

ADDRESS LENGTH

BINARY CONTROL CARDS.

0 447
447

IDENT CMDC
END

ENTRY POINTS.

STASCII -	232	OCTPK -	42	COUNT -	212	SETGENA -	303
PKASCII -	240	RLNPK -	220	ZROPK -	225	GENASC -	305
CRASCII -	257	OCTBPK -	52	DECNPK -	71	DSPLYW -	371
FNASCII -	253	OCTNPK -	61	DCBPK -	141	ADDCHAR -	436
DISLINE -	0	DIGIT -	161	LINEPK -	24	NXTPRM -	345

EXTERNAL SYMBOLS.

CALLR RTNR DSPLYWL PARAMBF PRMBFSZ

3

```
*
*
*
*
ERR      MACRO
          CALL
          ENRM

*
*
*
EXT      CALLR,RTNR
EXT      DSPLYWL,PARAMBF,PRMBFSZ

*
*
*
ENTRY    STASCII,PKASCII,CRASCII,FNASCII
ENTRY    DISLINE,OCTPK,BLNKPK,OCTBPK,OCTNPK
ENTRY    DIGIT,COUNT,ZROPK,DECNPK,DCBPK,LINEPK
ENTRY    SETGENA,GENASC
ENTRY    DSPLYW
ENTRY    ADDCHAR
ENTRY    NXTPRM
```

```
*
*
*
*
XJ      MACRO      LOC,FRTN,ERR,ERRTAG,RTN
*
      IFC          EQ,SRTNSS
+      VFD          12/0130B,18/LOC,12/0,18/2
      ENDIF
*
      IFC          NE,SRTNSS
+      VFD          12/0130B,18/LOC,12/1,18/3
      VFD          40/RTN
      ENDIF
*
      SB7          4+1
*
      IFC          EQ,SFRTNSS
      JP           UNEXFRTN
      ENDIF
*
      IFC          NE,SFRTNSS
      JP           FRTN
      ENDIF
*
      IFC          EQ,SEERRSS
      JP           UNEXERR
      ENDIF
*
      IFC          NE,SEERRSS
      JP           ERR
      ENDIF
*
      IFC          EQ,SEERRTAGSS
+      VFD          40/0
      ENDIF
*
      IFC          NE,SEERRTAGSS
+      VFD          40/ERRTAG
      ENDIF
*
      ENDM
*
*
CALL     MACRO      LOC,A,B,C,D,E
      SETBS      A,B,C,D,E
+      SB7          4+1
      JP          LOC
      ENDM
*
*
SAVE7    MACRO      L
      SX6          R7
```

```

SA6
ENDM

*
*
GET7    MACRO
SA1
SB7
ENDM    X1

*
*
CALLR   MACRO
SETBS   A,B,C,D,E
SB7     A,B,C,D,E
SX6     *+1
JP      CALLR
ENDM

*
*
SETBS   MACRO
IFC     A,B,C,D,E
SB1     NE,SSAS
IFC     A,B,C,D,E
SB2     NE,SSBS
IFC     A,B,C,D,E
SB3     NE,SSCS
IFC     A,B,C,D,E
SB4     NE,SSDS
IFC     A,B,C,D,E
SB5     NE,SSFS
ENDIF
ENDM

*
*
RTNR    MACRO
JP      RTNR
ENDM

*
*
POPR    MACRO
SB7     *+1
JP      POPR
ENDM

*
*
MCAP    MACRO
BSS     NAME
EQU     0
VFD     1/1,29/C.NAME,30/CX,MAST
ENDM

*
*
MXCAP   MACRO
VFD     NAME
1/1,29/C.NAME,30/CX,MAST

```

```

*
*
* ITEMS      MACRO      A,B,C,D,E,F,G,H
VFD          30/A
IFC          NE,SSBS
VFD          30/B
IFC          NE,SSCS
VFD          30/C
IFC          NE,SSDS
VFD          30/D
IFC          NE,SSFS
VFD          30/E
IFC          NE,SSFS
VFD          30/F
IFC          NE,SSGS
VFD          30/G
IFC          NE,SSHS
VFD          30/H
ENDIF
ENDM

```

```

*
*
* MAP      MACRO      NAME,X,FLAD,CMAD,1,ASTPI,RO
VFD        30/OL,NAME
VFD        30/X,30/FLAD
VFD        1/RO,29/CMAD,30/LASTPI
ENDM

```

```

*
*
*
* RTNCAP    MACRO      NAME
DATA        0
RMT
DATA        OL,NAME
RMT
ENDM

```

```

*
*
*
* PARCAP    MACRO      NAME
DATA        OL,NAME
ENDM

```

```

*
*
* L          MACRO      MICROX,L,NAME
MICRO       1, /NAME /
ENDM

```

```
*
*
*
EVECN      MACRO      CLASS,NUM,LOC
+          VFD        6/0,18/CLASS,18/NUM,18/LOC
          ENDM
```

```
*
*
*
EVECT      MACRO      XFER
+          VFD        60/-XFER
          ENDM
```

```
*
*
*
NAMEZ      MACRO      NAME
          IFC        NE,$$NAMES
          VFD        60/0L+NAME
          ENDIF
          IFC        EQ,$$NAMES
          VFD        60/0
          ENDIF
          ENDM
```

```
*
*
*
BDMAP      MACRO      FNAME,UNAME,FLAD,CMAD,LASTP1,RO
          NAMEZ      FNAME
          NAMEZ      UNAME
          VFD        60/FLAD,60/CMAD,1/1,59/LASTP1
          VFD        60/RO
          ENDM
```

```
*
*
*
L          MACRO      CAP,L,FNAME,UNAME
CXPNTR     EQU        CXPNTR
          SET        CXPNTR+1
          NAMEZ      FNAME
          NAMEZ      UNAME
          ENDM
```

```
*
*
*
L          MACRO      CAPZ,L
CXPNTR     EQU        CXPNTR
          SET        CXPNTR+1
          ENDM
```

```
*
*
*
DEBUG      MACRO      XJ
          XJ          ESAVE
          SB6         4
          XJ          READ
```

```

XJ      FRESTORE
ENDM

*
*
MSGOUT  MACRO
LOCAL  MSG
CALLR  D,E
JP      MSGOUT,D
DIS    .S,MSG,S
BSS    .S,MSG,S
ENDM

*
* IMPY
*
IMPY    MACRO
        INTO, FROM1, FROM2
        P, FROM1
        P, FROM2
        D, INTO
        U, INTO
        INTO
        ENDM

*
* IDIV
*
IDIV    MACRO
        INTO, FROM1, FROM2, SCRB
        P, FROM1
        P, FROM2
        N, FROM2
        F, INTO
        U, INTO
        L, INTO
        ENDM

*
* WORD
*
WORD    MACRO
        NO
        NO
        NO
        NO
        ENDM

