

```

*****
* SAMPLE CALLABLE SERVICE PROGRAM. *
* SHARE SESSION: INTRODUCING LE CALLABLE SERVICES *
* THOMAS PETROLINO *
* IBM LANGUAGE ENVIRONMENT *
* TAPETRO@US.IBM.COM *
* THANKS TO JOHN EHRMAN FOR THE CONVE ROUTINE *
*****
          TITLE 'LE CALLABLE SERVICE EXAMPLE SQRT '          00010000
          PRINT GEN                                          00013000
*****
* USE CEEENTRY TO SET UP AN LE ENVIRONMENT *
*****
CSSQRTA  CEEENTRY PPA=MAINPPA,MAIN=YES,BASE=(8,9,10),AUTO=WORKSIZE
          USING WORKAREA,13                                00015100
*****
* SETUP FOR A CALL TO CEEMOUT *
*****
          MVC   MOUTSTRXT,MYTXT  MOVE TEXT OF STRING
          LA    5,14             LENGTH OF TEXT IS 14
          STH   5,MOUTSTRLEN     STORE IT
          LA    5,2             THE MESSAGE DEST IS 2
          ST    5,MOUTDEST      STORE IT
          LA    5,MOUTSTR       BUILD PARMLIST - 1ST PARM IS THE STR
          ST    5,PLISTMOUTP1   STORE ADDR
          LA    5,MOUTDEST      2ND PARM IS MSG DEST
          ST    5,PLISTMOUTP2   STORE ADDR
          LA    5,0             3RD IS FEEDBACK CODE (NULL IGNORE)
          ST    5,PLISTMOUTP3   STORE ADDR
          LA    1,PLISTMOUT     PLIST IN R1
          CALL  CEEMOUT         MAKE THE CALL
*****
* SETUP FOR A CALL TO CESSSQRT (SQUARE ROOT) WITH 9.0 *
*****
          LA    5,FLOAT9        1ST PARM = 9.0
          ST    5,PLISTSQRTP1   STORE ADDR IN PARMLIST
          LA    5,FEEDBACK      2ND PARM = FEEDBACK CODE
          ST    5,PLISTSQRTP2   STORE ADDR IN PARMLIST
          LA    5,RESULT        3RD PARM = RESULT
          ST    5,PLISTSQRTP3   STORE ADDR IN PARMLIST
          LA    1,PLISTSQRT     PLIST IN R1
          CALL  CESSSQRT        CALL SQRT
*****
* CHECK THE RESULT FROM SQUARE ROOT CALL. *
*****
          LH    5,FBNUM         LOAD THE FEEDBACK CODE
          LA    6,0             SEE IF IT IS ZERO (OK)
          CR    5,6             MAKE THE CHECK
          BE    SQRTOK1        IF ZERO SKIP ERROR PATH
*****
* SQUARE ROOT CALL FAILED. *
* SETUP TO CALL CEEMSG WITH FEEDBACK CODE FROM CESSSQRT *
*****
          LA    5,FEEDBACK      PARM 1 IS FEEDBACK CODE
          ST    5,PLISTMSGP1    STORE ADDR IN PARMLIST
          LA    5,MOUTDEST      PARM 2 IS THE MSG DEST
          ST    5,PLISTMSGP2    STORE ADDR IN PARMLIST

```

```

    LA    5,0                PARM 3 IS NEW FEEDBACK CODE
    ST    5,PLISTMSGP3      WE SEND NULL TO IGNORE FAILURES
    LA    1,PLISTMSG        PLIST IN R1
    CALL  CEEMSG            CALL CEEMSG
*****
*    NOW CALL CEE3DMP TO TAKE A CEEDUMP OF THIS FAILURE                *
*****
    LA    5,DUMPTITLE       PARM 1 IS DUMP TITLE
    ST    5,PLISTDMPP1     STORE ADDR IN PARMLIST
