN MASK OVER CHARS
F1,RETUR   * LOCATION OF LAST WORD
X2         * LAST WORD
X3#X2      * CHARACTERS ARE SAVED $$$
B7-B3      * 60-COUNT-RT. EDGE
B7,X7      * LEFT JUSTIFY CHARS
B3-B7      * RT EDGE + LEFT SHIFT
B3-B6      * NEW SHIFT COUNT
B3-B0,EBUFF2 * JUMP IF ROOM LEFT
A7+1      * SAVE OUTPUT

B0-B6      * RESET SHIFT COUNT
B1,B0,EBUFF3 * FLAG FROM EOL
B1,B0,LINE2 * LINE CALLED
B0B        * RESTORE X2
BLNKSI     * FINISH UNPACKING

B0B        * ONE BLANK
B3,X4      * POSITIONED
X7-X4      * ADD ONE BLANK
B7+1      * STORE OUTPUT
FOL.1      * FINISH OUTPUTTING LINE

```

PRINTER DRIVER MAIN LOOPS
SEND EVENT

COMPASS - VER 2.

08/19/71 17.20.22.

PAGE 22

445 5150000000 X
7170000001
446 7160000020
20630
12767
447 14455
73642
63260
20640
450 12776
7160000000 X
37556
451 20514
12775
20544
12775
452 5140000000 X
20440
12774
453 20514
5140000000 X
76620
454 12747
5170000000 X
455 7170000000 X
37557
13777
456 0335000457 +
7170000001
457 5267000000 X
460 0130000000 X
461 0130000000 X
462 7266777774
0324000470 +
463 5150000000 X
0305000477 +
464 10655
13777
54750
63250
465 5160000000 X
7150000000 X
466 37665
5066777776
467 0270000000
470 0130000000 X
475 0100000000 X
476 0400000461 +

SENDEV

SENDEVI

REHANG

REHANG1

+

SA5
SX7
SX6
LX6
BX7
BX4
SX6
SB2
LX6
BX7
SX6
IX5
LX5
BX7
LX5
BX7
SA4
LX4
BX7
LX5
SA4
SX6
BX7
SA7
SX7
IX5
BX7
NG
SX7
SA6
CEJ
CEJ
SX6
PL
SA5
ZR
BX6
BX7
SA7
SR2
SA6
SX6
IX6
SA6
JP
JIMGREY
VFD
EQ

NBUFF

1

30B

34

X6+X7

X5

X4+82

X6

32

X7+X6

PRBUFF

X5-X6

12

X7+X5

36

X7+X5

XMYID

48

X7+X4

12

EBIT

82

X4+X7

EVNTDAT

SIZE

X5-X7

X7-X7

X5,SENDEVI

1

X7+BCOUNT

WRITEP

CALLPP

X6-3

X6,REHANG1

NEXTRUF

X5,GETEV

X5

X7-X7

A5

Y5

NBUFF

PRBUFF

X6-X5

A6-1

B7

OUT,ER,BRUCE

6/1,24/STOPE

REHANG

* PRINTER COMMAND

* BUFFER SIZE

* SAVE REQUEST COUNT IN R2

* REQUEST COUNT

* FIRST BUFFER FILE ADDRESS

* SEQUENCE NUMBER IS SAME

* INSERT SEQUENCE IDENTIFIER

* POSITION X5

* ERROR BIT

* STORE EVENT

* STORE COUNT

* WRITE BUFFER OUT

* CALL PP

* CHECK FOR REJECTED EVENT

* NEED EMPTY BUFFER

* DIDDLE FIRST TIME TO LOAD SECOND BU

* FFER

* RETURN

477	0130000000 X	GETEV	CEJ	GETPP	* GET RESPONSE AND EMPTY BUFFER
500	5140000000 X		SA4	FBIT	
	5170000000 X		SA7	BAD.EVN	* SAVE EVENTS FOR LATER INVESTIGATION
501	43560		MX5	60-12	
	0327000510 +		PL	X7+GOODEV	* CHECK ERROR BIT
	43201		MX2	1	
502	13642		BX6	X4-X2	* FLIP ERROR BIT
	5160000000 X		SA6	ERIT	* STORE
503	0307000574 +		ZR	X7.HELP	* EVENT CHANNEL GOT FULL SSSSS
	76670		SX6	B7	
504	5160000000 X		SA6	TEMP	* SAVE RETURN
	5170000506 +		CALL	STATUS	* FIND CAUSE OF ERROR
506	5140000000 X		SA4	TEMP	
	63740		SB7	X4	* RESTORE RETURN
507	0400000526 +		EQ	DIDDLE	* FIX UP EVENT
510	0357000525 +	GOODEV	OR	X7.DIDDLE2	* BAD ERROR BIT GIVEN
	15575		BX5	-X5*X7	* SEQUENCE NUMBER
	10455		BX4	X5	* SAVE X5 FOR LATER
511	43672		SEQCHECK	4	* CHECK AND CLEAR X5
	5140000000 X		SA4	XPRNMLG	
514	43673		MX6	1	
	54640		SA6	A4	* SET FLAG
	7160000000		SX6	0	