```

```

*
*
*
*
XSETXJ  MACRO      A,B,C,D,E
LOCALPXJ SET
          IFC
          SX1
LOCALPXJ SET
          IFC
          SX2
LOCALPXJ SET
          IFC
          SX3
LOCALPXJ SET
          IFC
          SX4
LOCALPXJ SET
          IFC
          SX5
LOCALPXJ SET
          ENDM

*
*
*
*
YSETXJ  MACRO      A,B,C,D,E,F,G,H,I,J
XSETXJ  A,B,C,D,E
SB2
CALL    SET
XSETXJ  F,G,H,I,J
ENDIF
SB2
CALL    LOCALPXJ
CALL    SET
ENDM

*
*
*
*
DOXJ    MACRO      A,B,C,D,E,F,G,H,I,J
SB1
XJLOC
YSETXJ  A,B,C,D,E,F,G,H,I,J
XJ
XJLOC
ENDM

*
*
*
*
*
*
MULTI LINE VERSIONS
INITIAL SET UP LINE
SETXJ   MACRO      A,B,C,D,E,F,G,H,I,J
SB1
XJLOC
YSETXJ  A,B,C,D,E,F,G,H,I,J

```


ENDM

*
*
*

FINAL CALL (USE YSETXJ IN BETWEEN)

XDOXJ

MACRO
YSETXJ
XJ
ENDM

A,B,C,D,E,F,G,H,I,J
A,B,C,D,E,F,G,H,I,J
XJLOC

					INSTALL DISPLAY CODE IN AN ASCII LINE	
		B1			POINTS TO 1ST WORD OF DISPLAY CODE	
		B2			COUNT	
					STOPS WHEN COUNT GOES TO ZERO OR ON ZERO CHARACTER	
				R3	CHARS LEFT IN CURRENT WORD	
				X1	CURRENT WORD	
				A1	ADDRESS CURRENT WORD	
				A0	HOLDS RETURN	
				X2	6 BIT MASK	
				X3	WILL HOLD COUNT	
				B2	WILL HOLD A SHIFT COUNT DURING XLATION	
				X4	PORTION OF XLATE TABLE	
0	56070	DISLINE	SA0	B7		
	56110		SA1	B1		
	43266		MX2	A0-6		
1	6130000012		SB3	10		
	74320		SX3	B2		
2	0303000013	DISLINE1	ZR	X3,DISLINE3	ALL DONE	
	0703000004		GT	B3,B0,DISLINE2	SOME LEFT IN CURRENT WORD	
3	5011000001		SA1	A1+1	NEED A NEW WORD	
	6130000012		SB3	10		
4	6133777776	DISLINE2	SB3	B3-1		
	7233777776		SX3	X3-1		
5	20106		LX1	6		
	15712		BX7	X2*X1		
	0407000013		ZR	X7,DISLINE3	ZERO CHARACTER, ALL DONE	
6	20771		LX7	A0-3	NOW BEGIN FANCY XLATION	
	63270		SB2	X7	WORD DISPLACEMENT	
	5142000014		SA4	DISASC+B2		
7	11727		BX7	X2*X7		
	20703		LX7	3		
	63270		SB2	X7	CHARACTER INCREMENT	
	20703		LX7	3	B*CHAR INC	
10	67202		SB2	B2		
	63272		SB2	X7+B2	7*CHARINC	
	22424		LX4	B2,X4	POSITION WORD FROM TABLE	
	20413		LX4	A+7	REST OF SHIFT	
11	43765		MX7	A0-7		
	15747		BX7	X7*X4	NOW HAVE CHARACTER XLATED	
12	0200000240		SB7	DISLINE1		
			JP	PKASCII	BACK INTO LINE	

13 64700
0270000000

*
DISLINE3 SB7 40
JP R7 EXIT

*
*
*
*
*
TOASC MACRO S,T,U,V,W,X,Y,Z
+ VFD 4/0,7/SPB,7/TPB,7/UPB,7/VPB,7/WPB
VFD 7/XPB,7/YPB,7/ZPB
ENDM

*
*
*
*
CONVERSION TABLE

14	0000412110644	DISASC	TOASC	00,41,42,43,44,45,46,47
15	0120512512654		TOASC	50,51,52,53,54,55,56,57
16	0140613114664		TOASC	60,61,62,63,64,65,66,67
17	0160713504021		TOASC	70,71,72,20,21,22,23,24
20	0052261346031		TOASC	25,26,27,30,31,13,15,12
21	0036100441035		TOASC	17,10,11,04,35,00,14,16
22	0006733646407		TOASC	03,73,75,32,07,77,01,06
23	0174021607401		TOASC	76,02,34,36,01,37,76,33

```

*
* LINEPK
* RIGHT JUSTIFY, BLANK FILL DISPLAY CODE INTO AN ASCII LINE
*
* INPUT
* B1=PTR TO DISPLAY CODE
* B2=FIELD COUNT
* B7=RETURN
*
* CAUSES ALL REGISTERS TO BE USED. CALLS DISLINE
* X1=CURRENT WORD OF DISPLAY CODE
* X2=6 BIT MASK (FOR BCD CHARACTERS)
* X3=CURRENT CHAR
* B1=1
* B2=COUNT OF BCD CHARS LEFT IN CURRENT WORD OF DISPLAY CODE
* B3=TOTAL NUM OF CHARS SEEN SO FAR
*
24 43206 76010 20044 76370 LINEPK MX2 6 . INITIALIZE THINGS. SAVE B1,B2,B7
      SX0 36
      LX0 36
      SX3 37
      SX4 32
      LX4 18
      BX4 X0+X4
      BX4 X3+X4
      SB3 30
      SB2 10
      SA1 B1+0
      SB1 B1
      JP LINEPK2
*
* LOOP LOOKING FOR BCD CHARACTERS
*
31 6120000012 5017000001 LINEPK1 SB2 10 . RESET B2
      SA1 A1+1 . FETCH NEXT WORD OF BCD CHARACTERS
32 67221 11321 0303000035 + LINEPK2 SB2 B2-B1 . DECREMENT INNER LOOP CTRL
      BX3 X2*X1 . ISOLATE CHAR
      ZR X3,LINEPK3 . JP IF NOMORE BCD CHARS
*
33 66331 20106 0702000032 + SB3 B3+B1 . INCREMENT COUNT
      LX1 6
      GT B2,B0,LINEPK2
34 0200000031 + JP LINEPK1
*
* NO MORE DISPLAY CODE. ADD BLANKS IF APPROPRIATE
*
35 20452 63240 67123 LINEPK3 LX4 60-18 . RESTORE B2, X2=VFD 18/B7,6/0,18/B1,18/B2
      SB2 84
      SB1 B2-B3 . B1=NUM BLANKS TO ADD, IF ANY
*
36 0632000040 + LE B2,B3,LINEPK4 . JP IF NO BLANK INSERTION
      SB7 LINEPK4
37 0200000220 + JP BLNPKP . INSERT BLANKS

