

PDP-5 PROGRAM LIBRARY

December 17, 1964

*A later version will directly
reference DEC's new System
Numbers ~~and~~*

ORIGINAL PROGRAM	SUPERCEDED BY	NAME
DEC	DIGITAL	
5-1-S	8-3-S	† PAL (Program Assembly Language for PDP-5)
5-2-A	8-11-F	Two's Complement Multiply Subroutine
5-3-0	8-18-U	Teletype Output Package
5-4-A	8-10-U	BCD to Binary Conversion
5-5-S	8-12-S	ODT(Octal Debugging Tape)
5-7-A	8-9-F	Square Root (Single Precision)
5-8-U	8-6-U	Octal Dump on Teletype (High)
5-9-A	8-12-F	2's Complement Single Precision Divide Subroutine
5-10-A		Signed Decimal to Binary Input
5-11-A		Double Precision Binary-Dec. Conversion and Output (Signed)
5-17-A	8-15-F	Single Precision Sine Calculation
5-18-A		Single Precision Cosine Calculation
5-20-U		Message-5
5-21-10	8-22-U	Unsigned Decimal Print, DCPT
5-22-1	8-1-U	Rim & Bin Loader
5-23-1	8-2-U	†Binary Loader 750
5-24-0	8-5-U	Binary Punch 33
5-25-0	8-5-U	†Binary Punch 75A
5-26-0	8-4-U	Read-in-Mode (RIM) Punch 33
5-29-1		†RIM Loader 750
5-30-A	8-5-S	Interpretive Floating Point Arithmetic Package
5-31-U		Signed Single Precision, Decimal Memory Dump (High)

ORIGINAL PROGRAM	SUPERCEDED BY	NAME
DEC	DIGITAL	
5-32-A		Single Precision Binary to Decimal Conversion and Typeout (Signed)
5-33-A	8-13-F	Double Precision Multiply
5-34-A	8-14-F	Double Precision Divide
5-36-A	8-5-S	PDP-5 Flooding Point and I/O Package
5-37-S	8-1-S	PDP-5 Paper Tape Editor
5-41-U		Octal Dump on Teletype under Program Control
5-42-U		Signed, Single Precision, Decimal Memory Dump on Teletype under Program Control
5-43-D	8-10-S	Expensive Adding Machine
5-46-A		Square Root Calculation in Floating Point
5-48-M		<i>34 DISPLAY TEST</i>
5-53-I/O	8-33-U	DECtog-5
	MAINDEC	Maintenance Routines
5-12-M	801	Maindec 501 (Instruction Test)
5-13-M	810	Maindec 510 (Reader Test Program)
5-14-M	812	Maindec 512 (Punch Test)
5-15-M	802	Maindec 502 (Memory Checkerboard)
5-16-M	803	Maindec 503 (Address Test Program)
5-19-M	814	Maindec 514 (Teleprinter Test Program)
5-38-M	814	PDP-5 Read Alpha Test

ORIGINAL PROGRAM	SUPERCEDED BY	Maintenance Routines
DEC	MAINDEC	
5-40-M	811	†High Speed Reader Test, Type 750
5-44-M	817	†High Speed Punch Test
5-50-M	820	Memory Extend Test
5-52-M		Teletype 634S Test
5-63-I/O		5/8 TOG - DECTOG
5-54-M		57A Basic Program
5-55-M		Drum 250

† Programs for High Speed Reader and Punch

PDP-5 PROGRAM LIBRARY

NUMBER: DEC - 5 - 10 - A

NAME: Signed Decimal to Binary Input

DATE: Revised June 18, 1964

AUTHOR: Larry Portner - DEC

SPECS: Length: 64_{10} Registers - TS

ABSTRACT: This routine will accept a string of decimal digits from the teletype and convert them to a signed two's complement binary number (if a sign is specified).

