

SHARE PROGRAM LIBRARY AGENCY



PROGRAM NUMBER

121024

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257

SHARE PROGRAM LIBRARY SUBMITTAL FORM

SHARE PROGRAM LIBRARY AGENCY
Triangle Universities Computation Center
Post Office Box 12076
Research Triangle Park, North Carolina
27709 USA
Attention: Mr. Joe Ragland

SPLA CONTROL NUMBER:

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the "SHARE Program Library Standards Manual".

- (1) Program Number (to be filled in by SPLA) 360D-12.1.024
- (2) System Type (machine) IBM 370-155
- (3) Search Key Interactive Hex Decimal Octal
Calculator
- (4) Programming Language PL1
- (5) Author's Name and Address
Dr. Dianne K. Sakaguchi
The Aerospace Corporation
P. O. Box 92957
Los Angeles, CA 90009
- (6) Direct Inquiries to Name and Address
(if different than Author)
- (7) Title of Program Interactive Hex-Decimal-Octal Calculator
- (8) Submitter's Installation Membership Code..... ASC
- (9) Submitter's Own Program Identification and Suffix(Optional)....
- (10) Primary Subject Code..... 12.1
- (11) Operating or Monitor System Required TSO
- (12) New or Revision Code (if revision, show prior Program Number in Item 1).. N
- (13) Year Completed..... 1974
- (14) Date of Submittal..... 1974
- (15) Documentation (number of original pages submitted)..... 5
- (16) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

SHARE PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide:

- a. Purpose
- b. Programming Language used
- c. Version and modification level or release number
- d. Field of application
- e. Type of routine (main program, subroutine, etc.)
- f. Specific description of machine requirements

ABSTRACT

The program provides TSO with a calculator mode which will accept hexadecimal, octal or decimal integers. It requires no knowledge to use, and has good response time. It is meant to be used to help read dumps, work with the test command to check out programs, provide a method for composing hex to octal tape conversions, and aid in similar tasks where octal or hex numbers are required.

DISCLAIMER

Triangle Universities Computation Center (TUCC) serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, INC., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

(Please attach additional pages if necessary).....Total pages attached _____

Permission to Publish

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program."