515	5160000000 X		SA6	XMSGNAME	* AND CLEAR ERROR MESSAGE
	0304000546 +		ZR	X4.GOODEVAP	* PRINT MESSAGE
516	7160000000 X	GOODEVA	SX6	SIZE	
	37556		IX5	X5-X6	
517	7160000000 X		SX6	PBUFF	
	0335000521 +		NG	X5+GOODEV1	
520	7266000000 X		SX6	X4+SIZE	
521	63260	GOODEV1	SB2	X6	* FOUND GOOD BUFFER
	5160000000 X		SA6	NPBUFF	
522	7170000000 X		SX7	PPBUFF	
	37647		IX6	X6-X7	
523	5066777776		SA6	A6=1	
	7170000012		SX7	10	
524	5170000000 X		SA7	FUNNY.C	* RESET UNKNOWN ERROR COUNTER
	0270000000		JP	B7	* RETURN
525	7130000001	DIDDLE2	SX3	1	* NO REQUESTS DONE, BUT
	20344		LX3	36	* FAKE OUT SUBTRACTION
	12737		BX7	X3+X7	* THAT COMES LATER
526	15475	DIDDLE	BX4	-X5*X7	* START ADDRESS
	43672		SEQCHECK	4	* CLEAR AND FIX UP X4
	7160000001		SX6	1	
531	20730		LX7	60-36	* REQUEST COMPLETED COUNT
	7130000020		SX3	20B	
	20330		LX3	24	
532	12663		BX6	X6+X3	* INSERT BUFFER SIZE
	15375		BX3	-X5*X7	* COUNT+20B
	20304		LX3	4	
533	7233777757		SX3	X3-20B	* HOWARD LIES S S S S
	36234		IX2	X3+X4	* NEW START ADDRESS
	20214		LX2	12	

PRINTER DRIVER MAIN LOOPS
GET EVENT

COMPASS - VER 2.

08/19/71 17.20.23.

PAGE 24

534	12762		BX7	X6+X2	* ADD START ADDRESS
	20244		LX2	36	
	12772		BX7	X7+X2	* SEQUENCE NUMBER
535	7120000000	X	SX2	SIZE	
	37242		IX2	X4-X2	
536	7160000000		SX6	0	
	0332000540	+	NG	X2, DIDDLE1	
537	7160000001		SX6	1	
540	5226000000	X	SA2	X6+BCOUNT	
	37623		IX6	X2-X3	* NEW REQUEST COUNT
	54620		SA6	A2	* STORE
541	0306000545	+	ZR	X6, DONE1	
	5140000000	X	SA4	FRIT	
542	20640		LX6	32	
	12767		BX7	X6+X7	* INSERT COUNT
	12747		BX7	X4+X7	* ERROR BIT
543	5150000000	X	SA5	XMYID	
	20540		LX5	48	
	12775		BX7	X7+X5	* INSERT SEQUENCE ID
544	5170000000	X	SA7	EVNTDAT	* STORE EVENT
	0400000461	+	EQ	REHANG	* SEND EVENT
545	73540		SX5	X4	
	0400000516	+	EQ	GOODEVA	
546	0130000000	X	GOODEVA P	JIMGREY	OUT, PRINTING PRINT MESSAGE
553	0400000516	+	EQ	GOODEVA	AND CONTINUE


```

*
*
7   E.6681 EQU 7
1   E.REU EQU 1
10  PAP.OUT EQU 10B
1   PRNT.RDY EQU 1
*
*
554 20760 STATUS LX7 40-12
      15375 BX3 X5*X7 * X3 = 6681 STATUS
      7160000046 SX6 1R-
555 20666 LX6 54
      20760 LX7 40-12
      15475 BX4 X5*X7 * X4 = PRINTER STATUS
      20730 LX7 24 * LEAVE X7 NICE
556 7120000010 SX2 PAP.OUT
      5160000000 X SA6 XPOPBUFF . FLASHING MESSAGE
557 11224 BX2 Y2*X4 * CHECK FOR PAPER OUT
      0302000561 + ZR X2.STATUS.1 * NOT PAPER OUT
560 7120000166 + MESOUT PR.PAPER
561 7120000001 STATUS.1 SX2 PRNT.RDY
      11224 BX2 Y2*X4 * CHECK PRINTER READY BIT
562 0312000564 + NZ X2.STATUS.2 * PRINTER IS READY
      7120000171 + MESOUT PR.NTRDY
564 7120000007 STATUS.2 SX2 E.6681
      11223 BX2 X2*X3 * CHECK FOR 6681 ERRORS
565 0302000573 + ZR X2.STATUS.3 * NONE FOUND
      7120000001 SX2 E.REU
566 11223 BX2 Y2*X3 * CHECK FOR REJECT
      0302000572 + ZR X2.STATUS.4 * NO REJECT
      20201 LX2 1
567 11223 BX2 Y2*X3 * INTERNAL OR EXTERNAL
      0302000571 + ZR X2.STATUS.5 * NOT EXTERNAL
570 7120000157 + MESOUT PR.EXTRJ
571 7120000154 + STATUS.5 MESOUT PR.INTRJ