The string may contain a sign (-or space) and the digits from 0-9; a rubout will erase the word and allow the operator to re-enter the value. The sign, if one appears, must appear first. Any character other than -, space, 0-9 or rubout will cause the routine to terminate, as will a sign in any but the first position. The character which causes the routine to terminate will be found in location "save". If the first legal character is not a sign (space or-) the sign of the conversion will be positive.

DESCRIPTION

This routine will convert a signed or unsigned string of decimal numbers read from the teletype keyboard to its binary equivalent. If a minus sign is specified, the result is in two's complement form.

The first character is examined, and if it is sign (space or -), a switch is set to provide the correct sign for the conversion.

Succeeding characters are then examined. A rubout will wipe out intermediate results and allow a restart. Any non-digit, including a sign, will cause the routine to terminate. Upon the return, the binary value will be in the accumulator. The character which caused termination will be in "save", and a zero in test indicates no conversion has taken place.

The user must initialize the keyboard flag if there is a possibility the flag may be on before the input string has been started.

/decimal to binary input conversion routine
 /routine accepts a string of digits, a space or a minus sign
 /routine will terminate on any other character
 /character which terminates will be in save.
 /calling sequence: jms dbcv
 / return after terminator. result in ac

```

*200
dbcv,      0
           cla cll
           dca test
           tad sevt
           dca swch
           dca temp
           jms chin
           dca save
           tad save
           tad rbt
           sna
           jmp dbcv+1 / sp. bc. + 1 + 2 + 3 + 4 + 5 + 6 + 7
           tad spc / 137
           sna
           jmp tst
           tad neg
           sza
           jmp skip
           tad test
           sza cla
           jmp exit
pos,      tad fort
           tad sevt
           dca swch
           isz test
           jmp dbcv+6
skip,     cla
           tad save
           tad numb
           sma
           jmp exit
           tad nine
           sma
           jmp cont
exit,     cla
           tad temp
swch,     0
           jmp i dbcv
cont,     dca sav2
           tad temp
           rtl cll
           ral

```



```
oct1,    dca save
         tad temp
         ral
         tad save

ret,     tad sav2
         isz test
         jmp dbcv+5

tst,     tad test
         sza cla
         jmp exit
         jmp pos

nine,    12
temp,    0
save,    0
sav2,    0
rbt,     7401
spc,     0137
test,    0
neg,     7763
fort,    0041
sevt,    7000
numb,    7506
chin,    0
         ksf
         jmp .-1
         krb
         tls
         jmp i chin

$
```

*200

DBCV, 0
CLA CLL
DCA TEST
TAD SEVT
DCA SWCH
DCA TEMP
JMS CHIN
DCA SAVE
TAD SAVE
TAD RBT
SNA
JMP DBCV+1
TAD SPC
SNA
JMP TST
TAD NEG
SZA
JMP SKIP
TAD TEST
SZA CLA
JMP EXIT
TAD FORT
POS, TAD SEVT
DCA SWCH
ISZ TEST
JMP DBCV+6
SKIP, CLA
TAD SAVE
TAD NUMB
SMA
JMP EXIT
TAD NINE
SMA
JMP CONT
EXIT, CLA
TAD TEMP
SWCH, 0
JMP I DBCV
CONT, DCA SAV2
TAD TEMP
RTL CLL
RAL
OCTL, DCA SAVE
TAD TEMP
RAL
TAD SAVE
RET, TAD SAV2
ISZ TEST
JMP DBCV+5
TST, TAD TEST
SZA CLA
JMP EXIT
JMP POS

NINE, 12
TEMP, 0
SAVE, 0
SAV2, 0
RBT, 7401
SPC, 0137

TEST, 0
NEG, 7763
FORT, 0041
SEVT, 7000
NIMB, 7506
CHIN, 0
KSF
JMP .-1
KRB
TLS
JMP I CHIN

S

PDP-5 PROGRAM LIBRARY

NUMBER: DEC-5-11-A

NAME: Double Precision Binary-Decimal Conversion and Output (Signed)

AUTHOR: Larry Portner

DATE: November 27, 1963

SPECS: Length: 134₈
TS (Teletype Symbolic)

NEEDED: Self-contained

ABSTRACT: This routine will convert two binary words (where the high order word contains a sign) to a signed decimal number and type it out.