    LA    5,DUMPOPTS        PARM 2 IS DUMP OPTIONS
    ST    5,PLISTDMPP2     STORE ADDR IN PARMLIST
    LA    5,FEEDBACK        PARM 3 IS FEEDBACK CODE
    ST    5,PLISTDMPP3     STORE ADDR IN PARMLIST
    LA    1,PLISTDMP        PLIST IN R1
    CALL  CEE3DMP           CALL CEE3DMP
*****
*    SINCE WE HAD A FAILURE - LETS GET OUT NOW...                      *
*****
    LA    5,ABDCODE         PARM1 IS ABEND CODE (1234)
    ST    5,PLISTABD1      STORE ADDR IN PARMLIST
    LA    5,TIMING          PARM2 IS TIMING
    ST    5,PLISTABD2     STORE ADDR IN PARMLIST
    LA    1,PLISTABD       PLIST IN R1
    CALL  CEE3ABD
*****
*    SQUARE ROOT CALL WAS SUCCESSFUL - LETS OUTPUT THE RESULT.        *
*    FIRST CALL CONVE TO CONVERT OUTPUT TO A STRING.                  *
*    THEN CALL CEEMOUT WITH THE CONVERTED RESULT.                      *
*****
SQRTOK1 LA    5,RESULT       PARM 1 IS THE RESULT (FLOAT)
        ST    5,PLISTCONVEP1 STORE ADDR IN PARMLIST
        LA    5,MYTXT        PARM 2 IS THE CONVERTED STRING
        ST    5,PLISTCONVEP2 STORE ADDR IN PARMLIST
        LA    1,PLISTCONVE   PLIST IN R1
        LA    15,CONVE       LOAD ADDR OF ROUTINE
        BALR  14,15          CALL CONVE
        MVC   MOUTSTRTXT,MYTXT MOVE CONVERTED STRING TO OUTPUT STR
        LA    5,12           LENGTH OF STRING IS 12
        STH   5,MOUTSTRLEN   STORE THE LENGTH
        LA    5,2            MESSAGE DESTINATION IS 2
        ST    5,MOUTDEST     STORE THE DESTINATION
        LA    5,MOUTSTR      PARM 1 IS THE STRING
        ST    5,PLISTMOUTP1  STORE ADDR IN PARMLIST
        LA    5,MOUTDEST     PARM 2 IS THE MSG DEST
        ST    5,PLISTMOUTP2  STORE ADDR IN PARMLIST
        LA    5,0            USE NULL AS THE FEEDBACK CODE
        ST    5,PLISTMOUTP3  STORE IN PARMLIST
        LA    1,PLISTMOUT    PLIST IN R1
        CALL  CEEMOUT        CALL CEEMOUT TO OUTPUT RESULT
*****
*    SETUP FOR A CALL TO CEESSTQT (SQUARE ROOT) WITH 144.0            *
*****
    LA    5,FLOAT144        1ST PARM = 144.0
    ST    5,PLISTSQRTP1     STORE ADDR IN PARMLIST
    LA    5,FEEDBACK        2ND PARM = FEEDBACK CODE
    ST    5,PLISTSQRTP2     STORE ADDR IN PARMLIST
    LA    5,RESULT          3RD PARM = RESULT

```

```

      ST      5,PLISTSQRTP3      STORE ADDR IN PARMLIST
      LA      1,PLISTSQRT       PLIST IN R1
      CALL    CEESSTQT          CALL SQRT
*****
* CHECK THE RESULT FROM SQUARE ROOT CALL. *
*****
      LH      5,FBNUM           LOAD THE FEEDBACK CODE
      LA      6,0              SEE IF IT IS ZERO (OK)
      CR      5,6              MAKE THE CHECK
      BE      SQRTOK2          IF ZERO SKIP ERROR PATH
*****
* SQUARE ROOT CALL FAILED. *
*   SETUP TO CALL CEEMSG WITH FEEDBACK CODE FROM CEESSTQT *