572 7120000162 + STATUS.4 MESOUT PR.PAR * WAS PARITY ERROR, THEN
573 7120000173 + STATUS.3 MESOUT PR.UNKNO * UNKNOWN ERROR
*
*
574 7120000175 + HELP SX2 PR.HOWRD
      0400000575 + EQ HELP.1
*
*
575 0130000000 X HELP.1 JIMGREY OUT.X2
602 0100000000 X VFD A/1,24/STOPE
603 0130000000 X HELP.2 JIMGREY OUT.CANTFIX
610 0130000000 X JIMGREY OUT.ABORT
615 0400000000 X EQ EXIT2 * QUIT
*
*
*
616 43600 STATMES MX6 0
      5160000000 X SA6 XPRNMF LG . CLEAR PRINTING MESSAGE FLAG
617 5140000000 X SA4 XMSGNAME . GET ERROR MESSAGE LAST SENT

```

	13442		BX4	x4-x2	
	10622		BX6	x2	. COMPARE
620	0304000626 +		ZR	x4.SKIPMSG	. SKIP IF SAME ERROR
	5160000000 x		SA6	=XMSGNAME	. SET TO THIS MESSAGE AND PRINT IT
621	0130000000 x		JIMGREY	OUT,X2	* AND ISSUE MESSAGE
626		SKIPMSG	BSS	0	
626	5140000000 x		PDPST	PDPWAIT,4	
627	0130000000 x		CEJ	SAVE	
630	0130000000 x		CEJ	=XTIMEOUT	
631	0130000000 x		CEJ	RESTOR	
632	0270000000	STATMES1	JP	R7	* RETRY
		*			
		*			
		*			
633	0130000000 x	SEQERR	JIMGREY	OUT,SEQERRM	
640	0400000603 +		ENTRY	SEQERR	
			EQ	HELP.2	
		*			
		*			
641	0130000000 x	PDPWAIT	CEJ	SAVE	
642	7170000007		SX7	7	. WAIT SEC REQUEST
	6170000644 +		CALL	XPDPROST	
644	6170000645 +		CALL	XPDPRESF	
645	7160000045		SX6	1R+	
	2666		LX6	54	
646	5160000000 x		SA6	XPDPRUFF	
647	0130000000 x		CEJ	RESTOR	
650	0400000632 +		EQ	STATMES1	. RETURN

```

*
*
*      EOLF      HANDLES NULL LINES
*
651  7140000010      EOLF      SX4      10B      * CLEAR FCN
      6170000653 +    CALL      PUTBYTE
653  6133777763      SAVED
663  6133777771      EOLF.1    SB3      B3-6      * ADD WHOLE BYTE
      7140002020      SX4      2020B
664  22434          12774      LX4      B3,X4      * ADD 2 BLANKS
      5077000001      BX7      X7+X4
665  6150000002      SA7      A7+1      * OUTPUT WD
      0400000672 +    SB5      ?
      *      EQ      FOL.1      * FINISH UP
*
*
*      EOLCR     HANDLES NORMAL LINES
*
666  6170000002      EOLCR     SB7      2
      0757000663 +    LT      B5,B7,EOLF.1      * MAY BE NULL LINE
667  7140000020      SX4      20B      * ONE BLANK
      22434          LX4      B3,X4
      12774          BX7      X7+X4      * ADD ONE BLANK
670  5077000001      SA7      A7+1      * SAVE WD
      6170000204      SB7      132
671  6110777776      SB1      -1      * FLAG
      0775000413 +    GT      B5,B7,EBUFF      * CHECK FOR LINE TOO LONG
      *      EOL.1    INSERTN (B5+1)      * PUT BYTE COUNT IN
672  5120000000 x    GETBUFF      * GET NEXT BUFFER
675  6122000020      SA3      =XLINECT
701  5130000000 x    SX6      X3+1      * BUMP LINE COUNT
      7263000001      SA6      =XLINECT      * AND STORE IT BACK
702  5160000000 x    SA3      S.RETUR      * GET SAVED RETURN
      5130000000 x    SB7      Y3
703  63730          NE      B4,B6,NWORD      * GET NEW WORD AND RETURN
      0546000344 +    JP      B7
704  0270000000

```

PRINTER DRIVER MAIN LOOPS
END OF INFORMATION

COMPASS - VER 2.

08/19/71 17.20.25.

