



DECUS

PROGRAM LIBRARY

DECUS NO.	8-607
TITLE	CALCUI
AUTHOR	J. V. Hopson
COMPANY	Bureau of Customs 2100 K Street N. W. Washington, D. C.
DATE	March 1, 1973
SOURCE LANGUAGE	PAL-8

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

CALCUI

DECUS Program Library Write-up

DECUS NO. 8-607

ABSTRACT

Makes the PDP-8 perform like a printing calculator, with addition, subtraction, multiplication, division, and exponentiation. Prints out subtotals and totals on command. Recognizes control/C for return to monitor. Utilizes one of the DEC floating point packages (EAE--if so equipped, NON-EAE, or 27-BIT). Introductory dialog gives essential operating instructions.

CORE REQUIREMENTS

THE PROGRAM ITSELF USES LOCS. 0-1154. A RUNNING PROGRAM WILL USE THIS, PLUS THE CORE AREAS REQUIRED FOR THE PARTICULAR FLOATING POINT PACKAGE UTILIZED.

STARTING ADDRESS: 00200

LOADING AND STARTING

FOR PAPER TAPE OPERATION, FIRST LOAD THE BINARY TAPE FOR THE FLOATING POINT PACKAGE TO BE USED, THEN THE BINARY FOR CALCUI. START AT 0200.

TO BUILD FROM TAPES ON OS/8 OR PS/8, USE THE FOLLOWING COMMAND SEQUENCE:

```
.R.ABSLDR
*SYS:EAEFPP (FOR INSTANCE) [CR/LF]
*PTR:$
.SA SYS CALCUI
```

IF YOU HAVE THE OS/8 DECTAPE VERSION, THINGS ARE MUCH SIMPLER: JUST MOUNT THE DECTAPE ON A CONVENIENT DRIVE AND CALL UP CALC1A.SV (THE EAE VERSION), CALC1B.SV (THE NON-EAE VERSION), OR CALC1C.SV (THE 27-BIT VERSION).

PROGRAM OPERATION

CONDENSED OPERATING INSTRUCTIONS ARE GIVEN IN A DIALOG WHICH OPENS THE PROGRAM. FOR ANYONE NOT FAMILIAR WITH THE TERM, ARITHMETIC OPERATORS, FOR THIS PURPOSE, ARE +, -, *, /, AND $\uparrow$. "E14.6 FORMAT" IS FORTRAN TALK, AND MEANS A NUMBER 14 CHARACTERS WIDE WITH 6 DIGITS TO THE RIGHT OF THE DECIMAL POINT. ACTUALLY, THE FLOATING POINT PACKAGES WILL ACCEPT DECIMAL NUMBERS IN ALMOST ANY FORM; YOU DON'T REALLY HAVE TO WORRY ABOUT IT.

IF YOU SHOULD ATTEMPT A NO-NO, LIKE DIVIDING BY ZERO, OR TRYING TO EXPONENTIATE A NON-POSITIVE NUMBER, AN APPROPRIATE ERROR MESSAGE WILL BE PRINTED, AND THE PROGRAM WILL CLEAR ITSELF.

AT ANY TIME THE PROGRAM IS WAITING FOR INPUT, IT WILL RECOGNIZE CONTROL/C AND TRANSFER CONTROL TO 07600 FOR RETURN TO THE MONITOR, IF YOU'VE GOT ONE.

/CALCUI - VERSION 1

/THIS PROGRAM CAUSES THE PDP-8 TO SIMULATE A PRINTING
/CALCULATOR, CAPABLE OF ADDITION, SUBTRACTION,
/MULTIPLICATION, DIVISION, AND EXPONENTIATION.
/SUBTOTALS AND TOTALS ARE PRODUCED ON COMMAND.
/AN INITIAL DIALOG GIVES ESSENTIAL OPERATING IN-
/STRUCTIONS.

/THIS PROGRAM IS DESIGNED TO UTILIZE ONE OF THE DEC
/FLOATING POINT PACKAGES. THE BINARY FOR THE
/APPROPRIATE FLOATING POINT PACKAGE (EAE, NONEAE, 27-BIT)
/MUST BE LOADED WITH THE BINARY ASSEMBLED FROM
/THIS SOURCE PROGRAM OR RESULTS WILL BE UNPREDICTABLE.