(17) Signature of Submitter and Date D.K. Solagucki 7/2/74

(18) Signature of Installation Addressee _____

SAMPLE TERMINAL SESSION

ENTER D,O, OR H AND THE STRING
h 3cf-2d

DECIMAL 930 OCTAL 1642 HEX 3A2

ENTER D,O OR H AND THE STRING
55

FIRST CHARACTER MUST BE D,H OR O

} invalid input... base not indicated

ENTER D,O OR H AND THE STRING
d 55

DECIMAL 55 OCTAL 67 HEX 37

ENTER D,O OR H AND THE STRING
o 55

DECIMAL 45 OCTAL 55 HEX 2D

ENTER D,O OR H AND THE STRING
o 777+18

} invalid input ... '8' is not valid as octal

INVALID CHARACTER

ENTER D,O OR H AND THE STRING
o 7777+1*3

DECIMAL 12288 OCTAL 30000 HEX 3000

ENTER D,O OR H AND THE STRING
h aa/2

DECIMAL 85 OCTAL 125 HEX 55

ENTER D,O OR H AND THE STRING
end

} TO END PROGRAM EXECUTION

READY - TSO response

```

CALC PROCEDURE OPTIONS(MAIN)
/* THIS IS AN ON-LINE HEX-DECIMAL-OCTAL CALCULATOR
   WHICH IS INTENDED TO AID IN DEBUGGING PROGRAMS,
   READING DUMPS, CONVERTING NUMBERS, AND CHECKING
   THE RESULTS OF TAPE CONVERSIONS */
/* THE PROGRAM SHOULD BE COMPILED AND LINKED INTO A LOAD MODULE.
   IT CAN THEN BE EXECUTED BY THE FOLLOWING ISO COMMANDS
   ALLOC FILE(SYSIN) DATASET(*)
   ALLOC FILE(SYSPRINT) DATASET(*)
   CALL LOADMODULE      NOTE THESE MAY BE STORED AS A CLIST */
/* CNCE CALLED THE PROGRAM IS REQUESTS INPUT
   THE FIRST CHARACTER MUST BE D, O OR H INDICATING THE NATURE
   OF THE INPUT (DECIMAL, OCTAL, OR HEX). THIS IS FOLLOWED BY A
   SINGLE NUMBER IF JUST CONVERSION IS REQUIRED OR BY A STRING
   COMPOSED OF NUMBERS SEPARATED BY ARITHMETIC SYMBOLS (THESE
   CAN BE + * / OR -). THERE MUST BE AT LEAST ONE BLANK BETWEEN
   THE FIRST CHARACTER AND THE STRING, AND THE STRING MAY HAVE NO
   EMBEDDED BLANKS. THE RESULT IS RETURNED IN ALL THREE BASES.
   END IS ENTERED TO EXIT THE PROGRAM. */
DECLARE LINE CHAR(72), TBL CHAR(16), ONECH CHAR(1)
/* INPUT LINE      CONVERSION TABLE      TEMPORARY STORAGE */
DECLARE JWRK(10), NUMB FIXED BIN(31), MODN FIXED BIN(31)
/* WORKING STORAGE  NUMBER IN BINARY      INTERMEDIATE RESULT */
DECLARE SAVEN FIXED BIN(31), OCTL CHAR(10), VARYING
/* STORAGE
   CHARACTER STRING OCTAL RESULT */
DECLARE EIGHT FIXED BIN(31) INIT(08), SIXTEEN FIXED BIN(31) INIT(16)
/* NUMBERS USED AS BASES */
DECLARE HEXL CHAR(10) VARYING
/* CHARACTER STRING HEX RESULT */
DECLARE NEASE FIXED BIN(31)
/* WORKING BASE */
DECLARE TBL2 CHARACTER(5)
/* TABLE OF ARITHMETIC OPERATIONS */
TBL = _0123456789ABCDEF_
TBL2 = _+-*/-
/* FIRST TIME THROUGH IS HANDLED DIFFERENTLY */
PUT SKIP LIST (_ENTER D, O, OR H AND THE STRING_) PUT SKIP
GET EDIT (LINE) (A(72))
GO TO IFFFF1
/* CONTROL COMES HERE AFTER EACH RESULT IS GIVEN */
PTL1 PUT SKIP LIST (_ENTER D, O OR H AND THE STRING (OR END)_)
PUT SKIP
GET SKIP EDIT (LINE) (A(72))
/* SEE IF EXIT HAS BEEN REQUESTED */
IF PTL1 IF SUBSTR(LINE, 1, 3) = _END_ THEN GO TO QUIT
IFRASE=0

```

```

/* FIND WHICH BASE THE INPUT REPRESENTS */
IF SUBSTR(LINE,1,1)=D_ THEN IBASE=10
00000355
00000360

IF SUBSTR(LINE,1,1)=H_ THEN IBASE=16
00000370

IF SUBSTR(LINE,1,1)=O_ THEN IBASE=8
00000380
00000390

IF IBASE C THEN GO TO INIT1
00000395

/* IF FIRST CHARACTER WAS ILLEGAL -- ERROR MESSAGE */
PUT SKIP LIST (_FIRST CHARACTER MUST BE D,H OR O_) PUT SKIP
00000400
GO TO PTL1 /* GET INPUT AGAIN */
00000410

/* INITIALIZE AND FIND FIRST CHARACTER OF STRING */
00000420
INIT1 SAVEN=0 IOP=0
00000430
DO I1=2 TO 72
00000440
IF SUBSTR(LINE,I1,1)=_ THEN GO TO E1 ELSE GO TO FIRSTMQ
00000450
E1 END
00000460
GO TO QUIT /* IF STRING IS BLANK PROGRAM ENDS */
00000470

/* SCAN HAS FOUND FIRST CHARACTER */
00000471
FIRSTMQ IF SUBSTR(LINE,I1,1)=_ THEN IOP=-1
00000480
IF SUBSTR(LINE,I1,1)=_ THEN I1=I1+1
00000490

/* THIS PIECE OF CODE SEPARATES NUMBERS */
00000495
FIRSTC DO I2=I1+1 TO 72
00000500
ONECH=SUBSTR(LINE,I2,1)
00000510

NIOP=INDEX(TBL2,ONECH) /* NIOP INDICATES ARITHMETIC OPER */
00000520
IF NIOP 0 THEN GO TO LASTC
00000530
END
00000540

/* CONTROL HERE ONLY IF MESSAGE TOO LONG */
00000545
PUT SKIP LIST(_MESSAGE TOO LONG_) PUT SKIP
00000550
GO TO PTL1
00000560

/* NEXT NUMBER MUST BE CONVERTED FROM CHARACTER TO BINARY */
00000565
LASTC LENG=I2-I1 /* LENG IS LENGTH OF NUMBER */
00000570
K=1
00000580

/* JWRK IS AN ARRAY STORING THE POWERS REPRESENTED BY THE DIGITS */
00000585
DO J=1 TO I2-1
00000590
ONECH=SUBSTR(LINE,J,1)
00000600

JWRK(K)=INDEX(TBL,ONECH)-1
00000610
/* VALID CHARACTER CHECK */
00000615
/* IF INVALID REQUEST NEW INPUT */
00000616

IF JWRK(K)-O JWRK(K)=IBASE THEN PUT SKIP LIST
00000620
{ _ INVALID CHARACTER _ }
00000630
IF JWRK(K)-O JWRK(K)=IBASE THEN GO TO PTL1
00000640

K=K+1
00000650
END
00000660

/* BY MULTIPLYING BY APPROPRIATE POWERS OF BASE (AS STORED IN
00000665
JWRK) THE NUMBER IS CONVERTED TO BINARY */
00000666
NUMB=0
00000670
NRASE=1
00000680

DO J=LENG TO 1 BY -1
00000690
NUMB=NUMB+JWRK(J)*NRASE
00000700