```

```

      *
40  63240      LINEPK4  SB2      X4      . RESTORE REGISTERS
      21422      AX4      18
      63140      SB1      X4
      21430      AX4      6+18
41  63740      SB7      X4
      0200000000 +      JP      DISLINE
```

[illegible]

[illegible]

[illegible]


```

*
* DECNPk
*
* CONVERT AN OCTAL INTEGER INTO DECIMAL AND
* INSTALL THE DECIMAL INTO AN ASCII LINE
*
* LEADING ZEROS ARE DELETED
* IF INPUT=0, A SINGLE ZERO WILL BE PRINTED
*
* INPUT
*
* X1=OCTAL NUMBER
* B1=MAX NUM CHARS TO USE
* B2=POWER OF TEN TO USE AS SCALING FACTOR
* B7=RETURN
*
* IF NUMBER IS TOO BIG TO FIT IN MAX NUM CHARS (B1) THEN
* B1 ASTERICKS ARE PRINTED OUT.
*
* USES ALL REGISTERS
*
* CALLS DIGIT,COUNT,PKASCII
*
71 43214      10311      0321000073 +
72 46000      7130000000      37331
73 11332      76410      43252      76720
74 15772      20422      76270      20244
75 6170000100 +      12747      12727
76 0303000161 +      21744      73470
77 0200000136 +

*
* DECNPk
*
* MX2      12
* BX3      X1
* PL      X1,DCNPK1
* NO      0
* SX3      0
* IX3      X3-X1      . X3=ABS(X1)
*
* DCNPK1
*
* BX3      X3*X2      . TEST INPUT
* SX4      B1
* MX2      10-18
* SX7      B2
* BX7      -X2*X7      . WATCH OUT FOR B2<0
* LX4      18
* SX2      B7      . SAVE B7
* LX2      36
* SB7      DCNPK2      . RETURN FROM DIGIT
* BX7      X4+X7
* BX7      X2+X7      . SAVE B7,B1,B2: VFD 6/0,18/B7,18/B1,18/B2
* ZR      X3,DIGIT      . CONVERT NUMBER IN X1
* AX7      36
* SX4      X7      . DCN.FAIL EXPECTS OLD B7 IN X4
* JP      DCN.FAIL      . NUMBER TOO BIG
*
* DIGIT RETURNS CHARCOUNT IN B3, PACKED DECIMAL NUMBER IN X1
*
100 63270      10277      21252      63120
101 6170000102 +

*
* DCNPK2
*
* SB2      X7      . RETRIEVE B1 AND B2
* BX2      X7
* AX2      18
* SB1      X2
*
* CALL      COUNT      . COUNT RETURNS THE REAL NUMBER OF

```

```

*
*
*
102 76310          37323          10477          21444
DECNPK20 SX3      R1          . MAX ALLOWABLE CHARS
          IX3      X2-X3
          BX4      X7
          AX4      R6          . X4=OLD RETURN FOR DCNPKFIN
103 7233777776    0323000136 +  SX3      X3-1          . IN CASE B1=X2 (WHICH IS OK)
          PL       X3,DCN.FAIL          . JP IF NUMBER TOO BIG
*
* BEGIN ADDING CHARACTERS. CHECK FOR MINUS SIGN FIRST
*
104 43270          15312          17332
          MX2      A0-4
          BX3      -X2*X1          . ISOLATE FIRST CHAR
          BX3      -X2-X3          . COMPARE CHAR WITH MASK
105 0313000110 +  NZ          X3,DCNPK4          . JP IF NO MINUS SIGN
          AX1      4
106 6133777776    21184          SB3      R3-1          . DECREMENT COUNT
*
*
107 6170000110 +  7170000015          SX7      ASCII MIN          . ADD A MINUS SIGN
          CALL     PKASCII
*
*
110 0720000117 +  DCNPK4 LT          R2,B0,DCNPK6.0          . JP IF SCALING FACTOR < 0
*
* OUTPUT B3 DIGITS
*
          CALL     NUMPK,B3          . INSTALL B3 CHARS
*
* ADD ANY NECESSARY ZEROS
*
112 66120          CALL     ZROPK,B2          . INSTALL B2 ZEROS
*
114 7170000016          SX7      ASCII PD          . ADD PERIOD (DECIMAL POINT)
115 6170000116 +  CALL     PKASCII
116 0200000135 +  JP          DCNPKFIN
*
* CHECK IF THERE ARE LEADING ZEROS. SCALING FACTOR WAS < 0
*
117 46000          DCNPK6.0 NO          0
          67202          SB2      B2          . B2=AB2(B2)
          0723000127 +  LT          B2,B3,DCNPK7.0          . NO LEADING ZEROS
120 7170000016          SX7      ASCII PD          . ADD DECIMAL POINT
121 6170000122 +  CALL     PKASCII
*
* ADD ANY LEADING ZEROS
*
122 67123          CALL     ZROPK,B2-B3          . ADD LEADING ZEROS
*
* INSERT THE DIGITS
*
124 66130          CALL     NUMPK,B3          . INSERT B3 DIGITS

```

```

126 0200000135 +          JP      DCNPKFIN
      *
      * DECIMAL POINT IS IN MIDDLE OF DIGITS
      *
127 67132          DCNPK7.0 CALL    NUMPK,B3-B2      . INSERT DIGITS BEFORE
      *                                     DECIMAL POINT
      *
131 7170000016          SX7      ASCIIPO . ADD DECIMAL POINT
132 6170000133 +          CALL    PKASCII
      *
133 66120          CALL    NUMPK,R2 . OUTPUT REMAINING DIGITS IN NUMBER
      *
      * DONE
      *
135 63740          DCNPKFIN SB7      X4      . RESTORE R7 FROM X4S BEWARES
      *                                     JP      R7
      *
      * FAIL
      *
136 0601000135 +          DCN.FAIL LE      R1,B0,DCNPKFIN
      *                                     SB1
137 7170000012          SX7      B1-1
      *                                     ASCIISTR
      *                                     SB7      DCN.FAIL
140 0200000240 +          JP      PKASCII