/FPP PSEUDO-INSTRUCTIONS:

1000	FIXMRI	FADD=1000
2000	FIXMRI	FSUB=2000
3000	FIXMRI	FMPY=3000
4000	FIXMRI	FDIV=4000
5000	FIXMRI	FGET=5000
6000	FIXMRI	FPUT=6000
0000		FEXT=0000
0006		FEXP=0006
0007		FLOG=0007
0010		FNEG=0010
0011		FIN=0011
0012		FOUT=0012
0013		FFIX=0013
0014		FLOT=0014
7110		FSMA=7110
7050		FSZA=7050
7100		FSPA=7100
7040		FSNA=7040
7010		FNOP=7010
7020		FSKP=7020
4407		FPP=JMS I 7
4427		PRINT=JMS I TYPE
4426		CRLF=JMS I CRLF1

00020	0020	*20
00020	0000	TEMP1, 0
00021	0000	0
00022	0000	0
00023	0000	TEMP2, 0
00024	0000	0
00025	0000	0
00026	0264	CRLF1, CRLF2
00027	0473	TYPE, TYPE1
00030	0221	GTOP1, GTOP
00031	0044	FAC0, 44
00032	0000	TEMP, 0

	0056		*56	
00056	0001		1	/SET OUTPUT FORMAT OF
00057	0016		16	/FLOATING POINT PKG
00060	0006		6	/TO E14.6
	0070		*70	
00070	0000	GTCHK,	0	/ROUTINE TO INPUT A NUMBER
				/AND CHECK FOR PROPER TER-
				/MINATOR BEFORE PROCEEDING
00071	4407		FPP	
00072	6020		FPUT TEMP1	/SAVE CONTENTS OF FAC
00073	0011		FIN	/GET NO. FROM KEYBOARD
00074	0000		FEXT	
00075	7200		CLA	/AND CHECK FOR VALID INPUT
00076	1053		TAD 53	/C(53=TERMINATOR
00077	1107		TAD KM215	/ASCII FOR C.R.
00100	7450		SNA	/WAS IT A C.R.?
00101	5470		JMP I GTCHK	/YES, EXIT TO MAIN PROG
00102	4407		FPP	/NO, . . .
00103	5020		FGET TEMP1	/RESTORE CONTENTS OF FAC
00104	0000		FEXT	
00105	4426		CRLF	
00106	5430		JMP I GTOP1	/AND TRY AGAIN
00107	.7563	KM215,	-215	
00200	6046	START,	TLS	
00201	6032		KCC	
00202	4671		JMS I INIT	
00203	4672		JMS I INIT3	
00204	7300		CLA CLL	
00205	4407		FPP	
00206	5311		FGET K0	
00207	6020		FPUT TEMP1	
00210	6023		FPUT TEMP2	
00211	0000		FEXT	
00212	4427		PRINT	
00213	4543		TEXT "%#CLEAR%##"	
00214	0314			
00215	0501			
00216	2245			
00217	4343			
00220	0000			
00221	7200	GTOP,	CLA	
00222	6031		KSF	
00223	5222		JMP .-1	
00224	6036		KRB	/GET OPERATOR
00225	3032		DCA TEMP	
00226	1032		TAD TEMP	
00227	1307		TAD KM203	
00230	7650		SNA CLA	/CTRL/C?
00231	5710		JMP I MON	/YES, BACK TO MONITOR
00232	1032		TAD TEMP	/NO, GO ON
00233	6041		TSF	
00234	5233		JMP .-1	
00235	6046		TLS	/ECHO OPERATOR
00236	1306		TAD KM252	
00237	7450		SNA	/""?
00240	5701		JMP I MULTI	/YES, GO TO MULTIPLY SUB
00241	1305		TAD KM1	/(-253 OVERALL)
00242	7450		SNA	/""?