*****
      LA      5,FEEDBACK        PARM 1 IS FEEDBACK CODE
      ST      5,PLISTMSGP1      STORE ADDR IN PARMLIST
      LA      5,MOUTDEST        PARM 2 IS THE MSG DEST
      ST      5,PLISTMSGP2      STORE ADDR IN PARMLIST
      LA      5,0              PARM 3 IS NEW FEEDBACK CODE
      ST      5,PLISTMSGP3      WE SEND NULL TO IGNORE FAILURES
      LA      1,PLISTMSG        PLIST IN R1
      CALL    CEEMSG           CALL CEEMSG
*****
*   NOW CALL CEE3DMP TO TAKE A CEEDUMP OF THIS FAILURE *
*****
      LA      5,DUMPTITLE       PARM 1 IS DUMP TITLE
      ST      5,PLISTDMPP1      STORE ADDR IN PARMLIST
      LA      5,DUMPOPTS        PARM 2 IS DUMP OPTIONS
      ST      5,PLISTDMPP2      STORE ADDR IN PARMLIST
      LA      5,FEEDBACK        PARM 3 IS FEEDBACK CODE
      ST      5,PLISTDMPP3      STORE ADDR IN PARMLIST
      LA      1,PLISTDMP        PLIST IN R1
      CALL    CEE3DMP          CALL CEE3DMP
*****
*   SINCE WE HAD A FAILURE - LETS GET OUT NOW... *
*****
      LA      5,ABDCODE         PARM1 IS ABEND CODE (1234)
      ST      5,PLISTABD1       STORE ADDR IN PARMLIST
      LA      5,TIMING          PARM2 IS TIMING
      ST      5,PLISTABD2       STORE ADDR IN PARMLIST
      LA      1,PLISTABD        PLIST IN R1
      CALL    CEE3ABD
*****
*   SQUARE ROOT CALL WAS SUCCESSFUL - LETS OUTPUT THE RESULT. *
*   FIRST CALL CONVE TO CONVERT OUTPUT TO A STRING. *
*   THEN CALL CEEMOUT WITH THE CONVERTED RESULT. *
*****
SQRTOK2 LA      5,RESULT         PARM 1 IS THE RESULT (FLOAT)
      ST      5,PLISTCONVEP1     STORE ADDR IN PARMLIST
      LA      5,MYTXT           PARM 2 IS THE CONVERTED STRING
      ST      5,PLISTCONVEP2     STORE ADDR IN PARMLIST
      LA      1,PLISTCONVE       PLIST IN R1
      LA      15,CONVE           LOAD ADDR OF ROUTINE
      BALR   14,15             CALL CONVE
      MVC    MOUTSTRXT,MYTXT     MOVE CONVERTED STRING TO OUTPUT STR
      LA      5,12             LENGTH OF STRING IS 12
      STH   5,MOUTSTRLEN        STORE THE LENGTH

```

```

LA      5,2                MESSAGE DESTINATION IS 2
ST      5,MOUTDEST        STORE THE DESTINATION
LA      5,MOUTSTR         PARM 1 IS THE STRING
ST      5,PLISTMOUTP1     STORE ADDR IN PARMLIST
LA      5,MOUTDEST        PARM 2 IS THE MSG DEST
ST      5,PLISTMOUTP2     STORE ADDR IN PARMLIST
LA      5,0                USE NULL AS THE FEEDBACK CODE
ST      5,PLISTMOUTP3     STORE IN PARMLIST
LA      1,PLISTMOUT       PLIST IN R1
CALL    CEEMOUT           CALL CEEMOUT TO OUTPUT RESULT
*****
* SETUP FOR A CALL TO CEESST (SQUARE ROOT) WITH 2500.0 *
*****
LA      5,FLOAT2500       1ST PARM = 2500.0
ST      5,PLISTSQRTP1     STORE ADDR IN PARMLIST