DESCRIPTION

Given the address of the high order word of two words which comprise a 23 bit binary number and a 1 bit sign, this routine will convert the binary value to its decimal equivalent and type it out, preceded by a space or a minus sign.

USAGE

Calling sequence:

```
jms dbnd  
address of high order word  
return:
```

Routine will type a sign +7 digits. User must provide formatting.

```
*200
DBND,      0
           CLA CLL
           TAD I DBND
           DCA GET
           TAD I GET
           SPA
           CML
           CLA
           TAD PLUS
           SZL
           TAD MNS
           JMS PRNT
           TAD I GET
           CLL
           SPA
           CMA CML
           DCA HIGH
           TAD LOOP
           DCA CNT
           TAD ADDR
           DCA Z 0010
           ISZ GET
           ISZ DBND
           TAD I GET
           SZL
           CMA CLL IAC
           SZL
           ISZ HIGH
           DCA LOW
ARND,      TAD I Z 0010
           DCA HSUB
           JMS DO
           ISZ CNT
           JMP ARND
           TAD PLUS
           JMS PRNT
           JMP I DBND
DO,        0
           TAD I Z 0010
           DCA LSUB
           CLL
           TAD LSUB
           TAD LOW
```

	DCA TEMPL
	RAL
	TAD HSUB
	TAD HIGH
	SPA
	JMP OUT
	ISZ BOX
	DCA HIGH
	TAD TEMPL
	DCA LOW
	JMP DO+3
OUT,	CLA
	TAD BOX
	JMS PRNT
	DCA BOX
	JMP I DO
PRNT,	0
	TAD TWO
	TSF
	JMP .-1
	TLS
	CLA
	JMP I PRNT
LOOP,	7771
CNT,	0
HIGH,	0
LOW,	0
ADDR,	CON1-1
TWO,	0260
PLUS,	7760
MNS,	0015
HSUB,	0
LSUB,	0
BOX,	0
TEMPL,	0
GET,	0
CON1,	7413
	6700
	7747
	4540
	7775
	4360
	7777

6030
7777
7634
7777
7766
7777
7777

\$

/DOUBLE PRECISION BINARY TO DECIMAL CONVERSION AND TYPEOUT
*200

0200	0000	DEND,	0
0201	7300	CLA	CLL
0202	1600	TAD	I DEND
0203	3316	DCA	GET
0204	1716	TAD	I GET
0205	7510	SPA	
0206	7020	CML	
0207	7200	CLA	
0210	1310	TAD	PLUS
0211	7430	SZL	
0212	1311	TAD	MNS
0213	4273	JMS	PRNT
0214	1716	TAD	I GET
0215	7100	CLL	
0216	7510	SPA	
0217	7060	CMA	CML
0220	3304	DCA	HIGH
0221	1302	TAD	LOOP
0222	3303	DCA	CNT
0223	1306	TAD	ADDR
0224	3010	DCA	Z 0010
0225	2316	ISZ	GET
0226	2200	ISZ	DEND
0227	1716	TAD	I GET
0230	7430	SZL	
0231	7141	CMA	CLL IAC
0232	7430	SZL	
0233	2304	ISZ	HIGH
0234	3305	DCA	LOW
0235	1410	ARND,	TAD I Z 0010
0236	3312	DCA	HSUB
0237	4245	JMS	DO
0240	2303	ISZ	CNT
0241	5235	JMP	ARND
0242	1310	TAD	PLUS
0243	4273	JMS	PRNT
0244	5600	JMP	I DEND
0245	0000	DO,	0
0246	1410	TAD	I Z 0010
0247	3313	DCA	LSUB
0250	7100	CLL	
0251	1313	TAD	LSUB
0252	1305	TAD	LOW
0253	3315	DCA	TEML
0254	7004	RAL	
0255	1312	TAD	HSUB
0256	1304	TAD	HIGH
0257	7510	SPA	
0260	5266	JMP	OUT
0261	2314	ISZ	BOX
0262	3304	DCA	HIGH
0263	1315	TAD	TEML
0264	3305	DCA	LOW
0265	5250	JMP	DO+3
0266	7200	OUT,	CLA
0267	1314	TAD	BOX
0270	4273	JMS	PRNT
0271	3314	DCA	BOX
0272	5645	JMP	I DO