PAGE 28

705	6177777564 -	EOI	SB7	R7-LINE	
	0470000710 +		EQ	R7,B0,EOI1	* JUST FINISHED THIS LINE
706	7160000710 +		SX6	EOI1	* ELSE FINISH UP THIS ONE
	5160000000 x		SA6	S,RETUR	
707	0400000666 +		EQ	EOLCR	* FINISH LAST LINE
		*			
710	5110000000 x	EOI1	SA1	FRMTSUP	
	13666		BX6	X6-X6	
	54610		SA6	A1	* TURN ON FORMATS
711	0301000715 +		ZR	X1,EOIX	* JUMP IF SCOPE CONTROLS
		*			
	6160000010		CALLINE	CONVEX,EOIX	* SKIP TO TOP OF PAGE
		*			
714	00575500000000000000	CONVEX	VFD	11/27R,7/155B,42/0	* ASCII 7 AND CR
		*			
715	0130000000 x	EOIX	CEJ	SAVE	
716	0130000744 +		CEJ	TIME	* GET TIME AND DATE
717	5160000000 x		SA6	=XTTYBUFF	
	7170000155		SX7	155B	
720	20761		LX7	49	
	5076000001		SA7	A6+1	
721	6170000722 +		SB7	EOI2	
	0400000000 x		EQ	=XTYPEIN.	* MAKE DISPLAY CODE
		*			
722	5110000000 x	EOI2	SA1	=XPARAM	* GET CODED TIME
	5120000745 +		SA2	COLTPER	* GET CONSTANT
723	13112		BX1	X1-X2	* CHANGE COLONS TO PERIODS
	7120005746		SX2	2R,-	
	20260		LX2	60-12	
724	12612		BX6	X1+X2	
	20606		LX6	4	* -HH.MM.SS.
	5160000000 x		SA6	=XTIMSG	* STORE INTO MESSAGE
		*			
725	6110000000 x		SB1	=XLINECT	
	6020000001		SB2	A6+1	
726	6130777776		SB3	1	
	6140000055		SB4	55B	
727	6170000730 +	*	SB7	*+1	* RETURN
	0400000000 x		EQ	=XCONVNUM	* MAKE DECIMAL NUMBER
		*			
730	6110000000 x		SB1	=XTIMSG	* NEW SOURCE POINTER
	6170000732 +		SB7	*+2	
731	0400000000 x		EQ	=XDISASCI	* MAKE ASCII
		*			
		*			
732	0130000000 x		CEJ	RESTOR	* RESTORE REGISTERS
		*			
733	6140000010		CALLINE	TTYBUFF,EOI3	* PRINT TIME AND COUNT
		*			
736	0100000000 x	EOI3	RJ	=XTITLE	* PRINT TITLE LINES NOW
		*			
737	5120000000 x		SA2	NBUFF	* FINISH UP
	14222		BX2	-X2	

PRINTER DRIVER MAIN LOOPS
END OF INFORMATION

COMPASS - VER 2.

08/19/71 17.20.25.

PAGE 29

73222
740 0302000742 +
6170000742 +
742 6170000743 +
743 0400000000 X

ENDRUN

SX2
ZR
CALL
CALL
EQ
EXT
ENTRY

x2+82
x2,ENDRUN * JUST SENT EVENT
SENDEV * SEND LAST BUFFER
GETEV * MAKE SURE EVERYTHING GETS OUT
EXIT2
EXIT2
ENDRUN.LINE,STATUS,SLOPPY

744 4000000000 X
0000000000 X
745 00000000340000340000
63
23
31

*
TIME
COLTPER
ASCII5
ASCII3
ASCII9

VFD
DATA
EQU
EQU
EQU
END

1/1.20/=XEC:TIME.30/=XC.OPERC
340000340000B
63B
23B
31B

746

36211

STORAGE USED
6600 ASSEMBLY

1849 STATEMENTS
9.449 SECONDS

375 SYMBOLS
673 REFERENCES

000042 INVENTED SYMBOLS

PRINTER DRIVER MAIN LOOPS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

08/19/71 17.20.26.

PAGE 30

ABORT	205	PROGRAM*	8/10 L	25/47			
ASCII5	63		29/12 D				
ASCII3	23		29/13 D				
ASCII9	31		29/14 D				
ASCITAB	0	PROGRAM*	4/15 D	14/23	15/21		
ATACHAD	0	EXTERNAL*	17/48	18/23	18/33 S	19/08	
ATACHCT	0	EXTERNAL*	17/49	18/24	18/44 S	19/12 S	
ATTACH	0	EXTERNAL*	18/5				
BAD.EVN	0	EXTERNAL*	23/03 S				
BCOUNT	0	EXTERNAL*	22/30 S	24/09			
BEAD	0	EXTERNAL*					
BLNKS	247	PROGRAM*	14/32	14/40 L			
BLNKS1	253	PROGRAM*	14/43 L	14/46	16/06	21/17	
BUFFSIZ	0	EXTERNAL*					
CALLPP	0	EXTERNAL*	22/32				
CANTFIX	202	PROGRAM*	8/09 L	25/46			
COLTPER	745	PROGRAM*	28/26	29/11 L			
CONVEX	714	PROGRAM*	28/13	28/14 L			
CONVNUM	0	EXTERNAL*	28/39				
C.BEAD	0	EXTERNAL*					
C.CPCAL	0	EXTERNAL*					
C.FILE	0	EXTERNAL*					
C.FILE2	0	EXTERNAL*					
C.LINEA	0	EXTERNAL*					
C.OPERC	0	EXTERNAL*	29/10				
C.OWNFI	0	EXTERNAL*					
C.PPBUF	0	EXTERNAL*					