```

```

*
* DCBPK
*
* CONVERT A NUMBER FROM OCTAL TO DECIMAL.
* RIGHT JUSTIFY, BLANK FILL.
*
* INPUT
* X1=OCTAL NUMBER (MAX 48 BITS)
* B1=FIELD SIZE
* B2=POWER OF TEN SCALING FACTOR
* B7=RETURN
*
* CALLS DECNOBK. USES SIMILAR REGISTER ALLOCATION
*
141 43214      10311      0323000143 +
142 13333      37331
143 11332      76470      76210
144 6170000150 +      76720      43052
145 15770      20222      26444      12747
146 12727      0303000141 +      21444
147 0200000136 +
*
* DIGIT RETURNS CHAR COUNT IN B3, PACKED DECIMAL NUMBERS IN X1
*
150 63270      10377      21322      63130
151 6170000152 +
152 46000      10477      63220      67112
153 0710000157 +

```

DCBPK MX2 12 . MAX SIZE MASK
BX3 X1
PL X3,DCBPK1 . JP IF INPUT > 0
BX3 X3-X3
IX3 X3-X1 . X3=ABS(X1)
*
DCBPK1 BX3 X3*X2 . ISOLATE BAD BITS, IF ANY
SX4 B7 . SAVE B7
SX2 B1 . AND B1
SB7 DCPK2 . NEW RETURN
SX7 B2 . SAVE B2, TOO
MX0 60-18
BX7 X0*X7 . WATCH OUT FOR B2 < 0
LX2 18
LX4 36
BX7 X4*X7
BX7 X2*X7 . X7=VFD 6/0.18/B7.18/B1.18/B2
*
ZR X3.DIGIT . CONVERT INPUT
AX4 36 . RIGHT JUSTIFY RETURN FOR DCN.FAIL
JP DCN.FAIL . INPUT TOO BIG
*
DCBPK2 SB2 X7 . COPY SAVED REGISTERS, RESTORE B2,B1
BX3 X7
AX3 18
SB1 X3
*
CALL COUNT . COUNT RETURNS MIN NUM CHARS NECESSARY
IN X2
NO 0
*
BX4 X7 . COPY X7, BLNPKK WILL Clobber IT.
SB2 X2
SB1 B1-B2 . B1 =NUM UNUSED SPACES IN FIELD
LT B1,B0,DCB.FAIL . JP IF INPUT REQUIRES TOO
MANY REGISTERS

154	6170000155 +		CALL	BLNPKR	. OUTPUT B1 BLANKS
		*			
155	63240		SB2	X4	. RESTORE R2
	21422		AX4	18	
	63140		SB1	X4	. AND B1
	20422		LX4	18	. REPOSITION FOR DECNPK (WILL BE RT SHFTED)
156	10744		BX7	X4	. OLD RETURN IN X7 (USED BY DECNPK)
	0200000102 +		JP	DECNPK20	. FINISH UP

* FIELD IS TOO SMALL FOR SIZE OF INPUT

157	21422		DCB.FAIL	AX4	18	
	63140		SB1	X4	18	. RESTORE B1
	21422		AX4	18	18	. X4=OLD B7
160	0200000136 +		JP	RCN.FAIL		. PRINT OUT ASTERICKS

```

*
* DIGIT
* CONVERT AN OCTAL NUMBER TO A PACKED DECIMAL NUMBER
*
* INPUT
*       X1=OCTAL NUMBER TO CONVERT
*
* OUTPUT
*       X1=PACKED DECIMAL NUMBER, 4 BITS/DECIMAL DIGIT
*       RIGHT JUSTIFIED, ZERO FILLED, WITH 178 A TOKEN
*       WHICH REPRESENTS A MINUS SIGN. DIGITS ARE IN
*       REVERSE ORDERSS
*
*       R3=TOKEN COUNT (INCLUDES MINUS SIGN, EXCLUDES
*       ANY DECIMAL POINT)
*
* USES X0=CURRENT X1/10
*       X2=10, SCRATCH
*       X3=SCRATCH
*       X4=TEMP STORAGE FOR PACKED DECIMAL NUMBER
*       B1=MINUS SIGN FLAG
*       B2=SCRATCH REG FOR INTEGER DIVISION
*       B3=TOKEN COUNTER
*
*       303240 BIG EQU 100000 . BIG TOKEN COUNT FOR ERROR STATE
*
161 76400 DIGIT SX4 00 . ASSUME INPUT > 0
      66100 SB1 00
      66300 SB3 00
162 0321000164 + PL X1,DIG0
      6110777776 SB1 1 . INPUT < 0
163 46000 NO 10000
      43200 MX2 0
      37121 IX1 X2-X1 . MAKE INPUT POSITIVE
*
164 7120000012 DIG0 SX2 10 . INITIALIZE
      43014 MX0 12
      11001 BX0 X0*X1 . TEST SIZE OF ABS(INPUT)
165 0300000167 + ZR X0,DIGLOOP
      6130303240 SB3 BIG
166 0270000000 JP B7
*
167 0301000176 + DIGLOOP ZR X1,DIG1
      27101 IDIV X0,X1,X2,B2 . X0=X1/X2 (INTEGER)
171 26101 UX1 X1 . MAKE X1 INTEGER AGAIN
      7120000012 SX2 10
      27000 IMPY X2,X0,X2 . X2=X0*X2 (INTEGER, PACKS X0)
      37312 IX3 X1-X2 . X3=X1 MODULO 10
*
173 20404 LX4 4
      12443 BX4 X4+X3 . ADD NEW DIGIT
*
      24100 UX1 X0 . UNDO EVIL DONE BY ↑IMPY↑

```

```

174 6133000001          SB3      R3+1
                        SX2      10      . RESET X2
                        JP        DIGLOOP
175 0200000167 +
*
*
*
*      CHECK FOR NEGATIVE NUMBER
*
176 0610000203 +      DIG1      GE      B1,B0,DIGRTN1      . DONE (NOT <0)
                        0703000200 +      GT      B3,B0,DIG2      . JP IF CHARCOUNT > 0
*
*      INPUT = -0
*
177 46000          NO      0
                        SB3      1
200 20404          DIG2      LX4      4      . ZERO COUNTS AS A DIGIT....
                        SX2      17B     . ROOM FOR MINUS SIGN TOKEN
                        BX4      X2+X4
201 6133000001          SB3      R3+1      . INSERT MINUS SIGN
202 10144          DIGRTN     BX1      X4      . PUT PACKED DIGITS IN X1
                        JP        R7
                        0270000000
*
203 0703000202 +      DIGRTN1    GT      B3,B0,DIGRTN      . DONE IF INPUT .NE. +0
                        6130000001    SB3      1
204 0200000202 +      JP        DIGRTN