LA      5,FEEDBACK        2ND PARM = FEEDBACK CODE
ST      5,PLISTSQRTP2     STORE ADDR IN PARMLIST
LA      5,RESULT          3RD PARM = RESULT
ST      5,PLISTSQRTP3     STORE ADDR IN PARMLIST
LA      1,PLISTSQRT       PLIST IN R1
CALL    CEESST           CALL SQRT
*****
* CHECK THE RESULT FROM SQUARE ROOT CALL. *
*****
LA      5,FBNUM           LOAD THE FEEDBACK CODE
LA      6,0                SEE IF IT IS ZERO (OK)
CR      5,6                MAKE THE CHECK
BE      SQRTOK3           IF ZERO SKIP ERROR PATH
*****
* SQUARE ROOT CALL FAILED. *
* SETUP TO CALL CEEMSG WITH FEEDBACK CODE FROM CEESST *
*****
LA      5,FEEDBACK        PARM 1 IS FEEDBACK CODE
ST      5,PLISTMSGP1      STORE ADDR IN PARMLIST
LA      5,MOUTDEST        PARM 2 IS THE MSG DEST
ST      5,PLISTMSGP2      STORE ADDR IN PARMLIST
LA      5,0                PARM 3 IS NEW FEEDBACK CODE
ST      5,PLISTMSGP3      WE SEND NULL TO IGNORE FAILURES
LA      1,PLISTMSG        PLIST IN R1
CALL    CEEMSG           CALL CEEMSG
*****
* NOW CALL CEE3DMP TO TAKE A CEEDUMP OF THIS FAILURE *
*****
LA      5,DUMPTITLE       PARM 1 IS DUMP TITLE
ST      5,PLISTDMP1       STORE ADDR IN PARMLIST
LA      5,DUMPOPTS        PARM 2 IS DUMP OPTIONS
ST      5,PLISTDMP2       STORE ADDR IN PARMLIST
LA      5,FEEDBACK        PARM 3 IS FEEDBACK CODE
ST      5,PLISTDMP3       STORE ADDR IN PARMLIST
LA      1,PLISTDMP        PLIST IN R1
CALL    CEE3DMP          CALL CEE3DMP
*****
* SINCE WE HAD A FAILURE - LETS GET OUT NOW... *
*****
LA      5,ABDCODE         PARM1 IS ABEND CODE (1234)
ST      5,PLISTABD1       STORE ADDR IN PARMLIST
LA      5,TIMING          PARM2 IS TIMING

```

```

        ST      5,PLISTABD2      STORE ADDR IN PARMLIST
        LA      1,PLISTABD      PLIST IN R1
        CALL   CEE3ABD
*****
*   SQUARE ROOT CALL WAS SUCCESSFUL - LETS OUTPUT THE RESULT.           *
*   FIRST CALL CONVE TO CONVERT OUTPUT TO A STRING.                     *
*   THEN CALL CEEMOUT WITH THE CONVERTED RESULT.                         *
*****
SQRTOK3  LA      5,RESULT        PARM 1 IS THE RESULT (FLOAT)
        ST      5,PLISTCONVEP1   STORE ADDR IN PARMLIST
        LA      5,MYTXT         PARM 2 IS THE CONVERTED STRING
        ST      5,PLISTCONVEP2   STORE ADDR IN PARMLIST
        LA      1,PLISTCONVE     PLIST IN R1
        LA      15,CONVE        LOAD ADDR OF ROUTINE
        BALR   14,15           CALL CONVE
        MVC    MOUTSTRTXT,MYTXT  MOVE CONVERTED STRING TO OUTPUT STR
        LA      5,12           LENGTH OF STRING IS 12
        STH    5,MOUTSTRLEN     STORE THE LENGTH
        LA      5,2            MESSAGE DESTINATION IS 2
        ST      5,MOUTDEST      STORE THE DESTINATION
        LA      5,MOUTSTR      PARM 1 IS THE STRING