C.PPCAL	0	EXTERNAL*					
C.PRINT	0	EXTERNAL*					
C.SELF	0	EXTERNAL*					
DETACH	0	EXTERNAL*	18/16				
DETHAD	0	EXTERNAL*	18/15 S				
DETHCT	0	EXTERNAL*	18/06 S				
DIDDLE	526	PROGRAM*	23/15	23/42 L			
DIDDLE1	540	PROGRAM*	24/07	24/09 L			
DIDDLE2	525	PROGRAM*	23/16	23/39 L			
DISASCI	0	EXTERNAL*	28/43				
DISKSYS	1		2/02 D				
DONE1	545	PROGRAM*	24/12	24/22 L			
EBIT	0	EXTERNAL*	22/21	23/02	23/08 S	24/13	
EBUFCHRS	4		20/22 D	20/37			
EBUFF	413	PROGRAM*	14/26	14/46	20/23 L	27/29	
EBUFF2	441	PROGRAM*	21/10	21/14 L			
EBUFF3	443	PROGRAM*	21/14	21/20 L			
EBUFSHFT	30		20/21 D	20/30			
EBUFWD	412	PROGRAM*	20/20 L	20/27			
EC:TIME	0	EXTERNAL*	29/09				
EJECTFL	0	EXTERNAL*	15/44	15/50 S	16/02 S		
ENDRUN	742	PROGRAM*	29/02	29/04 L	29/07 E		
EOI	705	PROGRAM*	17/08	18/52	19/10	28/01 L	
EOIX	715	PROGRAM*	28/10	28/13	28/16 L		
EOI1	710	PROGRAM*	28/02	28/03	28/07 L		
EOI2	722	PROGRAM*	28/22	28/25 L			

PRINTER DRIVER MAIN LOOPS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

08/19/71 17.20.26.

PAGE 31

EO13	736	PROGRAM*	28/49	28/50 L					
EOLCR	666	PROGRAM*	14/30	27/21 L	28/05				
EOLF	681	PROGRAM*	15/08	27/06 L					
EOLF.1	663	PROGRAM*	27/09 L	27/22					
EOL.1	672	PROGRAM*	21/24	27/15	27/30 L				
ER.BRUC	200	PROGRAM*	8/08 L	22/47					
ER.EXTRJ	157	PROGRAM*	8/02 L	25/34					
ER.HOWRD	175	PROGRAM*	8/07 L	25/40					
ER.INTRJ	154	PROGRAM*	8/01 L	25/35					
ER.NTRDY	171	PROGRAM*	8/05 L	25/24					
ER.PAPER	166	PROGRAM*	8/04 L	25/20					
ER.PAR	162	PROGRAM*	8/03 L	25/36					
ER.UNKNO	173	PROGRAM*	8/06 L	25/37					
EVNTDAT	0	EXTERNAL*	22/24 S	24/20 S					
EXIT2	0	EXTERNAL*	25/47	29/05					
E.REJ	1		25/04 D	25/28					
E.6681	7		25/03 D	25/25					
FADDR	0	EXTERNAL*	17/36	18/10	18/26	19/07			
FDONE	332	PROGRAM*	15/22	15/45	15/47	15/53	16/04 L		
FILEBUF	0	EXTERNAL*	17/24	17/28	18/52	19/19 S			
FIXREAD	400	PROGRAM*	18/46	19/07 L					
FLEN	0	EXTERNAL*							
FRMT	263	PROGRAM*	14/18	15/04 L					
FRMTABL	136	PROGRAM*	7/21 D	7/45	15/34				
FRMTSUP	0	EXTERNAL*	14/16	28/07					
FRMT1	275	PROGRAM*	15/06	15/11	15/15 L				
FRMT2	300	PROGRAM*	15/16	15/21 L					
FRMT3	305	PROGRAM*	15/28	15/31 L					
FRMT4	326	PROGRAM*	15/43	15/49 L					
FRMT5	321	PROGRAM*	15/41	15/41	15/41	15/41	15/41 L		
FRMT.CL	0	EXTERNAL*	15/15	15/25 S	15/30 S				
FTEMP	0	EXTERNAL*	15/33 S	15/41	15/51				
FUNNY.C	0	EXTERNAL*	23/37 S						
F.RETUR	0	EXTERNAL*	20/25 S	21/03					
GETEV	477	PROGRAM*	22/36	23/01 L	29/05				
GETPP	0	EXTERNAL*	23/01						
GOODEV	510	PROGRAM*	23/05	23/16 L					
GOODEVA	516	PROGRAM*	23/26 L	24/23	24/27				
GOODEVAP	546	PROGRAM*	23/25	24/26 L					
GOODEV1	521	PROGRAM*	23/29	23/31 L					
HELP	574	PROGRAM*	23/09	25/40 L					
HELP.1	575	PROGRAM*	25/41	25/43 L					
HELP.2	603	PROGRAM*	25/45 L	26/17					
LRUFF	0	EXTERNAL*							
LINE	213	PROGRAM*	14/06 L	28/01	28/13	28/49	29/07 E		
LINECT	0	EXTERNAL*	27/32	27/34 S	28/34				
LINE1	220	PROGRAM*	14/17	14/19 L					
LINE2	230	PROGRAM*	14/20 L	14/26	14/43	16/07	21/15		
LINE3	234	PROGRAM*	14/23 L	14/34					
LOG2SIZ	0	EXTERNAL*	17/39						
MSGNAME	0	EXTERNAL*	23/24 S	25/53	26/04 S				
MYID	0	EXTERNAL*	22/17	23/20	23/44	24/17			
NBUFF	0	EXTERNAL*	20/27	22/01	22/41 S	23/32 S	27/32	28/52	

PRINTER DRIVER MAIN LOOPS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

08/19/71 17.20.26.