```

```

*
* NUMPK
* ADD PACKED DECIMAL DIGITS TO AN ASCII LINE
*
* INPUT
*      X1=WD OF PACKED DECIMAL DIGITS, RIGHT JUSTIFIED,
*      NO MINUS SIGN TOKEN (17B)
*      R1=DIGIT COUNT
*      R7=RETURN
*
* OUTPUT
*      X1 CIRCULAR LEFT SHIFTED 4 BITS/DIGIT
*
* USES
* A0,X1,X2,X7,R1,R7
* CALLS
* PKASCII
*
205 46000      NUMPK      NO      0
      43270      MX2      60-4
      5107000000      SA0      R7+0
*
* MAIN LOOP
*
206 0601000211 *      NUMPK1  LE      B1,B0,NUMPK2      . DONE
      6176000206 +      SB7      NUMPK1      . RETURN
207 15712      BX7      -X2*X1      . ISOLATE CURRENT DIGIT
      20170      LX1      60-4      . RIGHT JUSTIFY NEXT DIGIT
      7277000020      SX7      X7+ZROASC      . COMPUTE ASCII VALUE
210 6111777776      SB1      B1-1      . DECREMENT LOOP COUNT
      0206000240 +      JP      PKASCII      . DO IT
*
211 64700      NUMPK2  SB7      A0
      0270000000      JP      B7      . RETURN

```



```

*
* COUNT
*
* THE NUMBER OF DIGITS ACTUALLY REQUIRED BY A DECIMAL NUMBER
*
* INPUT
*
* B1=MAX NUMBER OF DIGITS TO USE
* B2=POWER OF 10 TO BE APPLIED TO THE NUMBER
* B3=NUMBER OF DIGITS IN THE NUMBER (INCLUDING MINUS
*     SIGN, EXCLUDING DECIMAL POINT)
* B7=RETURN
*
* OUTPUT
*
* Y2=REAL NUMBER OF REQUIRED CHARACTERS
*
* USES X2,X3,X4
*
212  7123000001  COUNT  SX2      B3+1      . DIGIT STRING+DECIMAL POINT
      0720000214 +  LT      B2,B0,COUNT1  . DECIMAL MUST BE MOVED LEFT
*
213  73222      SX2      Y2+B2      . NUMBER OF ZEROS, IF ANY, WHICH MUST BE ADDED
      0270000000  JP      B7
*
* SEE IF ANY ZEROS MUST BE ADDED TO LEFT OF DIGIT STRING
*
214  67202      COUNT1  SB2      B2      . GET ARS(B2)
      0732000216 +  GT      B2,B3,COUNT2  . JP IF MUST ADD ZEROS
*
215  0270000000  67202  SB2      B2      . RESET B2
      JP      B7
*
* COMPUTE NUMBER OF ZEROS WHICH MUST BE ADDED
*
216  76330      COUNT2  SX3      B3
      76420      SX4      B2
      37343      IX3      X4-X3      . NUM ZEROS
      36223      IX2      X2+X3      . NEW TOTAL CHARS
217  67202      SB2      B2      . RESET B2
      0270000000  JP      B7      . RETURN

```

```

*
*
*
*
*
220 56070          BLNKPK  SA0  R7
*
221 0601000224    BLNKPK1 LE    R1,B0,BLNKPK2
      6111777776      SB1  R1-1
222 7170000000      SX7  BLNKASC
      6170000221 +      SB7  BLNKPK1
223 0200000240 +      JP    PKASCII
*
224 64700          BLNKPK2 SB7  R0
      0270000000      JP    R7

```

```

*
* ZROPK
*
* INSTALL ZEROS IN AN ASCII LINE
*
* INPUT      B1=ZERO COUNT
* USES       A0,B7,X7
*
* CALLS      PKASCII
*
225 46000      ZROPK      NO      0
      5107000000      SA0      R7+0      . SAVE RETURN
*
226 0601000231 +      ZROPK1  LE      B1,B0,ZROPK2      . JP IF DONE
      6111777776      SB1      B1-1      DECREMENT COUNT
227 7170000020      SX7      7ROASC
      6170000226 +      SB7      ZROPK1
230 0200000240 +      JP      PKASCII      . GO DO IT
*
231 64700      ZROPK2  SB7      A0
      0270000000      JP      R7      . RETURN

```

THIS PACKAGE FORMS ASCII LINES

FORM OF A LINE IS A WORD CONTAINING PSEUDO CHARACTER
COUNT, COMPRESSED BLANKS COUNTED AS 2 CHARS, FOLLOWED
BY WORDS CONTAINING THE SYSTEXT.

IT HAS 4 ENTRY POINTS

STASCII SET UP
B4 = 1ST WORD OF BUFFER (CHAR COUNT WORD)
B5 = TOTAL WORDS IN BUFFER (INCL CHAR CNT)
B6 = 0 NO BLANK COMPRESSION
NE 0 BLANK COMPRESSION

PKASCII PACK A CHARACTER FROM X7
BOTTOM 7 BITS ALL THAT ARE LOOKED AT
(IGNORED IF BUFFER TOO FULL)

FNASCII FINISH THE LINE WITHOUT CR

CRASCII FINISH THE LINE WITH CR

THIS PACKAGE MAINTAINS 1ST STATE IN REGISTERS
AS FOLLOWS

A5 ADDRESS OF CHAR COUNT WD
X5 ALLOWED NO OF CHARS BEFORE BUFF TOO FULL
A6 ADDRESS OF WD BEFORE NEXT WORD TO STORE
X6 PARTIAL PACKED WORD

B4 BLANK COMPRESSION CONTROL
+1 NO BLANK COMPRESSION
0 NO PREVIOUS BLANKS
-128+K K PREVIOUS BLANKS

B5 PSEUDO CHARACTER COUNT
B6 BITS LEFT IN PARTIAL PACKED WORD
B7 RETURN

X7 CHARACTER WILL GET MODIFIED
X0 SCRATCH

FREE REGISTERS ARE

A0,A1,X1,A2,X2,A3,X3,A4,X4, B1,B2,B3

```

*
*
*
0      BLNKASC EQU 000B      BLANK
173    BLNKFLG EQU 173B      BLANK COUNT FLAG
155    CRASC   EQU 155B      CR
20     ZROASC  EQU 20B      ZERO
*
41     ASCIIIA EQU 41B
42     ASCIIIB EQU 42B
44     ASCIIID EQU 44B
72     ASCIIIZ EQU 72B
20     ASCIIIO EQU 20B
31     ASCIII9 EQU 31B
0      ASCIIIBLN EQU 000B
16     ASCIIIPD EQU 16B
12     ASCIIISTR EQU 12B
155    ASCIIICR EQU 155B
*
13     ASCIIIPUS EQU 13B
15     ASCIIIMIN EQU 15B
12     ASCIIITMS EQU 12B
17     ASCIIIDIV EQU 17B
10     ASCIIILFN EQU 10B
11     ASCIIIRPN EQU 11B
*
3      ASCIIINMB EQU 03B
32     ASCIIICLN EQU 32B

