SHARE PROGRAM LIBRARY AGENCY

PROGRAM NUMBER

114002

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA

(305) - 284-6257

SHARE Program Library Agency
Triangle Universities Computation Center
P. O. Box 12076
Research Triangle Park, N. C. 27709

CONTRIBUTED PROGRAM LIBRARY SUBMI
(for IBM S/360, 1130 and 1800)

This form should be completed and submitted with the program package to PID at the address shown above. Standards and instructions for submitting programs are in your *User Group Reference Manual* or the *Contributed Program Submittal Standards Manual* available from PID.

(1) **Program Order Number** *(to be filled in by PID)* 360D 11.4.002

(2) **System Type** *(machine)* S/360

(3) **Search Key** . . . . . /PL-I/ /DESK-CALCULATOR
/FOR USE ON /2741/ /CONSOLE/

(4) **Name of Author** *(if different than submitter's)* . . . . .

(5) **Submitter's Name** *(direct technical inquiries to)*

(6) **Submitter's Address** . . . . . Mr. Richard Osgood
Yale Computer Center
175 Whitney Avenue
New Haven, CT 06520

(7) **Title of Program** . . . . DCALC

(8) **Submitter's User Group Affiliation Code and Installation Code** . . . . . S YU

(9) **Submitter's Own Program Identification and Suffix** *(optional)* . . . . . DCAL

(10) **Primary Subject Code** 11.4

(11) **Secondary Subject Codes** . . . . . 03.3 30.2 . .

(12) **Operating or Monitor System Required** OS/360

(13) **New or Revision Code** *(if revision, show prior Program Order Number in item 1)* . . . . . N

(14) **Year Completed** 68

(15) **Date of Submittal** 050668

(16) **Documentation** *(number of original pages submitted)* 22

(17) **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.*

CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide

a. Purpose
b. Programming Language used
c. Version and modification level or release number of IBM Programming System used, or program order number for non-IBM authored program used
d. Field of application
e. Type of routine (main program, subroutine, etc.)
f. Specific description of machine requirements
g. Engineering Changes (EC) level of equipment (if pertinent)

ABSTRACT

The purpose of DCALC is to provide an interactive desk-calculator facility under OS in an environment supporting 2741's and other interactive devices. The character strings in lines 80-138 of the listing describe its use, and show how it is much more than a simple desk-calculator.

DCALC uses SYSIN and SYSPRINT for all I/O. It is written entirely in PL/I, and is a self-contained main program.

(Please attach additional pages if necessary) . . . . . **Total pages attached** 0

Permission to Publish

"I hereby give anyone permission to reprint, reproduce, and distribute this program to anyone else."

(18) **Signature of Submitter and Date** Richard A. Cosier 5/6/68

(19) **Signature of Installation Addressee** Lois C. Frampton

T4SF

Table of Contents

Deck Key - PL/I source deck sequenced 0001 through 0524 in cc 77-80; 524 cards.

Purpose - to provide an interactive tool for calculations.

Functional and Mathematical Methods - The parse is based on the Ershov algorithm as described in Graham, "Bounded Context Analysis", SJCC, 1964, AFIPS, pp. 17-30. The recognizer and interpreter are incorporated directly into the parse. The function definition and expansion facility are supported by character string replacement and based on the fact that assignment operations also yield values.

Limitations - Arithmetic is carried out using PL/I floating point double precision arithmetic.

Equipment Specifications - most appropriately used with some variety of interactive device such as 2741, 2260, etc.

Storage Requirements - approximately 50K bytes of main store.

Operating Instructions - respond "?" to first prompting message to display instructions.

Input/Output - line-by-line character string interactive dialogue.

"LABEL TABLE"

ATTRIBUTE AND CROSS-REFERENCE TABLE

IDENTIFIER		ATTRIBUTES AND REFERENCES
ABS		GENERIC, BUILT-IN FUNCTION 421
ATAN		GENERIC, BUILT-IN FUNCTION 419
BLL	STRING CONTAINING 11 BLANKS	STATIC, INITIAL, STRING, CHARACTER 528, 533, 537
BLEN		STATEMENT LABEL CONSTANT 196
COM1	COMMENT FOR ERROR	STATIC, INITIAL, STRING, CHARACTER 437
COM2	" " "	STATIC, INITIAL, STRING, CHARACTER 437
CONST	SWITCH TO TEST TO CONTROL PRINTING	STATIC, STRING, BIT 255, 281, 303, 378, 428
COS		GENERIC, BUILT-IN FUNCTION 413
CUF	IS CURRENT SYMBOL AN OPERAND	STATIC, STRING, BIT 254, 258, 262, 283, 292, [illegible]
CVT	PERFORMS CONVERSIONS FOR DISPLAY	ENTRY, DECIMAL, FLOAT(SINGLE) 369, 425
D1	TEMPORARY FLOATING PT. VAR	STATIC, DECIMAL, FLOAT(DOUBLE) 302, 303, 347, 368, 394, 397, 337, 399, [illegible], 401, 401, 403, 403, 405, 405, 405, 407 407, 409, 411, 413, 415, 417, 419, 421, 426, 427, [illegible], 432, 432, 434, 44[illegible]
D2	" " " "	STATIC, DECIMAL, FLOAT(DOUBLE) 395, 397, 399, 401, 403, 404, 407, 409, 411, 413, 415, 417, 419, 421, 424, 43[illegible]
DCALC	MAIN ROUTINE	ENTRY, DECIMAL, FLOAT(SINGLE) 15
DLIST	STRING CONTAINING DIGITS 0-9 E and .	STATIC, INITIAL, STRING, CHARACTER 288, 2[illegible]1, 2[illegible]
DYNAL	ANALYZE DYNAMIC ASSIGNMENTS LOOP	STATEMENT LABEL CONSTANT 242, [illegible]
DYNAM	" " ENTRY "	STATEMENT LABEL CONSTANT
E1	CH. STRING TEMP - CONTAINS CURRENT ELEMENT	STATIC, STRING, CHARACTER 263, 26[illegible], 271, [illegible], [illegible], [illegible], [illegible], 315, [illegible], 333, [illegible], [illegible], 370, [illegible], 377, [illegible], [illegible] [illegible], [illegible], [illegible]

OCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
516	ELIST ERASE LIST OF VARIABLES	STATEMENT LABEL CONSTANT 504
458	ELOOP END OF MAIN LOOP	STATEMENT LABEL CONSTANT 381,455
141	EQSS INDEX OF '=' IN SS	STATIC,BINARY,FIXED(15,0) 151,152,167,179,179,192,212
514	ERASE ERASE ENTIRE SCRN	STATEMENT LABEL CONSTANT 503
479	ERROR2 ERASE MESSAGE	STATEMENT LABEL CONSTANT 301,312
491	ERROR3 " "	STATEMENT LABEL CONSTANT 344
489	ERROR4 " "	STATEMENT LABEL CONSTANT 160
487	ERROR5 " "	STATEMENT LABEL CONSTANT 157,225
485	ERROR6 " "	STATEMENT LABEL CONSTANT 170,235,245
483	ERROR7 " "	STATEMENT LABEL CONSTANT 238
481	ERRORN " "	STATEMENT LABEL CONSTANT 164,212,325
	EXP	GENERIC,BUILT-IN FUNCTION 417
2	EXPR VECTOR OF ESTABLISHED EXPRESSIONS	IN TABLE(*),AUTOMATIC,PACKED,STRING,CHARACTER,VARYING 171,179,187,199,234,236,239,355
371	GETBACK	STATEMENT LABEL CONSTANT 334,341,360
364	GTM	STATEMENT LABEL CONSTANT
362	GTV	STATEMENT LABEL CONSTANT 350
79	HOWTO Explains Use of Fundamental Factors	ENTRY,DECIMAL,FLOAT(SINGLE) 45
5	I Utility counter and index (points to current character)	STATIC,BINARY,FIXED(15,0) 31,32,35,258,261,266,268,269,269,272,272,274,274,276,277,277,279,288 288,291,[illegible],[illegible],[illegible],304,309,311,315,315,319,323,423,372,501,502,516

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		516,517,517,527,529,529,529,532,532,532,534,536,537,541,542
548	I " " " "	STATIC,BINARY,FIXED(15,0) 549,550,552,559
571	I " " " "	STATIC,BINARY,FIXED(15,0) 576,576,576,578
570	ID Determines Validity of identifier	ENTRY,STRING,BIT 195,211,324
	INDEX	GENERIC,BUILT-IN FUNCTION 149,151,152,[illegible],159,223,224,266,283,291,295,309,362,432,501,549,574 576
5	J Utility Counter & Index (points to the end of string)	STATIC,BINARY,FIXED(15,0) 30,34,34,35,38,43,41,144,146,147,147,170,194,201,202,202,206,206,215 215,216,224,236,236,237,247,247,253,258,291,293,295,309,501,517
548	J " " " "	STATIC,BINARY,FIXED(15,0) 552,553,555,561,564,564,564
5	K Utility Counter and pointer to current stack element in M	STATIC,BINARY,FIXED(15,0) 256,377,378,379,380,384,384,385,391,394,395,437,440,442,442,442,452 453,454,454,534,536,541
5	L Counter & Index	STATIC,BINARY,FIXED(15,0) 291,291,291,293,293,293,293,295,295,295,295,298,298,299,299,302,302 304,309,309,309,311,313,313,313,315,316,317,317,318,323,331,356,372
7	LAST Contains last Input String	STATIC,STRING,CHARACTER,VARYING 20,27,28
	LENGTH	GENERIC,BUILT-IN FUNCTION 199,234,236
140	LIST List of functions expanded so far	(*),STATIC,ALIGNED,STRING,CHARACTER 228,241
	LOG	GENERIC,BUILT-IN FUNCTION 415
24	LOOP	STATEMENT LABEL CONSTANT 19,39,46,51,61,74,77
99	LOOP	STATEMENT LABEL CONSTANT
493	LOOP	STATEMENT LABEL CONSTANT 177,199,366
513	LOOP	STATEMENT LABEL CONSTANT 515,521,524,545
391	LOOPA	STATEMENT LABEL CONSTANT