```

```

NBASE=NBASE#IBASE
END
/* CHECK IF NEGATIVE */
IF IOP 0 THEN GO TO IFF2
SAVEN=NUMB IF IOP=-1 THEN SAVEN=-NUMB
GO TO AGN
/* PERFORM REQUESTED OPERATION */
IFF2 IF IOP=1 THEN SAVEN=SAVEN+NUMB
IF IOP=2 THEN SAVEN=SAVEN-NUMB
IF IOP=3 THEN SAVEN=SAVEN*NUMB
IF IOP=4 THEN SAVEN=SAVEN/NUMB
AGN IF NIOP=5 THEN GO TO CONV
/* RESET BEGINNING POINTERS AND SCAN FOR NEXT NUMBER IN STRING */
IOP=NIOP
II=I2+1
GO TO FIRSIC
/* NUMB NOW CONTAINS THE BINARY NUMBER */
/* STRING HAS BEEN PROCESSED --- NOW BINARY MUST BE CONVERTED
BACK TO CHARACTER FOR DISPLAY */
CONV NUMB=SAVEN
IF NUMB=0 THEN NUMB=-NUMB
/* FIRST FIND OCTAL STRING */
/* THIS IS DONE CHARACTER BY CHARACTER SUBTRACTING AND SHIFTING */
OCTL=-
DO I=1 TO 10
MODN=MOD(NUMB,EIGHT)
K=MODN+1
OCTL=SUBSTR(TBL,K,1) OCTL
NUMB=(NUMB-MODN)/8
IF NUMB=0 THEN GO TO ODONE
END
ODONE NUMB=SAVEN
/* OCTAL STRING COMPLETE */
IF NUMB=0 THEN NUMB=-NUMB
/* HEX STRING OBTAINED IN THE SAME MANNER */
HEXL=-
DO I=1 TO 8
MODN=MOD(NUMB,SIXTEEN)
K=MODN+1
HEXL=SUBSTR(TBL,K,1) HEXL
NUMB=(NUMB-MODN)/16
IF NUMB=0 THEN GO TO HDONE
END
/* HEX STRING COMPLETE */
HDONE NUMB=SAVEN
IF NUMB=0 THEN OCTL=- HEXL

```

```

00000710
00000720
00000725
00000730
00000740
00000750
00000755
00000760
00000770
00000780
00000790
00000800
00000805
00000810
00000820
00000830
00000840
00000845
00000846
00000850
00000860
00000865
00000866
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000955
00000960
00000965
00000970
00000980
00000990
00010000
00010100
00010100
00010200
00010300
00010400
00010450
00010500
00010600

```

```
IF NUMB=0 THEN HEXL=-- HEXL
/* STANDARD ROUTINES CONVERT BINARY TO DECIMAL C1SPLAY */
PUT EDIT C1SPLAY, NUMB, OCTAL, HEX, HEXL
(SKIP, A(8), F(8, 0), A(7), A(10), A(4), A(8)) PUT SKIP
/* RETURN TO GET NEXT REQUEST */
GO TO PTL1
QUIT END CALC
00001070
00001074
00001080
00001090
00001095
00001100
00001110
```