```

INITIALIZE ASCII PACK ROUTINES

```

*
*
*
232 56540      STASCII SA5 04      A5 POINTS TO CHAR COUNT
      56640      SA6 04      A6 NOW PNTS TO WD BEFORE 1ST PKED WD
      7155777776 SX5 05-1     NUMBER OF WORDS TO HOLD ACHARAS
233 20503      LX5 3         MAX CHAR COUNT
      7255777774 SX5 05-3     ALLOW ROOM FOR 2 BLNKS + CR
234 7160000011 SX6 11B      INITIALIZE 1ST PACKED WORD
      66500      SB5 00      INITIALIZE PSEUDO CHARACTER COUNT TO 0
*
235 6140000001 SB4 1         INDICATE NO BLANK COMP
236 0460000237 + EQ 06,00,++1 NO BLANK COMPRESSION IT IS
      66400      SB4 00      NOPE, ITS TO BE BLANK COMPRESSION
237 6160000070 + SB6 56      INITIALIZE PARTIAL WORD BIT COUNT
      0270000000 JP 07

```

```

*
*
*
*
240 0335000252 + 43045 15770 PKASCII NG
                                MX0
                                BX7
                                GT
                                X5,PKASCII4
                                60-7
                                X0*X7
                                R4,B0,PTASCII
                                NOT ENOUGH ROOM
241 0704000262 + 0300000246 + PKASCII1 SX0
                                ZR
                                EQ
                                X0,PKASCII2
                                R0,B4,PTASCII
                                CHECK FOR BLANK
                                HAVE BLANK
                                NOT A BLANK + NO PREVIOUS BLANK
243 0404000262 + 0200000267 + * SX0
                                JP
                                SB4
                                R0
                                PTASCII
                                FORCE OUT BLANKS
                                SET TO NO PREV BLANKS
                                PUT THE CHARACTER
244 7100000245 + 0200000262 + *
                                PKASCII2 EQ
                                SB4
                                LT
                                R0,B4,PKASCII3
                                R4+1
                                R4,B0,PKASCII4
                                R4-1
                                PKASCII3
                                FCASCII
                                FORCE OUT BLANKS
245 66400 0200000262 + *
                                PKASCII3 SB4
                                -128+1
                                SET TO 1 BLANK
251 6140777600 *
                                PKASCII4 JP
                                R7
                                EXIT
252 0270000000

```

```

*
*
*
*
253 0640000255 + FNASCII GE B4,B0.FNASCII NO PREVIOUS BLNKS
      76700      SX7      B0      CLEAR X7
254 7100000255 + * SX0      *+1
      0200000267 + JP      FCASCII      FORCE OUT PREV BLANKS
*
255 22666 FNASCII1 LX6      B6,X6
      5066000001 SA6      A6+1
      76650      SX6      B5
256 54650      SA6      B5
      76500      SX5      B0
      0270000000 JP      B7

```

```

*
*
*
*
257 76070          CRASCI  SX0      B7
      7170000155      SX7      CRASC      PUT OUT CR
260 6170000261 +    +      SB7      8+1
      0200000262 +    JP      PTASCI
261 63700          SB7      X0
      0200000255 +    JP      FNASCI1    FINISH UP

```



```

*
*
*
*
THIS IS BASIC CHARACTER PUT ROUTINE
262 20607 12667 PTASCII LX6 7
        6166777770 BX6 X6+X7 INSTALL CHARACTER
263 7255777776 SB4 R6-7 DECREMENT BIT COUNT
        6155000001 SX5 X5-1 DECREMENT INSTALLED CHARACTER COUNT
264 0406000265 EQ R5+1 INCREMENT PSEUDO CHAR COUNT
        0270000000 R0,B6,PTASCII1 FULL WD
        JP R7
*
265 5066000001 PTASCII1 SA6 A6+1 FULL WD
        76600 SB6 R0 SET WDEMPY
266 6160000070 SB6 S6 SET BIT COUNT
        0270000000 JP R7

```

```

*
*
*
*
*
*
267 20022 12007 20022 76770 FCASCI1 LX0 18 . SAVE
BX0 X0+X7 . B7
LX0 18 . AND
SX7 R7 . X7
BX0 X0+X7 . IN X0
*
270 12007
*
271 0604000277 6144000176 SB4 R4+128=2
LE R4,B0,FCASCI12 2 OR LESS BLANKS
*
272 6170000273 7170000173 SX7 BLNKFLG BLANK FLAG
SB7 *+1
JP BTASCI1
273 7174000002 SX7 R4+2 BLANK COUNT
274 6170000275 SB7 *+1
JP BTASCI1
*
275 63700 FCASCI11 SB7 X0 RESET
20052 LX0 X0-18 B7
73700 SX7 X0 AND X7
20052 LX0 X0-18
276 63400 SB4 X0
0240000000 JP R4
*
*
*
*
COUNT 2 OR LESS
*
277 7170000000 FCASCI12 SX7 BLNKASC BLANK
300 6170000301 SB7 *+1
JP BTASCI1
301 0740000275 LT R4,B0,FCASCI11 COUNT WAS ONLY 1
6170000275 SB7 ECASCI1
302 0200000262 JP BTASCI1 PUT OUT ONE MORE BLANK

```

THIS ROUTINE GENERATES ASCII CHARS FOR A COROUTINE
THE CHARS COME FROM AN ASCII LINE

INITIAL CALL

CALLR SETGENA, LINERUFFER, LOCAL

WHERE LOCAL IS 2 CELLS TO CONTAIN
INTERMEDIATE STATE

GENERATION CALL

CALLR GENASC, LOCAL, COROUTINE

FOR EACH CHARACTER THE COROUTINE IS CALLED VIA

+ SB7 *+7

JP B2

WITH THE CHARACTER IN X7

IF THE LINE WAS EMPTY, A CR IS RETURNED

DURING THE CO ROUTINE CALL A1, A2, X1, X2, B1, B3
ARE IN USE
AND X0 IS ALSO USED FOR SCRATCH

THE CO ROUTINE RETURNS WITH A NEW CALL ADDRESS IN B2,
OR 0 IF STOP, AND -1 IF STOP BUT REPEAT SAME LAST
CHAR ON NEXT START UP