DCL NO.	IDENTIFIER		ATTRIBUTES AND REFERENCES
			264,444
492	LOOPE	Error exit from Process	STATEMENT LABEL CONSTANT 438,480,492,484,486,488,490
11	M	Table of previously processed elements (Push-down Stack)	(*),STATIC,ALIGNED,STRING,CHARACTER 257,380,385,391,394,395,437,440,442,448,453,453
100	MOREHW2	Additional Instructions on use of table	ENTRY,BINARY,FIXED(15,0) 50
7	MSG	String for Output Message Buffering	STATIC,STRING,CHARACTER,VARYING 21,437,479,481,483,485,487,489,491,492,529,529,530,531,535,537,537 539,540,542,542,544
5	N	Counter and Index	STATIC,BINARY,FIXED(15,0) 149,150,152,155,156,158,162,165,165,165,169,171,171,174,179,180,182 183,184,187,187,197,197,197,199,199,220,221,223,224,226,239,239,239 246,246,246,328,329,332,333,348,349,353,370,518,519,521
13	N1	" " "	STATIC,BINARY,FIXED(15,0) 223,224,224,226,239,239,246,246
13	N2	" " "	STATIC,BINARY,FIXED(15,0) 158,159,227,228,231,232,234,236,239
2	NAME	Identifier List	IN TABLE(*),AUTOMATIC,PACKED,STRING,CHARACTER 165,182,187,197,232,329,349,354,432,432,519,528,528,528,537
5	NC	Number of Identifiers in table	STATIC,BINARY,FIXED(15,0) 22,165,169,174,174,180,184,197,199,231,329,343,348,353,353,354,355 358,359,514,518,521,522,522,527,536,541
141	NL	Number of elements in List	STATIC,BINARY,FIXED(15,0) 217,227,237,240,240,241
249	PARSE	Begin Parse Phase	STATEMENT LABEL CONSTANT 222
3	PLIST	List of Punctuation Characters	STATIC,INITIAL,STRING,CHARACTER 159,266,309,576
6	PRCSN	Current Precision	STATIC,INITIAL,BINARY,FIXED(15,0) 508,509,510,553,557,561,564
459	PREC	Returns Precedence of Operator	ENTRY,BINARY,FIXED(15,0) 392,392
3	PREV	IS PREVIOUS SYMBOL AN OPERAND	STATIC,STRING,BIT 259,382
527	PRINT	PRINT TABLE	STATEMENT LABEL CONSTANT 498

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
139	PROCESS PARSE AND EXECUTE	ENTRY,DECIMAL,FLOAT(SINGLE) 76
11	R TEMPORARY CHAR.STRING	STATIC,STRING,CHARACTER 368,370,424,427,431,432,444,543,552,557,557,559,561,561,561,561,564 564,564,564
1	RCALC	ENTRY,DECIMAL,FLOAT(SINGLE)
246	RMV REMOVE ITS	STATEMENT LABEL CONSTANT 229
4	S STRING WORK SPACE DEFINED ON SS	(*),AUTOMATIC,DEFINED,PACKED,STRING,CHARACTER 32,35,35,43,43,53,64,66,144,145,153,159,167,216,261,266,268,269,269 274,274,274,279,288,288,291,293,293,293,295,298,298,309,313,313,313 316,331,356,497,499,500
258	S3	STATEMENT LABEL CONSTANT
290	S4	STATEMENT LABEL CONSTANT 290
382	SCAN COMPARE PRECEDENCE	STATEMENT LABEL CONSTANT 286,319
7	SI STRING INPUT AREA	STATIC,STRING,CHARACTER,VARYING 25,26,27,28,29,364,365,367
	SIN	GENERIC,BUILT-IN FUNCTION 411
	SQRT	GENERIC,BUILT-IN FUNCTION 409
4	SS STRING WORK SPACE	STATIC,STRING,CHARACTER 29,40,40,149,151,152,155,162,179,194,201,201,205,205,210,210,220,223 226,239,239,239,239,246,246,246,246,251,312,315,323,501,504,517
375	STACK PUT ELEMENT ON LIST M	STATEMENT LABEL CONSTANT 285,305,373,386,457
	SUBSTR	GENERIC,BUILT-IN FUNCTION 40,152,155,162,179,194,201,210,223,226,239,239,239,246,246,246,312 315,323,347,362,432,501,504,517,552,557,559,561,561,561,561,564,564 564
495	SUPPORT UTILITY SUPPORT ROUTINES	ENTRY,DECIMAL,FLOAT(SINGLE) 73
	SYSIN	FILE,EXTERNAL 25,364
	SYSPRINT	FILE,EXTERNAL 23,24,51,69,79,83,91,92,93,94,95,96,97,98,99,101,102,103,104,105,106,

DCL NO.	IDENTIFIER	ATTRIBUTES AND REFERENCES
		98,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116 117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133 134,135,136,171,176,187,219,251,252,327,362,363,376,389,431,432,442 492,512,529,531,532,539,544
9	T	TEMPORARY CHARACTER STRING AUTOMATIC,DEFINED,STRING,CHARACTER 517,519
8	TAB	STATIC,STRING,CHARACTER 162,165,182,194,197,210,226,228,232,241,323,327,329,339,347,347,349 354,362,362,481,485,491
2	TABLE	NAME/VARIABLE TABLE (*),AUTOMATIC,PACKED,STRUCTURE 521,521
9	TABX	DEFINED ON T (*),AUTOMATIC,DEFINED,PACKED,STRING,CHARACTER 345,572,572,574,576,576,579
10	TST	SWITCH FOR DEBUGGING MODE STATIC,INITIAL,STRING,BIT 55,56,58,186,218,249,326,375,388,441
2	VALUE	VALUES OF VARIABLES IN TABLE(*),AUTOMATIC,PACKED,STRING,CHARACTER 171,183,333,358,370,427,530,532,530,542
460	X	PARAMETER,STRING,CHARACTER 459,461,461,463,463,465,467,469,471,473,475
390	X1	SCRATCH STRING AUTOMATIC,STRING,CHARACTER 391,392,396,398,400,402,404,406,408,410,412,414,416,418,420,422
3	XLIST	STRING CONTAINING DIGITS STATIC,INITIAL,STRING,CHARACTER 574

Testing - The program has been available and in use as part of one interactive time-sharing system (CYTOS at Yale University) since May 6, 1968, with no errors having been detected. It has been distributed to over one dozen installations in an earlier version starting in January, 1968, with no report of malfunction.