0273	0000	FRNT, 0
0274	1307	TAD TWO
0275	6041	TSF
0276	5275	JMP .-1
0277	6046	TLS
0300	7200	CLA
0301	5673	JMP I FRNT
0302	7771	LOOP, 7771
0303	0000	CNT, 0
0304	0000	HIGH, 0
0305	0000	LOW, 0
0306	0316	ADDR, CON1-1
0307	0260	TWO, 0260
0310	7760	PLUS, 7760
0311	0015	MNS, 0015
0312	0000	HSUB, 0
0313	0000	LSUB, 0
0314	0000	BOX, 0
0315	0000	TEML, 0
0316	0000	GET, 0
0317	7413	CON1, 7413
0320	6700	6700
0321	7747	7747
0322	4540	4540
0323	7775	7775
0324	4360	4360
0325	7777	7777
0326	6030	6030
0327	7777	7777
0330	7634	7634
0331	7777	7777
0332	7766	7766
0333	7777	7777
0334	7777	7777

ADDR	0306
ARND	0235
BOX	0314
CNT	0303
CON1	0317
DBND	0200
DO	0245
GET	0316
HIGH	0304
HSUB	0312
LOOP	0302
LOW	0305
LSUB	0313
MNS	0311
OUT	0266
PLUS	0310
PRNT	0273
TEML	0315
TWO	0307

PDP-5 PROGRAM LIBRARY

NUMBER: DEC - 5 - 31 - U

NAME: Signed, Single Precision, Decimal Memory Dump (High)

DATE: Writeup revised June 30, 1964

AUTHOR: Larry Portner - DEC

SPECS: Length 128_{10} Registers
ASR-33, PAL Binary Format
SA 7000; Occupies 7000-7177.

ABSTRACT: Given the limits of an area in memory, this routine will perform 2's complement binary to signed decimal conversion upon each word, and type it out, a variable number of words per line with the address of the first word in each line appearing in front of each line.

DESCRIPTION:

This routine will convert a series of 2's complement binary words to signed decimal and type them out, giving the user the option of specifying the area of memory to be converted and the number of words to be typed on each line. The maximum number per line is 11_8 .

USAGE:

- A) Start the routine by placing the starting address (7000) in the console switches and pressing LOAD ADDRESS.
- B) Enter the octal number of words per line in the console switches and press START.
- C) The program will immediately halt. Enter the beginning address of the area to be dumped in the switches, press CONTINUE.
- D) Enter the ending address in the switches, press CONTINUE.

To dump successive blocks using the same number of words per line, repeat C and D.

To change the number of words per line, repeat A through D.