FORM OF LOCAL STORAGE

FIRST WD VFD 2/0, 18/BLANKCOUNT,
18/ADDRESSCURRENTWD,
4/CHARSLEFTINCURRENTWD,
18/TOTALCHARSLEFT

SECOND WORD CURRENT WORD, PARTIALLY SHIFTED

303 76210
20226
56110
73110
304 12612
56620
0200000000 x

SETGENA SX2
LX2
SA1
SX1
BX6
SA6
RTNR

B1 BLANK CNT = 0, ADDRS CNT WD = B1
4+18 CHARS LEFT IN CRNT WD = 0
B1
X1 TOTAL CHARS LEFT
X1+X2
B2

305	56210		* GENASC	SA2 SB1 LX2 MX1 BX1 SB3 LX2 SA1 LX2 SX2 SA3 BX1	R1 X2 60-18 60-4 X1*X2 X1 60-4 X2 60-18 X2 42+1 X3	PICK UP PACKED STATE WD TOTAL CHARS LEFT
	63120					
	20252					
	43170					
306	15121					CHARS LEFT IN CURRENT WD
	63310					
	20270					ADDRESS CURRENT WD
	53120					
307	20252					BLANK COUNT
	73220					
	5032000001					ACTUAL CURRENT WD
310	10133					
			* GENASC1	ZR SX7 SB7 JP	X2, GENASC2 BLNKASC *+1 R2	BLANK COUNT EXHAUSTE
311	0302000315	+				
	7170000000					
312	6170000313	+				
	0220000000					
			* GENASC2	LT SX2 GE JP	R2, B0, GENASC6 X2-1 B0, B2, GENASC6 GENASC1	STOP AND REPEAT LAST CHAR STOP AND NO REPEAT
313	0720000335	+				
	7222777776					
314	0602000335	+				
	0200000311	+				
			* GENASC3	LE GT SA1 SB3 LX1	B1, B0, GENASC7 B3, B0, GENASC3 A1+1 B 4+7	NO CHARS, GENERATE CR STILL MORE IN CURRENT WD GET A NEW WORD POSITION IT
315	0601000342	+				
	0703000320	+				
316	5011000001					
	6130000010					
317	20113					
			* GENASC3	MX0 BX7 SX2 NZ	60-7 X0*X1 X7-BLNKFLG X2, GENASC5	GOT A CHAR NOT A BLANK FLAG
320	43065					
	15710					
	7227777604					
321	0312000330	+				
			* GENASC2	SB1 SB3 LX1 LE GT SA1 SB3 LX1	B1-1 B3-1 Z B1, B0, GENASC6 B3, B0, GENASC4 A1+1 B 4+7	ALL DONE STILL MORE IN CURRENT WD
322	6133777776					
	20107					
323	0601000335	+				
	0703000326	+				
324	5011000001					
	6130000010					
325	20113					
			* GENASC2	MX0 BX2 SB1 SB3 JP	60-7 X0*X1 B1-1 B3-1 GENASC1	GOT A BLANK COUNT PROCESS BLANKS
326	43065					
	15210					
	6111777776					
327	6133777776					
	0200000311	+				
			* GENASC5	SB7 JP	*+1 R2	
330	6170000331	+				
	0220000000					

331	76200		SX2	R0	JUST IN CASE OF STOP
	0720000335 +		LT	R2,B0,GENASC6	STOP AND REPEAT LAST CHAR
332	6111777776		SB1	B1-1	
	6133777776		SB3	B3-1	
333	20107		LX1	7	
	0602000335 +		GE	R0,B2,GENASC6	STOP. NO REPEAT
334	0200000315 +		JP	GENASC2	
335	10711	* GENASC6	BX7	X1	
	5072000001		SA7	A2+1	SAVE CURRENT WD
	73220		SX2	X2	
336	20222		LX2	L8	SAVE LINK COUNT
	74110		SX1	A1	
	12212		BX2	X1+X2	SAVE ADDRESS CURRENT WD
	20204		LX2	4	
337	76130		SX1	R3	
	43070		MX0	60-4	
	15110		BX1	X0*X1	
	12212		BX2	X1+X2	CHARS LEFT IN CURRENT WD
340	20222		LX2	L8	
	76110		SX1	R1	TOTAL CHARS LEFT
	12712		BX7	X1+X2	
	54720		SA7	A2	
341	0200000000 x		RTNR		
342	7170000155	* GENASC7	SX7	CRASC	NO CHARS LEFT, GENERATE CR
343	6170000344 +	+	SB7	B+1	
	0220000000		JP	R2	
344	0702000342 +	*	LT	R0,B2,GENASC7	THE FOOL WANTS EVEN MORE
	0200000335 +		JP	GENASC6	

```

*
*
*      EXTRACT NEXT PARAM FORM ASCII LINE
*
*      CALLR NXTPRM,LOCAL
*
*      WHERE LOCAL IS A PREVIOUSLY SET UP LOCAL
*      STORAGE FOR GENASC
*
*      RESULT IS PLACED AT PARAMBF
*
345 66110      NXTPRM  CALLR  GENASC,B1,NXTPRM0
350 0200000000 X      RTNR
*
*      PACKING CO ROUTINE
*
351 66270      NXTPRM0  SB2    B7
                        SB4    PARAMBF
352 6150000000 X      SB5    PRMBFSZ
                        SB6    B0
353 6170000354 +      CALL  STASCII
354 66720      SB7    B2
*
355 7237777622      NXTPRM1  SX3   X7-CRASC
                        ZR      X3,NXTPRM4
356 7237777777      SX3   X7-BLNKASC
                        NZ      X3,NXTPRM2
357 6120000355 +      SB2   X7-CRASC
                        JP      X3,NXTPRM1
                        X7-CRASC
*
*      SKIP INITIAL BLANKS
*
360 66270      NXTPRM2  SB2    B7
361 6170000362 +      CALL  STASCII
362 66720      SB7    B2
                        SB2    NXTPRM3
363 0270000000      JP      B7
*
*      SAVE RETURN
*
364 7237777777      NXTPRM3  SX3   X7-BLNKASC
                        ZR      X3,NXTPRM4
365 7237777622      SX3   X7-CRASC
                        NZ      X3,NXTPRM2
                        X7-CRASC
*
*      PACK THE CHARACTER
*
366 66270      NXTPRM4  SB2    B7
367 6170000370 +      CALL  CRASCII
370 66720      SB7    B2
                        SB2    B0
                        JP      B0
                        STOP SIGNAL
                        B7