"LISTINGS


```
 1    RCALC : PROC OPTIONS (MAIN);
        /*

      AUTHOR:  R. F. ROSIN
               COMPUTER CENTER
               YALE UNIVERSITY
               175 WHITNEY AVE.
               NEW HAVEN, CONN. 06520
               PHONE: (203) 787-3131, EXT. 8411

        */

 2    DCL 1 TABLE(30),
            2 NAME CHAR(10), 2 VALUE CHAR(22), 2 EXPR CHAR(80) VAR;

 3      DCL ( DLIST CHAR (12) INITIAL ('012345678'.E'),
            XLIST CHAR(10) INITIAL ('0123456789'),
            PLIST CHAR (9) INITIAL ('+-*/()=.|'),
                     (PREV, CUR) BIT (1)  ) STATIC ;
 4    DCL SS CHAR(400) STATIC         , S(400) CHAR(1) DEFINED SS PACKED;
 5      DCL (NC,N,L,K,J,I, (D1,D2) FLOAT DEC (14) ) STATIC ;
 6          DCL PRCSN FIXED BIN INITIAL(7) STATIC ;
 7         DCL (SI,MSG,LAST) CHAR(80) VAR STATIC;
 8     DCL TAB CHAR(22) STATIC;
 9    DCL T  CHAR (22) DEFINED FL, TABX(22) CHAR(1) DEFINED TAB PACKED;
10          DCL TST BIT(1) STATIC INITIAL ('0'B);
11    DCL (EL,R,M(30)) CHAR (22) STATIC;
12     DCL ID ENTRY RETURNS(BIT(1));
13       DCL (N1,N2) STATIC;
14      DCL CONST BIT(1) STATIC;

15          CALL DCALC;
16            RETURN;

      /* ============================================================ */

17    DCALC: PROC;

18      ON ERROR GO TO LOOP;
20     LAST = '';
21          MSG = ''; NC = 0;
23     PUT LIST ('IF YOU WANT HELP, TYPE ''?'' OTHERWISE BEGIN  ') SKIP;

24    LOOP:
          PUT SKIP;
25          GET EDIT (SI) (A(80));
26     /* */   IF SI = ' ' THEN SI = LAST; ELSE LAST = SI;
29      SS = SI;  J = 0;
31     DO I = 1 TO 80;
32          IF S(I) ¬= ' ' THEN DO; J=J+1; S(J)=S(I); END;  END;
38      /*  */    IF J = 0 THEN GO TO LOOP;
40             SS = SUBSTR(SS,1,J);
41     IF J = 1 THEN DO;
43       IF S(1) = '?' THEN DO; CALL HOWTO; GO TO LOOP;     END;
```

```
43            IF S(1) = '*' THEN DO; CALL MORTHW2; GO TO LOOP; END;
53            IF S(1) = '*' THEN DO;
55                IF TST THEN TST = '0'B;
57                ELSE DO;
58                    TST = '1'B;
59                    PUT LIST ('05/03-21:00 YOU HAVE ENTERED DEBUG MODE'
                        || ', TO GET OUT TYPE ''*''') SKIP;
60                END; GO TO LOOP;
62            END;
63        END;
64            IF S(1) = '/' THEN DO;
66                IF S(2) = 'T' THEN DO;
68                    PUT LIST ('      IT HAS BEEN A PLEASURE TO SERVE YOU  '
                        || ' . . . DCALC') SKIP;
69        /*   */ PUT LIST (' ') SKIP;
70                    REVERT ERROR;
71                        RETURN;
72                END;
73                CALL SUPPORT; GO TO LOOP;
75            END;
76            CALL PROCESS;
77            GO TO LOOP;


78        END DCALC;

          /* ============================================================ */

79        HOWTO: PROC;
80         PUT LIST ('DCALC ACCEPTS THE FOLLOWING FORMS, WHERE V IS A VARIABLE AN
          D E IS AN EXPRESSION:') SKIP;
81         PUT LIST ('      V = E    E =        E        (BLANKS ARE ALWAYS IGNORED)')
           SKIP;
82         PUT LIST (' IN EACH CASE THE EXPRESSION IS EVALUATED AND DISPLAYED, AN
          D, IN THE FIRST CASE,') SKIP;
83         PUT LIST ('      THE VALUE OF V IS REPLACED IN AN INTERNALLY STORED TAB
          LE.') SKIP;
84          PUT LIST ('    ') SKIP;
85         PUT LIST ('IF C IS ANY CONSTANT AND F A FUNCTION NAME, THEN THE FOLLOW
          ING ARE EXPRESSIONS:') SKIP;
86          PUT LIST ('   C  V  (E)  E+E  E-F  E*E  E/E ') SKIP;
87          PUT LIST ('     E.P.E E**E (ALTERNATIVE FORMS FOR EXPONENTIATION)')
               SKIP;
88         PUT LIST ('     F(E) F.(E) (ALTERNATIVE FORMS) WHERE F IS SQRT,SIN,COS
          ,ATAN,EXP,LOG OR ABS') SKIP;
89          PUT LIST ('    ') SKIP;
90         PUT LIST (' CONTROL FUNCTIONS ARE:') SKIP;
91         PUT LIST ('      /T(ERMINATE)') SKIP;
92         PUT LIST ('      /P(RINT) DISPLAYS THE VARIABLE TABLE') SKIP;
93         PUT LIST ('      /E(RASE) ERASES THE VARIABLE TABLE') SKIP;
94         PUT LIST ('      /E(RASE)/V ERASES V FROM THE VARIABLE TABLE') SKIP;
95         PUT LIST ('      /ON (N < 14) CHANGES PRECISION FOR NUMERICAL DISPLAY
          FROM 7 TO N') SKIP;
```

```
 96            PUT LIST (*     ? RE-DISPLAYS THIS SET OF INSTRUCTIONS*) SKIP;                  0115
 97            PUT LIST (*ED) ENTER, ON ADVANCED FEATURES TYPE **H*** ) SKIP;                  0116
 98              PUT LIST (*    *) SKIP;                                                      0117
 99            LOOP: RETURN;                                                                  0118
                                                                                              0119
100           MOREHW2: ENTRY ;                                                               0120
101           PUT LIST (*"V" = F  DEFINES A FUNCTION WHOSE NAME IS V*) SKIP;                  0121
102           PUT LIST (*ITS ARGUMENTS ARE ALL VARIABLES CONTAINED IN E*) SKIP;               0122
103           PUT LIST (*IF A FUNCTION HAS BEEN DEFINED FOR V, AND YOU TYPE  V,*)            0123
              SKIP;                                                                          0124
104           PUT LIST (*     THE EFFECT IS AS IF YOU HAD TYPED **(V = E)***) SKIP;          0125
105           PUT LIST (*TYPING "V" RESULTS IN A DISPLAY OF THE CURRENT FUNCTION AND          0126
               VALUES ASSOCIATED WITH V*) SKIP;                                              0127
                PUT SKIP;                                                                    0128
106           PUT LIST (* "V" APPEARING IN AN EXPRESSION CAUSES THE FUNCTION V TO BE         0129
107           EVALUATED (AND ASSIGNED) IN THAT PLACE*) SKIP;                                 0130
              PUT LIST (*     WHEN THE EXPRESSION IS EVALUATED*) SKIP;                       0131
108           PUT LIST (*FUNCTIONS DEFINED RECURSIVELY (IN TERMS OF THEMSELVES) WILL         0132
109           BE EVALUATED ONLY ONCE IN ANY EXPRESSION*) SKIP;                               0133
              PUT SKIP;                                                                      0134