PAGE 32

NEXTBUF	0	EXTERNAL*	22/35							
NODETACH	365	PROGRAM*	18/05	18/13	18/23 L					
NPOS	0	EXTERNAL*	14/20 S	16/05 S	20/23	20/34 S	27/09 S	27/30		
NWORD	344	PROGRAM*	14/21	14/41	15/05	15/13	17/07 L	17/29	18/53	27/37
PAP.OUT	10		25/05 D	25/16						
PARAM	0	EXTERNAL*	28/25							
PDPBUFF	0	EXTERNAL*	25/17 S	26/26 S						
PDPFLG	0	EXTERNAL*	22/47	24/27	25/44	25/46	25/47	26/06	26/07	26/16
PDPMSG	0	EXTERNAL*	22/47	24/27	25/44	25/46	25/47	26/06	26/16	
PDPRESP	0	EXTERNAL*	26/24							
PDPQRST	0	EXTERNAL*	26/23							
PDPWAIT	641	PROGRAM*	26/08	26/20 L						
PPRUFF	0	EXTERNAL*	22/11	22/42	23/28	23/33				
PRNMFLG	0	EXTERNAL*	23/20	25/52 S						
PRNTING	211	PROGRAM*	8/12 L	8/13 E	24/27					
PRNT.RDY	1		25/06 D	25/21						
PUTBYTE	406	PROGRAM*	15/21	15/41	15/41	15/47	27/08			
			15/41	15/41	15/41	20/06 L				
PUTBYTE1	410	PROGRAM*	20/09	20/11 L						
READ	375	PROGRAM*	18/51 L	19/15	19/20					
READF	0	EXTERNAL*	18/51							
REHANG	461	PROGRAM*	22/32 L	22/48	24/21					
REHANG1	470	PROGRAM*	22/34	22/46 L						
RESTOR	0	EXTERNAL*	22/47	25/44	25/47	26/10	26/27			
			24/27	25/46	26/06	26/16	28/46			
SAVE	0	EXTERNAL*	22/46	25/43	25/46	26/08	26/20			
			24/26	25/45	26/05	26/15	28/16			
SENDEV	445	PROGRAM*	20/27	22/01 L	27/32	29/04				
SENDEV1	457	PROGRAM*	22/28	22/30 L						
SEQERR	633	PROGRAM*	23/20	23/44	26/15 L	26/16 E				
SEQERRM	206	PROGRAM*	8/11 L	26/16						
SIZE	0	EXTERNAL*	20/27	22/25	23/26	23/30	24/04	27/32		
SKIPMSG	626	PROGRAM*	26/03	26/06 L						
SLOPPY	347	PROGRAM*	17/09	17/14 L	29/07 E					
SLOPPY1	353	PROGRAM*	17/27	17/34 E	17/36 L					
SPECIAL	244	PROGRAM*	14/22	14/29 L						
STATMES	616	PROGRAM*	25/21	25/25	25/35	25/36	25/37	25/38	25/51 L	
STATMES1	632	PROGRAM*	26/11 L	26/28						
STATUS	554	PROGRAM*	23/13	25/09 L	29/07 E					
STATUS.1	561	PROGRAM*	25/19	25/21 L						
STATUS.2	564	PROGRAM*	25/23	25/25 L						
STATUS.3	573	PROGRAM*	25/27	25/37 L						
STATUS.4	572	PROGRAM*	25/30	25/36 L						
STATUS.5	571	PROGRAM*	25/33	25/35 L						
STOPE	0	EXTERNAL*	22/47	25/44						
S.RETUR	0	EXTERNAL*	14/07 S	27/35	28/04 S					
TEMP	0	EXTERNAL*	23/11 S	23/13						
TIME	744	PROGRAM*	28/17	29/09 L						
TIMEOUT	0	EXTERNAL*	26/09							
TIMSG	0	EXTERNAL*	28/32 S	28/41						
TITLE	0	EXTERNAL*	28/50							
TITLECT	0	EXTERNAL*								
TITLEFM	0	EXTERNAL*								

PRINTER DRIVER MAIN LOOPS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

08/19/71 17.20.27.