        ST      5,PLISTMOUTP1   STORE ADDR IN PARMLIST
        LA      5,MOUTDEST      PARM 2 IS THE MSG DEST
        ST      5,PLISTMOUTP2   STORE ADDR IN PARMLIST
        LA      5,0            USE NULL AS THE FEEDBACK CODE
        ST      5,PLISTMOUTP3   STORE IN PARMLIST
        LA      1,PLISTMOUT     PLIST IN R1
        CALL   CEEMOUT         CALL CEEMOUT TO OUTPUT RESULT
*****
*   SETUP FOR A CALL TO CEESSTQT (SQUARE ROOT) WITH -99.0 (ERROR)      *
*****
        LA      5,FLOATM99      1ST PARM = -99.0
        ST      5,PLISTSQRTP1   STORE ADDR IN PARMLIST
        LA      5,FEEDBACK      2ND PARM = FEEDBACK CODE
        ST      5,PLISTSQRTP2   STORE ADDR IN PARMLIST
        LA      5,RESULT        3RD PARM = RESULT
        ST      5,PLISTSQRTP3   STORE ADDR IN PARMLIST
        LA      1,PLISTSQRT     PLIST IN R1
        CALL   CEESSTQT        CALL SQRT
*****
*   CHECK THE RESULT FROM SQUARE ROOT CALL.                             *
*****
        LH      5,FBNUM        LOAD THE FEEDBACK CODE
        LA      6,0            SEE IF IT IS ZERO (OK)
        CR      5,6            MAKE THE CHECK
        BE     SQRTOK4        IF ZERO SKIP ERROR PATH
*****
*   SQUARE ROOT CALL FAILED.                                           *
*   SETUP TO CALL CEEMSG WITH FEEDBACK CODE FROM CEESSTQT            *
*****
        LA      5,FEEDBACK      PARM 1 IS FEEDBACK CODE
        ST      5,PLISTMSGP1    STORE ADDR IN PARMLIST
        LA      5,MOUTDEST      PARM 2 IS THE MSG DEST
        ST      5,PLISTMSGP2    STORE ADDR IN PARMLIST
        LA      5,0            PARM 3 IS NEW FEEDBACK CODE
        ST      5,PLISTMSGP3    WE SEND NULL TO IGNORE FAILURES
        LA      1,PLISTMSG     PLIST IN R1

```

```

          CALL CEEMSG          CALL CEEMSG
*****
*   NOW CALL CEE3DMP TO TAKE A CEEDUMP OF THIS FAILURE   *
*****
          LA    5,DUMPTITLE      PARM 1 IS DUMP TITLE
          ST    5,PLISTDMPP1     STORE ADDR IN PARMLIST
          LA    5,DUMPOPTS      PARM 2 IS DUMP OPTIONS
          ST    5,PLISTDMPP2     STORE ADDR IN PARMLIST
          LA    5,FEEDBACK      PARM 3 IS FEEDBACK CODE
          ST    5,PLISTDMPP3     STORE ADDR IN PARMLIST
          LA    1,PLISTDMP      PLIST IN R1
          CALL  CEE3DMP          CALL CEE3DMP
*****
*   SINCE WE HAD A FAILURE - LETS GET OUT NOW...   *
*****
          LA    5,ABDCODE      PARM1 IS ABEND CODE (1234)
          ST    5,PLISTABD1     STORE ADDR IN PARMLIST
          LA    5,TIMING       PARM2 IS TIMING
          ST    5,PLISTABD2     STORE ADDR IN PARMLIST
          LA    1,PLISTABD     PLIST IN R1
          CALL  CEE3ABD
*****
*   SQUARE ROOT CALL WAS SUCCESSFUL - LETS OUTPUT THE RESULT.   *
*   FIRST CALL CONVE TO CONVERT OUTPUT TO A STRING.   *