110           PUT LIST (*? PRECEDING A VARIABLE NAME IN A FUNCTION CAUSES PROMPTING F        0135
111           OR A VALUE EVEN IF IT WAS PREVIOUSLY ESTABLISHED*) SKIP;                       0136
              PUT LIST (*TYPING A NULL LINE CAN HAVE TWO EFFECTS:*) SKIP;                    0137
112           PUT LIST (*     DURING VALUE PROMPTING IT TERMINATES EVALUATION OF THE         0138
113            CURRENT EXPRESSION*) SKIP;                                                    0139
              PUT LIST (*     WHEN A NEW EXPRESSION IS BEING REQUESTED IT CAUSES THE         0140
114            PREVIOUSLY USED EXPRESSION TO BE RE-USED*) SKIP;                              0141
              PUT SKIP;                                                                      0142
115           PUT LIST (*CONSIDER THE FOLLOWING EXAMPLE:*) SKIP;                             0143
116           PUT LIST (* >"X"=X+1*) SKIP;                                                   0144
117           PUT LIST (* >X=0*)SKIP;                                                        0145
118           PUT LIST (* >"Y"=10*"X"**2+B*X+2C*) SKIP;                                      0146
119           PUT LIST (* >Y*) SKIP;                                                         0147
120           PUT LIST (*   A = ?*) SKIP;                                                    0148
121           PUT LIST (* >3*) SKIP;                                                         0149
122           PUT LIST (*   X = 1.000000*) SKIP;                                           0150
123           PUT LIST (*   B = ?*) SKIP;                                                   0151
124           PUT LIST (* >4*) SKIP;                                                         0152
125           PUT LIST (*   C = ?*) SKIP;                                                   0153
126           PUT LIST (* >5*) SKIP;                                                         0154
127           PUT LIST (*   Y = 12.000000*) SKIP;                                         0155
128           PUT LIST (* >       (NOTE: NULL LINE RETURNED BY USER)*) SKIP;                 0156
129           PUT LIST (*   X = 2.000000*) SKIP;                                           0157
130           PUT LIST (*   C = ?*) SKIP;                                                   0158
131           PUT LIST (* >2*) SKIP;                                                         0159
132           PUT LIST (*   Y = 22.000000*) SKIP;                                         0160
133           PUT SKIP; PUT LIST (*TYPING **H** CAUSES A RE-DISPLAY OF THIS SET OF IN      0161
134           STRUCTIONS*) SKIP;                                                             0162
              PUT LIST (*CALC NOW READY FOR INPUT . . .*) SKIP;                              0163
136           RETURN;                                                                        0164
                                                                                             0165
137             END HOWTO;                                                                   0166
                                                                                             0167
                                                                                             0168
              /* ------------------------------------------------------------------- */     0169
                                                                                             0170
                                                                                             0171
```

```
139        PROCESS: PROC;
140        DCL LIST (20) CHAR(22) STATIC;
141            DCL (NL,EQSS) FIXED BINARY STATIC;
142          DCL PREC ENTRY(CHAR(4)) RETURNS(FIXED BIN);
143         DCL (COM1 CHAR (38) INITIAL('THE PRECEDING STATEMENT IS MALFORMED, ')
                , COM2 CHAR(30) INITIAL (' APPEARS TO BE AN OPERATOR    '))
             STATIC;

                         /* SET UP FOR PROCESSING    */
144              IF S(J) = '=' THEN DO; S(J)=' '; J=J-1; END;
149               N=INDEX(SS,'(');
150              IF N=0 THEN EQSS = INDEX(SS,'=');
152                   ELSE EQSS = INDEX(SUBSTR(SS,1,N),'=');
153              IF S(1) = '"' THEN DO;
155                   N = INDEX(SUBSTR(SS,2),'"');
156                   IF N = 0 THEN GO TO ERROR5;
158                   DO N2 = 2 TO N;
159                        IF INDEX(PLIST,S(N2)) ¬=' THEN GOTO ERROR4;
161                   END;
162                   TAB = SUBSTR(SS,2,N-1);
163                   IF S(2) = '?' THEN GO TO ERRORN;
165                   DO N = 1 BY 1 WHILE(¬(TAB=NAME(N)|N>NC));END;
167                   IF EQSS = 0 THEN DO;
169                        IF N > NC THEN GO TO ERROR6;
171                        PUT LIST (' = ' || EXPR(N) || ' = ' ||
                                VALUE(N)) SKIP;
172                   END;
173                   ELSE DO;
174                        IF N>NC & NC>30 THEN DO;
176                             PUT LIST ('TABLE IS FULL') SKIP;
177                             GO TO LOOP;
178                        END;
179                        EXPR(N) = SUBSTR(SS,EQSS+1,J-EQSS);
180                        IF N>NC THEN DO;
182                             NAME(N)=TAB; VALUE (N)='0';
184                             NC=N;
185                        END;
186        IF TST THEN PUT LIST ('  *** NEW ENTRY MADE FOR ' || NAME(N) ||
                 EXPR(N)) SKIP;
188                   END;
189             GO TO LOOP;
190             END;

           /*   NO "     */
191                   ELSE DO;
192                        IF EQSS = 0 THEN DO;
194                             TAB = SUBSTR(SS,1,J);
195                             IF ¬ IC THEN GO TO BLEQ;
197                   DO N = 1 BY 1 WHILE(¬(TAB=NAME(N) | N>NC)); END;
199                             IF N<=NC & LENGTH(EXPR(N))¬=0 THEN DO;
201                               SS='"'||SUBSTR(SS,1,J)||'"';
202                               J = J +2;
203                             END;
204                             ELSE DO;
205                                  BLEQ: SS = ' ='||SS;J=J+2;
207                             END;
208                        END;
```