00243	5700	JMP I ADD	/YES, GO TO ADD SUB
00244	1304	TAD KM2	/(-255)
00245	7450	SNA	/"-"?'
00246	5677	JMP I SUBT	/YES, GO TO SUBTRACT ROUTINE
00247	1304	TAD KM2	/(-257)
00250	7450	SNA	/"-"?'
00251	5676	JMP I DIVD	/YES, GO TO DIVIDE ROUTINE
00252	1303	TAD KM44	/(-323)
00253	7450	SNA	/"S"?'
00254	5675	JMP I STOT	/YES, OUTPUT SUBTOTAL
00255	1305	TAD KM1	/(-324)
00256	7450	SNA	/"T"?'
00257	5674	JMP I GTOT	/YES, OUTPUT TOTAL AND CLEAR
00260	1302	TAD KM12	/(-336)
00261	7650	SNA CLA	/"↑"?'
00262	5673	JMP I EXP	/YES, EXPONENTIATE
00263	5221	JMP GTOP	/NO, IGNORE IT AND TRY AGAIN
00264	0000	CRLF2, 0	
00265	4427	PRINT	
00266	4543	TEXT "%#"	
00267	0000		
00270	5664	JMP I CRLF2	
00271	0600	INIT, INIT1	
00272	1000	INIT3, INIT2	
00273	0430	EXP, EXP1	
00274	1131	GTOT, GTOT1	
00275	1123	STOT, STOT1	
00276	0421	DIVD, DIVD1	
00277	0405	SUBT, SUBT1	
00300	0400	ADD, ADD1	
00301	0414	MULTI, MULTI1	
00302	7766	KM12, -12	
00303	7734	KM44, -44	
00304	7776	KM2, -2	
00305	7777	KM1, -1	
00306	7526	KM252, -252	
00307	7575	KM203, -203	
00310	7600	MON, 7600	
00311	0000	K0, 0	
00312	0000	0	
00313	0000	0	
00400	4070	ADD1, JMS GTCHK	/GET AND CHECK OPERAND:
			/RETURN WITH PREVIOUS C(FAC IN
			/TEMP1 AND OPERAND IN FAC.
00401	4407	FPP	
00402	1020	FADD TEMP1	
00403	0000	FEXT	/EXIT WITH TOTAL IN FAC
00404	5430	JMP I GTOP1	
00405	4070	SUBT1, JMS GTCHK	
00406	4407	FPP	
00407	6023	FPUT TEMP2	/SUBTRAHEND IN TEMP2
00410	5020	FGET TEMP1	
00411	2023	FSUB TEMP2	
00412	0000	FEXT	
00413	5430	JMP I GTOP1	

```

00414 4070 MULTI1, JMS GTCHK
00415 4407 FPP
00416 3020 FMPY TEMP1
00417 0000 FEXT
00420 5430 JMP I GTOP1

00421 4070 DIVD1, JMS GTCHK
00422 4407 FPP
00423 6023 FPUT TEMP2
00424 5020 FGET TEMP1
00425 4023 FDIV TEMP2
00426 0000 FEXT
00427 5430 JMP I GTOP1

00430 4070 EXP1, JMS GTCHK
00431 4407 FPP
00432 6023 FPUT TEMP2
00433 5020 FGET TEMP1
00434 0007 FLOG
00435 3023 FMPY TEMP2
00436 0006 FEXP
00437 0000 FEXT
00440 5430 JMP I GTOP1

00441 4427 ERR1, PRINT
00442 4543 TEXT "%#CANNOT EXPONENTIATE NEGATIVE NUMBER OR 0%#"
00443 0301
00444 1616
00445 1724
00446 4005
00447 3020
00450 1716
00451 0516
00452 2411
00453 0124
00454 0540
00455 1605
00456 0701
00457 2411

00460 2605
00461 4016
00462 2515
00463 0205
00464 2240
00465 1722
00466 4060
00467 4543
00470 0000
00471 5672 JMP I START1
00472 0204 START1, START+4
00473 0000 TYPE1, 0

```

```

/TYPE MESSAGE FOLLOWING CALL;
/"%#" OUTPUTS A CR/LF COMBO

```

00474	7240		CLA CMA	
00475	1273		TAD TYPE1	
00476	3014		DCA 14	
00477	1414		TAD I 14	/NOTE: LOC 14 IS AUTOINDEX
00500	3032		DCA TEMP	
00501	1032		TAD TEMP	
00502	7012		RTR; RTR; RTR	
00503	7012			
00504	7012			
00505	4311		JMS DECODE	
00506	1032		TAD TEMP	
00507	4311		JMS DECODE	
00510	5277		JMP TYPE1+4	
00511	0000	DECODE,	0	
00512	0345		AND MASK5	/MASK OFF BITS 0-5
00513	7450		SNA	/END OF TEXT?
00514	5414		JMP I 14	/YES, RETURN
00515	1346		TAD KM40	/NO, CHECK IT OUT
00516	7500		SMA	/<40?
00517	5322		JMP DECODE+11	/NO
00520	1347		TAD K340	/YES, ADD 340 (300 TO ORIG.)
00521	5335		JMP PRI	/AND PRINT IT
00522	1350		TAD KM3	/(-43 OVERALL)
00523	7440		SZA	/WAS IT "#"?
00524	5327		JMP DECODE+16	/NO
00525	1351		TAD K212	/YES; GET ASCII FOR L.F.
00526	5335		JMP PRI	/AND PRINT IT
00527	1353		TAD KKM2	/(-45 IN ALL)
00530	7440		SZA	/WAS IT "%"?
00531	5334		JMP DECODE+23	/NO
00532	1352		TAD K215	/YES, GET ASCII FOR C.R.
00533	5335		JMP PRI	/AND PRINT IT
00534	1354		TAD K245	/ADD 245 TO OTHERS
00535	4337	PRI,	JMS TTYP	/AND PRINT
00536	5711		JMP I DECODE	
00537	0000	TTYP,	0	
00540	6046		TLS	
00541	6041		TSF	
00542	5341		JMP .-1	
00543	7200		CLA	
00544	5737		JMP I TTYP	
00545	0077	MASK5,	0077	
00546	7740	KM40,	-40	
00547	0340	K340,	340	
00550	7775	KM3,	-3	
00551	0212	K212,	212	
00552	0215	K215,	215	
00553	7776	KKM2,	-2	
00554	0245	K245,	245	