*   THEN CALL CEEMOUT WITH THE CONVERTED RESULT.   *
*****
SQRTOK4  LA    5,RESULT        PARM 1 IS THE RESULT (FLOAT)
          ST    5,PLISTCONVEP1  STORE ADDR IN PARMLIST
          LA    5,MYTXT        PARM 2 IS THE CONVERTED STRING
          ST    5,PLISTCONVEP2  STORE ADDR IN PARMLIST
          LA    1,PLISTCONVE    PLIST IN R1
          LA    15,CONVE        LOAD ADDR OF ROUTINE
          BALR  14,15          CALL CONVE
          MVC   MOUTSTRTXT,MYTXT MOVE CONVERTED STRING TO OUTPUT STR
          LA    5,12           LENGTH OF STRING IS 12
          STH  5,MOUTSTRLEN     STORE THE LENGTH
          LA    5,2           MESSAGE DESTINATION IS 2
          ST    5,MOUTDEST     STORE THE DESTINATION
          LA    5,MOUTSTR      PARM 1 IS THE STRING
          ST    5,PLISTMOUTP1  STORE ADDR IN PARMLIST
          LA    5,MOUTDEST     PARM 2 IS THE MSG DEST
          ST    5,PLISTMOUTP2  STORE ADDR IN PARMLIST
          LA    5,0           USE NULL AS THE FEEDBACK CODE
          ST    5,PLISTMOUTP3  STORE IN PARMLIST
          LA    1,PLISTMOUT    PLIST IN R1
          CALL  CEEMOUT        CALL CEEMOUT TO OUTPUT RESULT
*****
*   COMMON EXIT POINT   *
*****
ALLDONE  DS    0H            COMMON EXIT POINT
          CEETERM RC=0       USE CEETERM TO EXIT TO CALLER
*****
*   EQUATES   *
*****
R0      EQU  0
R1      EQU  1
R2      EQU  2
00018001

```

```

R13      EQU    13
R15      EQU    15
F0       EQU    0
ZONEF    EQU    X'0F'
*****
*        CONVE ROUTINE - THANKS TO JOHN EHRMAN        *
*****
CONVE    SAVE    (14,2),, *          SAVE ALL RELEVANT REGISTERS
         ST      R13,SAVE+4         SAVE R13 FOR TRACEBACKS
         LA      R0,SAVE            ADDRESS OF SAVE AREA
         ST      R13,SAVE+4         CHAIN TO CALLER'S AREA
         ST      R0,8(,R13)         MAKE A BELIEVER OUT OF THE CALLER
         LR      R13,R0             SAVE R13 IN CASE OF TRACEBACKS
         LM      R1,R2,0(R1)        GET ARGUMENT ADDRESSES
         USING   ESTRING,R2         MAPPING FOR OUTPUT STRING
         MVC     DWORD(4),0(R1)     ALIGN FLOATING ARGUMENT
         SR      R1,R1              CLEAR FOR DECIMAL EXPONENT
         LD      F0,UNNORMER        CLEAR RIGHT HALF F0 FOR ARGUMENT
         MVI     ESIGN,C'+ '        ASSUME POSITIVE RESULT
         TM      DWORD,X'80'        CHECK SIGN OF FRACTION
         BZ      *+8                SKIP SETTING OF - SIGN IF > 0
         MVI     ESIGN,C'- '        RESULT IS NEGATIVE.
         LE      F0,DWORD           LOAD ARGUMENT
         MVI     DECIMAL,C'.'       SET INITIAL DECIMAL POINT
         LTER    F0,F0              CHECK INPUT VALUE SIGN, ETC.
         BZ      ZERO              IT'S ZERO, BY GUM. NO WORK ATALL.