16

```
209                                   ELSE DO;                                  0229
210                                        TAB= SUBSTR(SS,1,EOSS-1);            0230
211                                        IF ¬ IO THEN GO TO ERRORN;            0231
213                                   END;                                      0232
214                              END;                                           0233
                                                                                0234
                              /*  DYNAMIC SUPPORT  */                           0235
215                     DYNAM:                                                  0236
216                       J = J +1; S(J)='(';                                   0237
217                       NL=0;                                                 0238
218                     DYNAL:                                                  0239
219                      IF TST THEN PUT LIST ('*** ' || SS) SKIP;              0240
220                       N = INDEX(SS,'*');                                    0241
221                       IF N = 0 THEN GO TO PARSE;                            0242
223                       NL = INDEX(SUBSTR(SS,N+1),'*');                       0243
224                       IF NL=0 | NL+N>J THEN GO TO ERROR5;                   0244
226                       TAB = SUBSTR(SS,N+1,NL-1);                            0245
227                       DO N2 = 1 TO NL;                                      0246
228                            IF LIST(N2) = TAB THEN GO TO RMV;                0247
230                       END;                                                  0248
231                       DO N2=1 TO NC;                                        0249
232                            IF NAME(N2)=TAB THEN DO;                         0250
234                                 IF LENGTH(EXPR(N2)) = 0 THEN GOTO ERROR6;   0251
236                                 J=J+LENGTH(EXPR(N2))+1;                     0252
237                                 IF J > 400 | NL>=20 THEN GO TO ERROR7;      0253
239                                 SS=SUBSTR(SS,1,N-1)||'('||                  0254
                                        SUBSTR(SS,N+1,NL-1)||'='||              0255
                                            EXPR(N2)||')'||SUBSTR(SS,NL+N+1);   0256
240                                 NL=NL+1;                                    0257
241                                 LIST(NL)=TAB;                               0258
242                                 GO TO DYNAL;                                0259
243                            END;                                             0260
244                       END;                                                  0261
245                       GO TO ERROR6;                                         0262
246                     RMV:                                                    0263
                            SS=SUBSTR(SS,1,N-1)||SUBSTR(SS,N+1,NL-1)||        0264
                                                 SUBSTR(SS,NL+N+1);           0265
247                         J = J-2;                                            0266
248                       GO TO DYNAL;                                          0267
                                                                                0268
                              /*  PARSE  */                                     0269
                                                                                0270
249                     PARSE:                                                  0271
250                     IF TST THEN DO; PUT LIST('****'||SS) SKIP;              0272
252                       PUT LIST (' ** J = ' || J) SKIP; END;                 0273
254                         CUR = '0'B; CONST = '1'B;                           0274
256                               K = 1; M(1) = '.L';                           0275
258                       S3: DO I = 1 TO J;                                    0276
259                         PREV=CUR;CUR='0'B;                                  0277
261                          IF S(I) = '(' THEN DO; EL='.R'; GO TO LOOPA; END;  0278
266                     IF INDEX(PLIST,S(I)) ¬= 0 THEN DO;  EL = S(I);          0279
269                          IF S(I) = '*' & S(I+1) = '*' THEN DO;              0280
271                                    EL    = '**'; I = I +1; END;            0281
274                          ELSE IF S(I) = '.' & S(I+1) = 'P' & S(I+2) = '.' THEN DO;  0282
276                                 EL    = '***'; I = I + 2; END;              0283
                        /*                                                      0284
                        */                                                      0285