/single precision decimal memory dump
*7000

```
                t1s
                jms lfcr
                osr
                sna
                jmp .+3
                cia
                dca parm
                jmp ddmp
sbnc,           0
                spa
                cml
                dca bal
                dca box
                tad cntr
                dca cnt
                tad addr
                dac xyz+2
                tad plus
                szl
                tad mns
                jms prnt
                tad val
                spa
                cma iac
xyz,           dca val
                tad val
                tad con
                sma
                isz box
                sma
                jmp xyz
                cla
                tad box
                jms prnt
                dca box
                isz xyz+2
                isz cnt
                jmp xyz+1
                tad plus
                jms prnt
                tad plus
                jms prnt
                jmp i sbnc
addr,          tad con
box,           0
cnt,           0
val,           0
cntr,         7774
con,          6030
                7634
                7766
                7777
```

```
prnt,      0
           tad two
           tsf
           jmp .-1
           tls
           cla cll
           jmp i prnt
two,       0260
plus,     7760
mns,     0015
ddmp,
           cla
           tls
           hlt
           osr
           dca bgn
           hlt
           osr
           dca end
           tad bgn
           cma iac
           cll
           tad end
           cma
           dca end
           cll
newl,     tad form
           dca olup
           tad parm
           dca line
           tad bgn
           jms ocpt
loop,     tad i bgn
           jms sbnc
           isz bgn
           isz end
           jmp .+3
           jms lfer
           jmp ddmp
           isz line
           jmp loop
           jms lfer
           cla
           jmp newl
msk4,    7000
form,    7774
line,    0
end,     0
bgn,     0
olup,    0
```

```
ocpt,      0
           cll
           dca wait
           tad msk4
           and wait
           rtl
           rtl
           jms prnt
           tad wait
           rtl
           ral
           isz olup
           jmp ocpt+1
           cla
           tad plus
           jms prnt
           cla
           jmp i ocpt
lfer,      0
           tad cr
           jms prnt
           tad lf
           jms prnt
           jmp i lfer
cr,        7735
lf,        7732
wait,      0
parm,     0-4
           $
```

PDP-5 PROGRAM LIBRARY

NUMBER: DEC - 5 - 32 - A

NAME: Single Precision Binary to Decimal Conversion and typeout
(signed.)

AUTHOR: Larry Portner - DEC

DATE: January 16, 1964

SPECS: Length: 51₁₀ Registers
TS (ASR-33, PAL Symbolic)

ABSTRACT: This routine will convert a two's complement binary word to it's signed decimal equivalent and type it out.

DESCRIPTION:

This routine is entered with a two's complement binary word in the accumulator and will convert it to its signed decimal equivalent and type it out, preceded by a space or a minus sign.

USAGE:

Calling sequence:

```
tad binary word
jms sbnc
return
```

```
/binary to decimal conversion and typeout  
/single precision  
/calling sequence:   tad binary word  
/  
/                   jms sbnc  
/                   return
```

```
sbnc,      0  
           cll  
           spa  
           cml  
           dca val  
           dca box  
           tad cntr  
           dca cnt  
           tad addr  
           dca xyz+2  
           tad plus  
           szl  
           tad mns  
           jms prnt  
           tad val  
           spa  
           cma iac  
xyz,       dca val  
           tad val  
           tad con  
           sma  
           isz box  
           sma  
           jmp xyz  
           cla  
           tad box  
           jms prnt  
           dca box  
           isz xyz+2  
           isz cnt  
           jmp xyz+1  
           jmp i sbnc  
addr,     tad con  
box,      0  
cnt,      0  
val,      0  
cntr,     7774  
con,      6030  
           7634  
           7766  
           7777
```

```
prnt,      0  
           tad two  
           tsf  
           jmp .-1  
           t1s  
           cla c11  
           jmp i prnt  
two,       0260  
plus,     7760  
mns,     0015  
           $
```