         LPER    F0,F0              FORCE ARGUMENT SIGN + IN REGISTER
         CD      F0,=D'1'          SEE IF ARGUMENT HAS +/- EXPONENT
         BE      FRACTONE          ALL DONE IF EXACTLY 1.0
         BH      POSEXP            BRANCH IF POSITIVE EXPONENT
NEGEXP   DS      0H                EXPONENT IS NEGATIVE
         MD      F0,=D'1.E10'      GO UP BY TENS FIRST
         SH      R1,=H'10'         AND COUNT DECIMAL EXPONENT DOWN
         CD      F0,=D'1'          WHERE DID THAT TAKE US?
         BL      NEGEXP            IF IT'S STILL TOO SMALL, GO AGAIN
         BE      FRACTONE          IF EXACTLY EQUAL, THAT'S NICE.
POSEXP   DS      0H                EXPONENT NOW POSITIVE
         CD      F0,=D'1.E10'      CHECK FOR REALLY BIG FELLOW
         BL      REDUCE            IF IN RANGE, REDUCE SLOWLY
         MD      F0,=D'1.E-10'     MULTIPLY GIVES MORE ACCURACY.
         AH      R1,=H'10'         BUMP DECIMAL EXPONENT ACCORDINGLY
         B       POSEXP            AND DO ANOTHER REDUCTION CYCLE
REDUCE   DS      0H                EXPONENT NOW BETWEEN +10 AND 0
         CD      F0,=D'1'          COMPARE TO 10**0
         BE      FRACTONE          FRACTION NOW IS 1
         BL      CONVERT           READY TO CONVERT IF A FRACTION
         MD      F0,=D'.1'         REDUCE BY 10**1
         AH      R1,=H'1'          INCREMENT EXPONENT ACCORDINGLY
         B       REDUCE            AND GO AROUND AGAIN
CONVERT  DS      0H                READY TO DO THE JOB
         MD      F0,=D'1.E6'       CONVERT TO INTEGER IN (1,10**6)
         AD      F0,=D'.5'         ROUND IT PROPERLY
         CD      F0,=D'1.E6'       DID IT ROUND UP TO 1000000?
         BNL     FRACTONE          IT DID, USE FRACTION .100000
CONVERTX DS      0H
         AW      F0,UNNORMER        ADD UNNORMALIZER
         STD     F0,DWORD           STORE FOR THE NONCE

```

```

L      R0,DWORD+4      GET INTEGER PART
CVD    R0,DWORD        CONVERT TO PACKED DECIMAL
OI     DWORD+7,ZONEF   SET CORRECT ZONE
UNPK   EDIGITS,DWORD   PLACE DIGITS INTO STRING
DOEXPON DS  0H         CONVERT EXPONENT
MVI    EXPONE,C'E'     SET THE 'E'
MVI    EXPSIGN,C'+'    ASSUME POSITIVE EXPONENT
LTR    R1,R1          CHECK FOR CORRECT ASSUMPTION
BNM    *+8            SKIP IF IT WAS RIGHT
MVI    EXPSIGN,C'-'    SET EXPONENT SIGN -
CVD    R1,DWORD        CONVERT TO DECIMAL
OI     DWORD+7,ZONEF   SET CORRECT (POSITIVE) ZONE
UNPK   EXPONENT,DWORD  UNPACK TO ZONED DECIMAL
EXIT   DS  0H
L      R13,SAVE+4      RESTORE CALLER'S R13
RETURN (0,2),T        RETURN TO CALLER
ZERO   MVC  EDIGITS,=6C'0' SET FRACTION TO ZEROS
B      DOEXPON        AND GO DO EXPONENT
FRACTONE MVC EDIGITS,=C'100000' SET FRACTION DIGITS TO .1