```

```
279                   ELSE IF S(I) = '.' THEN GO TO S4;
281                   CONST = '0';
282                   IF EL = 0 THEN CUR ='L'B; IF EL='(' THEN GO TO STACK;
286                         ELSE GO TO SCAN;
287             END;
288             ELSE IF INDEX(OLIST,S(I))¬=0 & S(I) ¬= ')' THEN DO;
290               S4: CUR = 'L'B;
291                   DO L=I+1 BY 1 WHILE(¬(INDEX(OLIST,S(L))=0 | L>J));END;
293                   IF (S(L) = '-' | S(L) = '+') & S(L-1) = 'E' & L < J   THEN DO;
295                         DO L=L+1 BY 1 WHILE(¬(INDEX(OLIST,S(L))=0 | L>J));END;
297                   END;
298                   IF  S(L-1)= '.' & S(L) = 'P' THEN L = L-1;
300                   IF L-I ≥ 21 THEN GO TO ERROR2;
302                         O1=SUBSTR(SS,I,L-I); EL=O1; I=L-1; GO TO STACK;
306             END;
307             ELSE DO;
308                   CONST = 'D'B;
309                   DO L = I+1 BY 1 WHILE(¬(INDEX(PLIST,S(L))¬=0 | L>J));END;
311                   IF L-I > 10 THEN GO TO ERROR2;
313                   IF S(L) = '.' & S(L+1) = '(' | S(L) = '(' THEN DO;
315                         EL = SUBSTR(SS,I,L-I); IF S(L)=',' THEN L=L+1;
318                         I = L - 1;
319                         GO TO SCAN;
320                   END;
                  /*
                  */
321                   ELSE DO;
322                         CUR = 'L'B;
323                         TAB = SUBSTR(SS,I,L-I);
                                    IF ¬ IC THEN GO TO ERRORN;
324
326             IF IST THEN PUT LIST (' EL***'||TAB) SKIP;
328                         DO N = 1 TO NC;
329                               IF NAME(N)=TAB THEN DO;
331                                     IF S(L) = '=' THEN EL = N;
333                                           ELSE EL = VALUE(N);
334                                     GO TO GETBACK;
335                               END;
336                         END;
337                   END;
338             IF TAB = ' ' THEN DO;
340               EL = 0; GO TO GETBACK;
342             END;

                /* PROMPT FOR A VALUE          */
343                   IF NC >= 30 THEN GO TO ERROR3;
345                   IF TABX(1) = '?' THEN DO;
347                         TAB=SUBSTR(TAB,2);
348                         DO N = 1 TO NC;
349                           IF TAB = NAME(N) THEN GO TO GTV;
351                         END;
352                   END;
353                   N, NC = NC +1; NAME(NC) =TAB; EXPR(NC)='';
356             IF S(L) = '=' THEN DO;
358               VALUE(NC)=0;  EL = NC; GO TO GETBACK;
361             END;
362           GTV:
                        PUT LIST (SUBSTR(TAB,1,INDEX(TAB,' ')) || ' = ') SKIP;
```

```
363                    PUT SKIP;
364          GTM:      GET EDIT (SI) (A(80));
365                 IF SI = ' ' THEN GO TO /* GTM */ LOOP    ;
367                       D1=SI ;R=D1 ;CALL CVT; VALUE(N)   ,EL=R;

371                    GETBACK:;
372           I = L-1;
373                GO TO STACK;
374          END;

                   /*   END PARSE   */

                   /*   EVAL   */

375          STACK:
376           IF TST THEN        PUT LIST ('    ST***'||EL) SKIP;
377                     IF EL = '=' & K<= 2 THEN CONST = '1'B;
379                     K = K + 1;M(K) = EL   ; GC TC ELOOP;

382          SCAN: IF ¬ PREV THEN DO;
384                     K=K+1;M(K)='0'; GO TO STACK; END;
388           IF TST THEN        PUT LIST ('    SC***'||EL) SKIP;
390           DCL  XI  CHAR (4);
391           LOOPA:  XI = M(K-1);  IF PREC(EL) <= PREC(XI) THEN DO;
394               D1 = M(K-2); D2 = M(K);
396                IF XI = '+    ' THEN D1 = D1 + D2;
398           ELSE IF XI = '-    ' THEN D1 = D1 - D2;
400           ELSE IF XI = '*    ' THEN D1 = D1 * D2;
402           ELSE IF XI = '/    ' THEN D1 = D1 / D2;
404           ELSE IF XI = '**   ' & D2=2 THEN D1 = D1 * D1;
406           ELSE IF XI = '**   ' THEN D1 = D1 ** D2;
408           ELSE IF XI = 'SQRT' THEN D1 = SQRT(D2);
410           ELSE IF XI = 'SIN ' THEN D1 = SIN (D2);
412           ELSE IF XI = 'COS ' THEN D1 = COS (D2);
414           ELSE IF XI = 'LOG ' THEN D1 = LOG (D2);
416           ELSE IF XI = 'EXP ' THEN D1 = EXP (D2);
418           ELSE IF XI = 'ATAN' THEN D1 = ATAN(D2,1.00000000E0);
420           ELSE IF XI = 'ABS ' THEN D1 = ABS(D2);
422           ELSE IF XI = '=   ' THEN DO;
424                         R = D2; CALL CVT; IF D1¬=0 THEN VALUE(D1)=R;
428                    IF ¬ CONST THEN DO;
430              IF D1 = 0 THEN PUT LIST ( ' = ' || R) SKIP;
432              ELSE  PUT LIST (SUBSTR(NAME(D1)  ,1,INDEX(NAME(D1),' ')) || ' = ' ||
                            R ) SKIP;
433                   END;   D1 = D2;
435                  END;
436               ELSE DO;
437               MSG = COM1 || M(K-1)||COM2     ;            GO TO LOOPE;