```

DSPLYW

DSPLYW SB5
CALLR
SX4
SB4
CALLR
BX1
LT
SX2
RTNR

DISPLAYW

SX2
RTNR

DSPLYWS

MX0
BX7

LX7
SA5
BX7
LX7
SB6
LX7
SB6
SB6
LX5
LX5
BX5

ZR
LT
LE
LX5
BX4

DSPLYW3

SB2
JP

```

CONVERT ASCII LINE AT LINEBUF TO A
WORD OF DISPLAY CODE, LEFT JUSTIFIED, ZERO FILLED
IN X1

```

```
x2 WILL = 0 IF OK
          = 1 IF MORE THAN MAXCNT OR 10 CHAPS
```

CALLR DSPLYW,LINEBUFF,MAXCNT

B0
B0
B0 GENA,BI,DSPLYWI
B0
B0-6
B0 GENASC,DSPLYWL,DSPLYW2
B0
B0 B0,DSPLYWI TOO MANY CHARS
B0

NO ROUTINE

60-7
-X0*X7

60-3 BEGIN FANCY XLATION

X7 + TT YC Dc
X0 * X7

CHAR SHIFT
8 * CHAR SHIFT

7 * CHAS SIFT

96.X5
4+7
-X0*X5 THE CHARACTER

```

X5,DSPLYW5
B4,B0,DSPLYW4
B5,B0,DSPLYW4
B4,X5
X4+X5
ZERO, STOP
FULL, SO TOO MANY
USER COUNT EXCEEDED, SO TOO MANY
SHIFT AND
PACK THE CHAR

```

84-6
85-1

DISPLYW2
B7

5	5	66	71	60	53	74	67	64
5	1	52	47	45	56	46	57	50
3	3	34	35	36	37	40	41	42
4	3	44	63	77	72	54	73	75
7	6	01	02	03	04	05	06	07
1	0	11	12	13	14	15	16	17
2	0	21	22	23	24	25	26	27
3	0	31	32	61	76	62	70	65
4	0	01	02	03	04	05	06	07
1	0	11	12	13	14	15	16	17
2	0	21	22	23	24	25	26	27
3	0	31	32	51	52	55	55	55
4	5	55	55	55	55	55	55	55
5	5	55	55	55	55	00	55	55
6	5	55	55	55	55	55	55	55
7	5	55	55	55	55	55	55	55

447

116 SYMBOLS
301 REFERENCES

3 TYPE ERROR
OCCURRED ON PAGES

DUPLICATE MACRO DEFINITION. NEW ONE OVERRIDES.
2

COMMAND PROCESSOR DECK C
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

09/07/71 03.03.47.

PAGE 43

ADDCHAR	436	PROGRAM*	2/22 E	41/08 L						
ASCIIA	41		29/09 D							
ASCIIB	42		29/10 D							
ASCIIBLN	0		29/15 D							
ASCIICLN	32		29/28 D							
ASCIICR	155		29/18 D							
ASCIID	44		29/11 D							
ASCIIDIV	17		29/23 D							
ASCIILPN	10		29/24 D							
ASCIIMIN	15		18/21	29/21 D						
ASCIINMB	3		29/27 D							
ASCIIPD	16		18/35	18/44	19/09	29/14 D				
ASCIIPLS	13		29/20 D							
ASCIIRPN	11		29/25 D							
ASCIISTR	12		19/23	29/17 D						
ASCIITMS	12		29/22 D							
ASCIIZ	72		29/12 D							
ASCIIO	20		29/13 D							
ASCIIO	31		29/14 D							
BIG	303240		22/25 D	22/40						
BLNKASC	0		15/26	26/11	29/04 D	30/10	34/34	36/16	38/25	38/34
BLNKFLG	173		29/05 D	34/17	36/33					
BLNKPK	220	PROGRAM*	2/18 E	12/53	21/02	26/07 L				
BLNKPK1	221	PROGRAM*	26/09 L	26/12						
BLNKPK2	224	PROGRAM*	26/09	26/15 L						
CALLR	0	EXTERNAL*	38/12	39/15	39/18					
COUNT	212	PROGRAM*	2/19 E	18/01	20/44	25/17 L				
COUNT1	214	PROGRAM*	25/18	25/25 L						
COUNT2	216	PROGRAM*	25/26	25/33 L						
CRASC	155		29/06 D	32/06	37/26	38/23	38/38			
CRASCI	257	PROGRAM*	2/17 E	32/05 L	38/43					
DCBPK	141	PROGRAM*	2/19 E	20/14 L						
DCBPK1	143	PROGRAM*	20/16	20/20 L						
DCBPK2	150	PROGRAM*	20/23	20/38 L						
DCB.FAIL	157	PROGRAM*	20/20	21/12 L						
DCNPKFIN	135	PROGRAM*	18/37	19/01	19/16 L	19/21				
DCNPK1	73	PROGRAM*	17/25	17/30 L						
DCNPK2	100	PROGRAM*	17/38	17/48 L						
DCNPK4	110	PROGRAM*	18/16	18/25 L						
DCNPK6.0	117	PROGRAM*	18/25	18/41 L						
DCNPK7.0	127	PROGRAM*	18/43	19/05 L						
DCN.FAIL	136	PROGRAM*	17/44	18/09	19/21 L	19/24	20/34	21/15		
DECNPK	71	PROGRAM*	2/19 E	17/23 L						
DECNPK20	102	PROGRAM*	18/04 L	21/08						
DIGIT	161	PROGRAM*	2/19 E	17/41	20/32	22/27 L				
DIGLOOP	167	PROGRAM*	22/39	22/43 L	23/03					
DIGRTN	202	PROGRAM*	23/20 L	23/23	23/25					
DIGRTN1	203	PROGRAM*	23/09	23/23 L						
DIG0	164	PROGRAM*	22/30	22/36 L						
DIG1	176	PROGRAM*	22/43	23/09 L						
DIG2	200	PROGRAM*	23/10	23/16 L						
DISASC	14	PROGRAM*	10/40	11/17 L						
DISLINE	0	PROGRAM*	2/18 E	10/20 L	13/07					

PAGE 44

[illegible]

COMMAND PROCESSOR DECK C
SYMBOLIC REFERENCE TABLE.

COMPASS - VER 2.

09/07/71 03.03.48.

PAGE 45

PRMBFSZ	0	EXTERNAL*	38/18				
PTASCII	262	PROGRAM*	30/08	30/16	33/05 L	34/22	34/39
			30/12	32/08	34/19	34/36	
PTASCII1	265	PROGRAM*	33/10	33/13 L			
RTNR	0	EXTERNAL*	35/53	37/24	38/12	39/21	39/24
SETGENA	303	PROGRAM*	2/20 E	35/47 L	39/15		
STASCII	232	PROGRAM*	2/17 E	29/41 L	38/21		
TTYCDC	416	PROGRAM*	39/32	40/18 L			
ZROASC	20		14/25	16/29	24/27	27/15	29/07 D
ZROPK	225	PROGRAM*	2/19 E	18/34	18/50	27/10 L	
ZROPK1	226	PROGRAM*	27/13 L	27/16			
ZROPK2	231	PROGRAM*	27/13	27/19 L			