AH     R1,=H'1'       COMPENSATE BY UPPING EXPONENT
B      DOEXPON        AND GO DO EXPONENT
SPACE 2
MAINPPA CEEPPA ,      CONSTANTS DESCRIBING THE CODE BLOCK      00019300
*****
*   CONSTANTS   *
*****
CEEBALCT
MYTXT   DC   C'THIS IS A TEST'
FLOAT9   DC   E'9.00'
FLOAT144 DC   E'144.00'
FLOAT2500 DC  E'2500.00'
FLOATM99 DC   E'-99.00'
DUMPTITLE DC CL80'SAMPLE DUMP TAKEN BY CEE3DMP'
DUMPOPTS DC  CL255'NOCOND'
ABDCODE  DC   F'1234'
TIMING   DC   F'0'
*
ZTABLE  DC   C'0123456789ABCDEF' HEX-TO-EBCDIC TRANSLATE TABLE
PATTERN DC   C' ',X'20202020202020202020'
SAVE     DC   3F'0'      A PHONY SAVE AREA FOR TRACING
DWORD    DC   D'0'
UNNORMER DC  X'4E00000000000000' FOR INTEGER CONVERSION
*****
*   CONVE WORK AREA   *
*****
ESTRING DSECT
ESIGN   DS   C           SIGN OF FRACTION
DECIMAL DS   C           DECIMAL POINT
EDIGITS DS  CL6         FRACTION DIGITS
EXPONE  DS   C           EXPONENT INDICATOR 'E'
EXPSIGN DS   C           EXPONENT SIGN
EXPONENT DS CL2         DECIMAL EXPONENT
*****
*   WORKAREA AND DSA   *
*****
WORKAREA DSECT
ORG     *+CEEDSASZ      LEAVE SPACE FOR THE DSA FIXED PART      00019700
                                00019800

```

```

*
                                00020000
                                00020400
      DS      0D
MOUTSTR      DS  0F      1/2 WORD PREFIX STRING FOR CEEMOUT
MOUTSTRLEN   DS  H      LENGTH
MOUTSTRTXT   DS  XL255   STRING
MOUTDEST     DS  F      MESSAGE DESTINATION (2)
*
FEEDBACK     DS  0F      FEEDBACK CODE (12 BYTES)
FBSEV        DS  H      FEEDBACK SEVERITY
FBNUM        DS  H      FEEDBACK MESSAGE NUMBER
FBFLAG       DS  X      FEEDBACK FLAGS
FBPREFIX     DS  CL3     FEEDBACK MESSAGE PREFIX
FBISI        DS  F      FEEDBACK INSTANCE SPECIFIC INFO
*
PLISTMOUT    DS  0F      PLIST FOR CEEMOUT
PLISTMOUTP1  DS  A
PLISTMOUTP2  DS  A
PLISTMOUTP3  DS  A
*
RESULT       DS  E      RESULT FOR MATH ROUTINE
PLISTSQRT    DS  0F      PLIST FOR SQRT CALL
PLISTSQRTP1  DS  A
PLISTSQRTP2  DS  A
PLISTSQRTP3  DS  A
*
PLISTMSG     DS  0F      PLIST FOR CEEMSG
PLISTMSGP1   DS  A
PLISTMSGP2   DS  A
PLISTMSGP3   DS  A
*
PLISTDMP     DS  0F      PLIST FOR CEE3DMP
PLISTDMPP1   DS  A
PLISTDMPP2   DS  A
PLISTDMPP3   DS  A
*
PLISTABD     DS  0F
PLISTABD1    DS  A
PLISTABD2    DS  A
*
PLISTCONVE   DS  0F      PLIST FOR CONVE ROUTINE
PLISTCONVEP1 DS  A
PLISTCONVEP2 DS  A
WORKSIZE EQU *-WORKAREA
      CEEDSA ,      MAPPING OF THE DYNAMIC SAVE AREA      00020500
      CEECAA ,      MAPPING OF THE COMMON ANCHOR AREA      00020600
*
*
                                00020700
                                00050000
                                00060000
                                00070001
      END      CSSQRTA
→

```

Assembler Example - 2 -

Assembler Page 1 of 10