PAGE 33

TTYBUFF	0	EXTERNAL*	28/18 S	28/49					
TYPEIN.	0	EXTERNAL*	28/23						
TYPEOUT	0	EXTERNAL*	22/47	24/27	25/44	25/46	25/47	26/06	26/16
WDCNT	0	EXTERNAL*	17/22	18/39	19/13				
WRITEF	0	EXTERNAL*	22/31						
.A	41	PROGRAM*	4/49 L						
.AMPRSD	6	PROGRAM*	4/22 L						
.APOSTR	100	PROGRAM*	5/27 L						
.AT	40	PROGRAM*	4/48 L						
.B	42	PROGRAM*	4/50 L						
.BKAROW	77	PROGRAM*	5/26 L						
.BKSLASH	74	PROGRAM*	5/23 L						
.BLANK	0	PROGRAM*	4/16 L						
.C	43	PROGRAM*	4/51 L						
.COLON	32	PROGRAM*	4/42 L						
.COMMA	14	PROGRAM*	4/38 L						
.D	44	PROGRAM*	4/52 L						
.DBQOT	2	PROGRAM*	4/18 L						
.DLRSGN	4	PROGRAM*	4/20 L						
.DOT	16	PROGRAM*	4/30 L						
.E	45	PROGRAM*	4/53 L						
.EQUAL	35	PROGRAM*	4/45 L						
.EXCLAM	1	PROGRAM*	4/17 L						
.F	46	PROGRAM*	5/01 L						
.FTRTHN	36	PROGRAM*	4/46 L						
.G	47	PROGRAM*	5/02 L						
.H	50	PROGRAM*	5/03 L						
.I	51	PROGRAM*	5/04 L						
.J	52	PROGRAM*	5/05 L						
.K	53	PROGRAM*	5/06 L						
.L	54	PROGRAM*	5/07 L						
.LBRK	73	PROGRAM*	5/22 L						
.LCASEA	101	PROGRAM*	5/28 L						
.LCASEB	102	PROGRAM*	5/29 L						
.LCASEC	103	PROGRAM*	5/30 L						
.LCASED	104	PROGRAM*	5/31 L						
.LCASEE	105	PROGRAM*	5/32 L						
.LCASEF	106	PROGRAM*	5/33 L						
.LCASEG	107	PROGRAM*	5/34 L						
.LCASEH	110	PROGRAM*	5/35 L						
.LCASEI	111	PROGRAM*	5/36 L						
.LCASEJ	112	PROGRAM*	5/37 L						
.LCASEK	113	PROGRAM*	5/38 L						
.LCASEL	114	PROGRAM*	5/39 L						
.LCASEM	115	PROGRAM*	5/40 L						
.LCASEN	116	PROGRAM*	5/41 L						
.LCASEO	117	PROGRAM*	5/42 L						
.LCASEP	120	PROGRAM*	5/43 L						
.LCASEQ	121	PROGRAM*	5/44 L						
.LCASER	122	PROGRAM*	5/45 L						
.LCASES	123	PROGRAM*	5/46 L						
.LCASET	124	PROGRAM*	5/47 L						
.LCASEU	125	PROGRAM*	5/48 L						

PRINTER DRIVER MAIN LOOPS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

08/19/71 17.20.27.

PAGE 34

.LCASEV	126	PROGRAM*	5/49 L
.LCASEW	127	PROGRAM*	5/50 L
.LCASEX	130	PROGRAM*	5/51 L
.LCASEY	131	PROGRAM*	5/52 L
.LCASEZ	132	PROGRAM*	5/53 L
.LOSS	34	PROGRAM*	4/44 L
.LPAREN	10	PROGRAM*	4/34 L
.LSQUIG	135	PROGRAM*	6/03 L
.M	55	PROGRAM*	5/08 L
.MINUS	15	PROGRAM*	4/29 L
.N	56	PROGRAM*	5/09 L
.NUMSGN	3	PROGRAM*	4/19 L
.O	57	PROGRAM*	5/10 L
.P	60	PROGRAM*	5/11 L
.PERCNT	5	PROGRAM*	4/21 L
.PLUS	13	PROGRAM*	4/27 L
.Q	61	PROGRAM*	5/12 L
.QUEST	37	PROGRAM*	4/47 L
.R	62	PROGRAM*	5/13 L
.RBRK	75	PROGRAM*	5/24 L
.RPAREN	11	PROGRAM*	4/25 L
.RSQUIG	133	PROGRAM*	6/01 L
.S	63	PROGRAM*	5/14 L
.SEMICOL	33	PROGRAM*	4/43 L
.SLASH	17	PROGRAM*	4/31 L
.SNQUOT	7	PROGRAM*	4/23 L
.STAR	12	PROGRAM*	4/26 L
.T	64	PROGRAM*	5/15 L
.U	65	PROGRAM*	5/16 L
.UPAROW	76	PROGRAM*	5/25 L
.V	66	PROGRAM*	5/17 L
.VERTBAR	134	PROGRAM*	6/02 L
.W	67	PROGRAM*	5/18 L
.X	70	PROGRAM*	5/19 L
.Y	71	PROGRAM*	5/20 L
.Z	72	PROGRAM*	5/21 L
.0	20	PROGRAM*	4/32 L
.1	21	PROGRAM*	4/33 L
.2	22	PROGRAM*	4/34 L
.3	23	PROGRAM*	4/35 L
.4	24	PROGRAM*	4/36 L
.5	25	PROGRAM*	4/37 L
.6	26	PROGRAM*	4/38 L
.7	27	PROGRAM*	4/39 L
.8	30	PROGRAM*	4/40 L
.9	31	PROGRAM*	4/41 L
^BOTTOM	12		4/41 D
^CONCAV	7		7/13 D
^CONVEX	10		7/14 D
^DOUBL	1		7/07 D
^EJECT	0		7/06 D
^ENTRSUP	13		5/19
^HALF	4		4/34

7/16 D
7/13 D
7/14 D
7/07 D
7/06 D
5/51
7/10 D

7/17 D 15/42 15/52

PRINTER DRIVER MAIN LOOPS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

08/19/71 17.20.28.