/BINARY TO DECIMAL CONVERSION AND TYPEOUT
 /SINGLE PRECISION
 /CALLING SEQUENCE: TAD BINARY WORD
 / JMS SBNC
 / RETURN

*200

0200	0000	SBNC, 0
0201	7100	CLL
0202	7510	SPA
0203	7020	CML
0204	3243	DCA VAL
0205	3241	DCA BOX
0206	1244	TAD CNTR
0207	3242	DCA CNT
0210	1240	TAD ADDR
0211	3223	DCA XYZ+2
0212	1261	TAD PLUS
0213	7430	SZL
0214	1262	TAD MNS
0215	4251	JMS FRNT
0216	1243	TAD VAL
0217	7510	SPA
0220	7041	CMA IAC
0221	3243	XYZ, DCA VAL
0222	1243	TAD VAL
0223	1245	TAD CON
0224	7500	SMA
0225	2241	ISZ BOX
0226	7500	SMA
0227	5221	JMP XYZ
0230	7200	CLA
0231	1241	TAD BOX
0232	4251	JMS FRNT
0233	3241	DCA BOX
0234	2223	ISZ XYZ+2
0235	2242	ISZ CNT
0236	5222	JMP XYZ+1
0237	5600	JMP I SBNC
0240	1245	ADDR, TAD CON
0241	0000	BOX, 0
0242	0000	CNT, 0
0243	0000	VAL, 0
0244	7774	CNTR, 7774
0245	6030	CON, 6030
0246	7634	7634
0247	7766	7766
0250	7777	7777
0251	0000	FRNT, 0
0252	1260	TAD TWO
0253	6041	TSF
0254	5253	JMP .-1
0255	6046	TLS
0256	7300	CLA CLL
0257	5651	JMP I FRNT
0260	0260	TWO, 0260
0261	7760	PLUS, 7760
0262	0015	MNS, 0015

ADDR	0240
BOX	0241
CNT	0242

CNTF 0244
CON 0245
MNS 0262
PLLS 0261
PRNT 0251
SBNC 0200
TWO 0260
VAL 0243
XYZ 0221

PDP-5 PROGRAM LIBRARY

NUMBER: DEC - 5 - 41 - U

NAME: Octal Dump on Teletype under Program Control

DATE: April 1, 1964

AUTHOR: R. Winslow, J.E. Richardson - DEC

SPECS: Length 76₁₀ Registers
PAL Symbolic

NEEDED: ASR-33

ABSTRACT: Subroutine to type out the contents of an area in
memory as octal numbers.

DESCRIPTION

This subroutine will convert a series of 2's complement binary numbers to their octal equivalence and type them out on the ASR-33. In the calling sequence, the user specifies the area of memory to be typed out.

The first number on each line is the address in octal of the following word. Four words are typed out following each address. A carriage return/line feed combination is generated both prior to and after the typeout.

Calling Sequence

.
. .
. .
jms odum /subroutine called
IA /address of initial word
FA /address of final word
. /return with AC and
. /link clear.
.

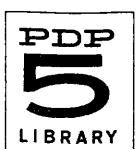


/PDP-5 OCTAL DUMP SUBROUTINE
x200

```

pnum,      0
           dca dac ptem
           tad pcon           /7774
           dca dcx           /initialize digit counter
           tad ptem
           ral
pnu2,      ral
           rtl
           dca ptem
           tad ptem
           and pcon+1        /7
           tad pcon+2        /260
           jms i tdit        /typn (type a digit)
           tad ptem
           isz dcx
           jmp pnu2
           cla
           jmp i pnum
den,       0           /digit counter
ptem,     0
tdit,     typn
pcon,     7774        /constants
           7
           260
odum,     0
           jms crlf
           tad i odum        /get lower limit
           dca lock
           isz odum
           tad i odum        /get upper limit
           isz odum
           cma
           tad lock
           dca lim           /initialize range counter
           jmp .+3
dum2,     isz lpcn        /end of line
           jmp dum3
           tad pcon          /-4
           dca lpcn         /reset item counter
           jms crlf         /carr,ret.and line feed
           tad lock
           jms pnum         /inter-com. to pnum
           tad pcon          /-4
           dca crlf
           tad cons+2        /240 (space)
           jms typn
           isz crlf
           jmp .-3

```



```
dum3,      cla
           tad i lock
           jms pnum          /inter-com. to pnum
           tad cons+2       /240 (space)
           jms typn
           isz lock         /index location pointer
           isz lim         /end of range
           jmp dum2
           jms crlf
           jmp i odum       /return to main program

typn,      0
           tsf
           jmp .-1
           tls
           cla
           jmp i typn

crlf,      0
           cla cll
           tad cons+1
           jms typn
           tad cons         /212 (line feed)
           jms typn
           jmp i crlf

lock,      0
lim,        0
lpcn,      0
cons,      212             /constants
           215
           240

$
```