439              END;
440               M(K-2)=D1;
441             IF TST THEN PUT LIST ('  NST***'||M(K-2)) SKIP;
443                 K = K - 2; GC TC LCOPA; END;
446               IF EL     = ',R' THEN DO; R  = M(2); RETURN; END;
451               IF EL     =')' THEN DO; M(K-1)=M(K);K=K-1;GO TO ELOOP; END;
457               GO TO STACK;
458          ELOOP: END S3;
```

```
459        PREC: PROC(X) FIXED BIN;  DCL X CHAR(4);
461              IF X = '+    ' | X = '-    ' THEN RETURN (8);
463              IF X = '*    ' | X = '/    ' THEN RETURN (9);
465              IF X = '**   ' THEN RETURN (10);
467              IF X = '.L   ' THEN RETURN ( 0);
469              IF X = '.R   ' THEN RETURN ( 1);
471              IF X = '(    ' THEN RETURN ( 2);
473              IF X = ')    ' THEN RETURN ( 3);
475              IF X = '=    ' THEN RETURN ( 4);
477              RETURN (11);
478        END PREC;

479        ERROR2: MSG =              'THIS LINE CONTAINS A CONSTANT OR '
                    || 'VARIABLE LONGER THAN ALLOWED';
480         GO TO LOOPE;
481        ERRORN: MSG ='ILLEGAL VARIABLE NAME ' || TAB; GO TO LOOPE;
483        ERROR7: MSG = 'FUNCTION EXPANSION EXCEEDS ALLOWED LIMITS'; GO TO LOOPE;
485        ERROR6: MSG = 'NO EXPRESSION FOUND FOR ' || TAB ; GO TO LOOPE;
487        ERROR5:MSG = 'UNBALANCED " IN PRECEDING LINE'; GO TO LOOPE;
489        ERROR4: MSG = 'OPERATOR ON LEFT HAND SIDE IN PRECEDING LINE';
490           GO TO LOOPE;
491        ERROR3: MSG = 'THE VARIABLE TABLE IS FULL;' || TAB || ' IS UNDEFINED';

492         LOOPE: PUT LIST (MSG) SKIP;
493         LOOP: RETURN;
494          END PROCESS;

           /* ============================================================ */

495         SUPPORT1: PROC;
496        DCL BIT CHAR(11) INITIAL('           ') STATIC;

497              IF S(2) = 'P' THEN GO TO PRINT;
499              ELSE IF S(2) = 'E' THEN DO;
501                    I=INDEX(SUBSTR(SS,2,J-1),'/'); IF I=0 THEN GO TO ERASE;
504                    ELSE GO TO FLIST;
505              END;
506              ELSE IF S(2) = 'D' THEN DO;  PRCSN = SUBSTR(SS,3,2);
509                IF PRCSN > 14 THEN PRCSN = 14;
511              END;
512              ELSE PUT LIST ('UNDEFINED CONTROL CODE') SKIP;
513         LOOP: RETURN;

            /*
                                     STORAGE OPERATIONS
            */

514        ERASE: NC =  0; GO TO LOOP;
516        FLIST:
                 I = I + 1;
517              T = SUBSTR(SS,I+1,J-I);
518              DO N = 1 TO NC;
519                    IF NAME(N) = T THEN DO;
521                  TABLE (N)=TABLE(NC);NC=NC-1; GO TO LOOP;
```

```
 524                    END;
 525              END; GO TO LOOP;

 527        PRINT:
                 DO I=3 BY 3 TO NC-1;
 528                  MSG = '      ' || NAME(I-2)||B11||NAME(I-1)||B11||NAME(I);
 529                      PUT LIST (MSG) SKIP;
 530                      MSG = VALUE(I-2)||VALUE(I-1)||VALUE(I);
 531                      PUT LIST (MSG) SKIP;
 532                      PUT LIST ('    ') SKIP;
 533             END;

 534             K = I-2;
 535             MSG = '     ';
 536             DO I=K TO NC;
 537                      MSG = MSG || NAME(I)||B11;
 538             END;
 539             PUT LIST (MSG) SKIP;
 540             MSG = '';
 541             DO I=K TO NC;
 542                      MSG = MSG || VALUE(I);
 543             END;
 544             PUT LIST (MSG) SKIP;
 545             GO TO LOOP;

           /*
                                       ERROR MESSAGE SETUP
           */
 546        END

 547       CVT: PROC;
 548        DCL (I,J) STATIC;
 549               I= INDEX(R,'E');
 550               IF I¬=0 THEN DO ;
 552                 J=SUBSTR(R,I+2);
 553                IF J < PRCSN THEN DO;
 555                    IF J = 0 THEN DO;
 557                          R = SUBSTR(R,1,PRCSN+2);
 558                    END; ELSE IF SUBSTR(R,I+1,1) = '-' THEN DO;
 561                          R = SUBSTR(R,1,1) || SUBSTR('.0000000000',1,J) ||
                                  SUBSTR(R,2,1) || SUBSTR(R,4,PRCSN);
 562                    END ; ELSE DO;
 564                          R = SUBSTR(R,1,2)||SUBSTR(R,4,J)||'.' ||
                                  SUBSTR(R,4+J,PRCSN-J-1);
 565                    END;
            /*      IF TST THEN PUT LIST ('****'||R||'*') ;        */
 566                END;
 567               END;
 568          RETURN;
 569        END CVT;


 570       ID: PROC BIT(1);
 571        DCL I STATIC;
```

21

```
572          IF TARX(1) = '?' & TARX(2) = ' ' THEN RETURN ('0'B);
574          IF INDEX(XLIST,TARX(1))¬=0 THEN RETURN('0'B);
576          DO I=1 BY 1 WHILE(¬(TARX(I)= ' ' | INDEX(OLIST,TARX(I))¬=0));
577          END;
578          IF TARX(I)= ' ' THEN RETURN('1'B);
580          RETURN('0'B);
581       END ID;

582    END RCALC;
```