0600 PAGE

00600	0000	INIT1, 0	/INITIAL DIALOG
00601	4427	PRINT	
00602	0301	TEXT "CALCULATOR PROGRAM;###AFTER 'CLEAR' IS PRINTED,%#"	
00603	1403		
00604	2514		
00605	0124		
00606	1722		
00607	4020		
00610	2217		
00611	0722		
00612	0115		
00613	7245		
00614	4343		
00615	0106		
00616	2405		
00617	2240		
00620	4703		
00621	1405		
00622	0122		
00623	4740		
00624	1123		
00625	4020		
00626	2211		
00627	1624		
00630	0504		
00631	5445		
00632	4300		
00633	4427	PRINT	
00634	0516	TEXT "ENTER #+##	-%# * (TO MULTIPLY)%# "
00635	2405		
00636	2240		
00637	4353		
00640	4543		
00641	4040		
00642	4040		
00643	4040		
00644	5545		
00645	4340		
00646	4040		
00647	4040		
00650	4052		
00651	4050		
00652	2417		
00653	4015		
00654	2514		
00655	2411		
00656	2014		
00657	3151		
00660	4543		
00661	4040		
00662	4040		
00663	4040		
00664	0000		

```

00665 4427 PRINT
00666 5740 TEXT "/" (TO DIVIDE)%# + (TO EXPONENTIATE)%#"
00667 5024
00670 1740
00671 0411
00672 2611
00673 0405
00674 5145
00675 4340
00676 4040
00677 4040
00700 4036
00701 4050
00702 2417
00703 4005
00704 3020
00705 1716
00706 0516
00707 2411
00710 0124
00711 0551
00712 4543
00713 0000
00714 4427 PRINT
00715 4040 TEXT " S (FOR SUBTOTAL), OR%# T (FOR TOTAL);%##"
00716 4040
00717 4040
00720 2340
00721 5006
00722 1722
00723 4023
00724 2502
00725 2417
00726 2401
00727 1451
00730 5440
00731 1722
00732 4543
00733 4040
00734 4040
00735 4040
00736 2440
00737 5006
00740 1722
00741 4024
00742 1724
00743 0114
00744 5173
00745 4543
00746 4300
00747 5600 JMP I INITI
1000 PAGE

```

01000	0000	INIT2, 0
01001	4427	PRINT
01002	0122	TEXT "ARITHMETIC OPERATORS ARE TO BE FOLLOWED BY%#"
01003	1124	
01004	1015	
01005	0524	
01006	1103	
01007	4017	
01010	2005	
01011	2201	
01012	2417	
01013	2223	
01014	4001	
01015	2205	
01016	4024	
01017	1740	
01020	0205	
01021	4006	
01022	1714	
01023	1414	
01024	1727	
01025	0504	
01026	4002	
01027	3145	
01030	4300	
01031	4427	PRINT
01032	0140	TEXT "A DECIMAL NUMBER IN E14.6 FORMAT, AND 'RETURN'%#"
01033	0405	
01034	0311	
01035	1501	
01036	1440	
01037	1625	
01040	1502	
01041	0522	
01042	4011	
01043	1640	
01044	0561	
01045	6456	
01046	6640	
01047	0617	
01050	2215	
01051	0124	
01052	5440	
01053	0116	
01054	0440	
01055	4722	
01056	0524	
01057	2522	
01060	1647	
01061	4543	
01062	0000	