PAGE 35

↑LEAVSUP	14	5/20	5/52	7/18 D	
↑OVER	3	4/27	7/09 D		
↑QTR	6	4/36	7/12 D		
↑SUPRES1	15	5/21	5/53	7/19 D	15/27
↑THIRD	5	4/35	7/11 D		
↑TOP	11	4/40	7/15 D		
↑TRIPL	2	4/29	7/08 D		
↑A	61	3/18 D	4/49	5/28	
↑AND	37	4/11 D	4/22		
↑B	62	3/19 D	4/50	5/29	
↑BLANK	20	3/49 D	4/16		
↑C	63	3/20 D	4/51	5/30	
↑COLON	0	3/11 D	4/42		
↑COMMA	33	4/07 D	4/28		
↑D	64	3/21 D	4/52	5/31	
↑DLRSGN	53	3/44 D	4/20		
↑DNAROW	56	3/47 D	4/18		
↑E	65	3/22 D	4/53	5/32	
↑EQUAL	13	3/12 D	4/45		
↑F	66	3/23 D	5/01	5/33	
↑G	67	3/24 D	5/02	5/34	
↑GRTR	57	3/48 D	4/46		
↑GTEQU	75	3/30 D	4/47		
↑H	70	3/35 D	5/03	5/35	
↑I	71	3/26 D	5/04	5/36	
↑IDENT	36	4/10 D	4/19		
↑J	41	3/34 D	5/05	5/37	
↑K	42	3/35 D	5/06	5/38	
↑L	43	3/36 D	5/07	5/39	
↑LBRAK	17	3/16 D	5/22		
↑LESS	72	3/27 D	4/44		
↑LESSEQU	15	3/14 D	4/17		
↑LPAREN	34	4/08 D	4/24	6/03	
↑M	44	3/37 D	5/08	5/40	
↑MINUS	40	3/33 D	4/29		
↑N	45	3/38 D	5/09	5/41	
↑NEQUAL	14	3/13 D	4/23	5/27	
↑NOT	76	3/31 D	4/48		
↑O	46	3/39 D	5/10	5/42	
↑OR	52	3/43 D	5/23	6/02	
↑P	47	3/40 D	5/11	5/43	
↑PERCNT	16	3/15 D	4/21		
↑PERIOD	73	3/28 D	4/30		
↑PLUS	60	3/17 D	4/27		
↑Q	50	3/41 D	5/12	5/44	
↑R	51	3/42 D	5/13	5/45	
↑RBRAC	32	4/06 D	5/24		
↑RPAREN	74	3/39 D	4/25	6/01	
↑RTAROW	35	4/09 D	5/26		
↑S	22	3/51 D	5/14	5/46	
↑SEMICOL	77	3/32 D	4/43		
↑SLASH	21	3/50 D	4/31		
↑STAR	54	3/45 D	4/26		

PRINTER DRIVER MAIN LOOPS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

08/19/71 17.20.28.

PAGE 36

↓T	23	3/52 D	5/15	5/47
↓U	24	3/53 D	5/16	5/48
↓UPAROW	55	3/46 D	5/25	
↓V	25	4/01 D	5/17	5/49
↓W	26	4/02 D	5/18	5/50
↓X	27	4/03 D	5/19	5/51
↓Y	30	4/04 D	5/20	5/52
↓Z	31	4/05 D	5/21	5/53
↓0	12	3/01 D	4/32	
↓1	1	3/02 D	4/33	
↓2	2	3/03 D	4/34	
↓3	3	3/04 D	4/35	
↓4	4	3/05 D	4/36	
↓5	5	3/06 D	4/37	
↓6	6	3/07 D	4/38	
↓7	7	3/08 D	4/39	
↓8	10	3/09 D	4/40	
↓9	11	3/10 D	4/41	

PRINTER DRIVER MAIN LOOPS
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

08/19/71 17.20.28.

PAGE 36

↑T	23	3/52 D	5/15	5/47
↑U	24	3/53 D	5/16	5/48
↑UPAROW	55	3/46 D	5/25	
↑V	25	4/01 D	5/17	5/49
↑W	26	4/02 D	5/18	5/50
↑X	27	4/03 D	5/19	5/51
↑Y	30	4/04 D	5/20	5/52
↑Z	31	4/05 D	5/21	5/53
↑0	12	3/01 D	4/32	
↑1	1	3/02 D	4/33	
↑2	2	3/03 D	4/34	
↑3	3	3/04 D	4/35	
↑4	4	3/05 D	4/36	
↑5	5	3/06 D	4/37	
↑6	6	3/07 D	4/38	
↑7	7	3/08 D	4/39	
↑8	10	3/09 D	4/40	
↑9	11	3/10 D	4/41	