PDP-5 PROGRAM LIBRARY

NUMBER: DEC - 5 - 42 - U

NAME: Signed, Single Precision, Decimal Memory Dump
on Teletype under Program Control

DATE: April 1, 1964

AUTHOR: Larry Portner, J.E. Richardson - DEC

SPECS: Length 126_{10} Registers - PAL Symbolic

NEEDED: ASR-33

ABSTRACT: Subroutine to type out the contents of an area in
memory as signed, decimal numbers.



DESCRIPTION

This subroutine will convert a series of 2's complement binary words to decimal and type them out on the ASR-33. The user specifies the area of memory to be typed out and the number of words per line.

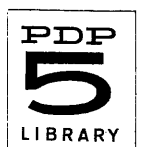
The first number on each line is the address in octal of the following word. The maximum number of words per line which can be specified is 118. A carriage return/line feed combination is given both prior to and after the typeout.

Calling Sequence:

```
      .  
      .  
      .  
tad N      /N= number of words per line  
           /in octal  
jms ddum   /subroutine called  
  
IA        /address of initial word  
FA        /address of final word  
      .   /return. AC and link clear  
      .  
      .
```

Note: If the AC contains zero when the subroutine is called, the most recent specification of number of words per line will be taken.

The subroutine itself is set up to type out 4 words per line unless otherwise specified.




```
/single precision decimal memory dump on teletype.  
/under program control  
/calling sequence:  
/tad n      /"N" = number of words per line  
/jms ddum  /subroutine called  
/ia        /initial address to be typed  
/fa        /final address to be typed  
x200  
ddum,      0  
           sna  
           jmp .+3  
           cia  
           dca parm  
           jms lfer  
           jmp ddmp  
sbnc,      0  
           spa  
           cml  
           dca val  
           dca box  
           tad cntr  
           dca cnt  
           tad addr  
           dca xyz+2  
           tad plus  
           szl  
           tad mns  
           jms prnt  
           tad val  
           spa  
           cma iac  
xyz,       dca val  
           tad val  
           tad con  
           sma  
           isz box  
           sma  
           jmp xyz  
           cla  
           tad box  
           jms prnt  
           dca box  
           isz xyz+2  
           isz cnt  
           jmp xyz+1  
           tad plus  
           jms prnt  
           tad plus  
           jms prnt  
           jmp i sbnc
```

```
addr,      tad con
box,       0
cnt,       0
val,       0
cntr,      7774
con,       6030
           7634
           7766
           7777
prnt,      0
           tad two
           tsf
           jmp .-1
           tls
           cla cll
           jmp i prnt
two,       0260
plus,      7760
mns,       0015
ddmp,      tad i ddum
           dca bgn
           isz ddum
           tad i ddum
           isz ddum
           dca end
           tad bgn
           cma iac
           cll
           tad end
           cma
           dca end
           cll
newl,      tad form
           dca olup
           tad parm
           dca line
           tad bgn
           jms ocpt
loop,      tad i bgn
           jms sbnc
           isz bgn
           isz end
           jmp .+3
           jms lfer
           jmp i ddum
           isz line
           jmp loop
           jms lfer
           cla
           jmp newl
```

```

msk4,      7000
form,      7774
line,      0
end,       0
bgn,       0
olup,      0
ocpt,      0
           cll
           dca wait
           tad msk4
           and wait
           rtl
           rtl
           jms prnt
           tad wait
           rtl
           ral
           isz olup
           jmp ocpt+1
           cla
           tad plus
           jms prnt
           cla
           jmp i ocpt
lfer,      0
           tad cr
           jms prnt
           tad lf
           jms prnt
           jmp i lfer
cr,        7735
lf,        7732
wait,      0
parm,      0-4
$

```