01063	4427	PRINT
01064	4311	TEXT "#IF ERROR IN ENTRY, PRESS 'RUBOUT' INSTEAD OF 'RETUR█
01065	0640	
01066	0522	
01067	2217	
01070	2240	
01071	1116	
01072	4005	
01073	1624	
01074	2231	
01075	5440	
01076	2022	
01077	0523	
01100	2340	
01101	4722	
01102	2502	
01103	1725	
01104	2447	
01105	4011	
01106	1623	
01107	2405	
01110	0104	
01111	4017	
01112	0640	
01113	4722	
01114	0524	
01115	2522	
01116	1656	
01117	4745	
01120	4343	
01121	0000	
01122	5600	JMP I INIT2

```

01123 4407 STOT1, FPP
01124 6020      FPUT TEMP1
01125 0012      FOUT
01126 5020      FGET TEMP1
01127 0000      FEXT
01130 5430      JMP I GTOP1

```

```

01131 4407 GTOT1, FPP
01132 6020      FPUT TEMP1
01133 0012      FOUT
01134 5020      FGET TEMP1
01135 0000      FEXT
01136 5737      JMP I START3
01137 0204 START3, START+4

```

/ERROR TRAPS

```

7575 *7575
07575 1140      ERR2          /DIVISION BY 0
07576 0441      ERR1          /ILLEGAL LOG

```

```

1140 *START3+1
01140 4427 ERR2, PRINT
01141 4543      TEXT "%#CAN'T DIVIDE BY 0%##"
01142 0301
01143 1647
01144 2440
01145 0411
01146 2611
01147 0405
01150 4002
01151 3140
01152 6045
01153 4343
01154 0000
01155 5737      JMP I START3
          $$$$

```

ADD	0300	K215	0552
ADD1	0400	K245	0554
CRLF	4426	K340	0547
CRLF1	0026	MASK5	0545
CRLF2	0264	MON	0310
DECODE	0511	MULTI	0301
DIVD	0276	MULTI1	0414
DIVD1	0421	PRI	0535
ERR1	0441	PRINT	4427
ERR2	1140	START	0200
FXP	0273	START1	0472
EXP1	0430	START3	1137
FAC0	0031	STOT	0275
FADD	1000	STOT1	1123
FDIV	4000	SUBT	0277
FEXP	0006	SUBT1	0405
FEXT	0000	TEMP	0032
FFIX	0013	TEMP1	0020
FGET	5000	TEMP2	0023
FIN	0011	TTYP	0537
FLOG	0007	TYPE	0027
FLOT	0014	TYPE1	0473
FMPLY	3000		
FNEG	0010		
FNOP	7010		
FOUT	0012		
FPP	4407		
FPUT	6000		
FSKP	7020		
FSMA	7110		
FSNA	7040		
FSPA	7100		
FSUB	2000		
FSZA	7050		
GTCHK	0070		
GTOP	0221		
GTOP1	0030		
GTOT	0274		
GTOT1	1131		
INIT	0271		
INIT1	0600		
INIT2	1000		
INIT3	0272		
KKM2	0553		
KM1	0305		
KM12	0302		
KM2	0304		
KM203	0307		
KM215	0107		
KM252	0306		
KM3	0550		
KM40	0546		
KM44	0303		
K0	0311		
K212	0551		

.R CALCUI

CALCULATOR PROGRAM: ~~XXXXXXXXXXXXXXXXXXXX~~ [SAMPLE RUN]

AFTER 'CLEAR' IS PRINTED,
ENTER

+
-
* (TO MULTIPLY)
/ (TO DIVIDE)
↑ (TO EXPONENTIATE)
S (FOR SUBTOTAL), OR
T (FOR TOTAL);

ARITHMETIC OPERATORS ARE TO BE FOLLOWED BY
A DECIMAL NUMBER IN E14.6 FORMAT, AND 'RETURN'

IF ERROR IN ENTRY, PRESS 'RUBOUT' INSTEAD OF 'RETURN.'

CLEAR

+17.775
+.17E2
S +34.775000
-27.992
/2.11
*5.33
S +17.134300
↑2.27
S +632.232000
/0.000001
S***** [FORMAT OVERFLOW--NUMBER IS STILL IN ACCUMULATOR
*0.000001 BUT CAN'T BE PRINTED OUT IN THIS FORMAT]
S +632.231000
/0.00
CAN'T DIVIDE BY 0

CLEAR

↑3.337
CANNOT EXPONENTIATE NEGATIVE NUMBER OR 0

CLEAR

