DECUS

PROGRAM LIBRARY

DECUS NO.	8-329a
TITLE	FOCARL, Version 14
AUTHOR	David Wolfe
COMPANY	Carleton College Northfield, Minnesota
DATE	May 14, 1973
SOURCE LANGUAGE	PAL

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.

FOCARL AND COLPAC

MANUAL

REVISED
SPRING, 1973

CARLETON COLLEGE
NORTHFIELD, MINNESOTA 55057

ACKNOWLEDGEMENTS

THE PROGRAMMING LANGUAGE, FOCAL, OF WHICH COLPAC AND FOCARL ARE EXTENSIONS WAS ORIGINALLY DESIGNED BY RICK MERRILL OF DIGITAL EQUIPMENT CORPORATION.

COLPAC WAS ORIGINALLY DESIGNED BY MARK BRAMHALL. FOCARL, AN ADAPTATION OF COLPAC, WAS WRITTEN BY TED EMIGH AND DAVID WOLFE.

PORTIONS OF THIS MANUAL WERE WRITTEN BY JIM WREDE AND DAVID WOLFE.

PLEASE ADDRESS CORRECTIONS AND SUGGESTIONS FOR ADDITIONS TO
R. W. NAU, DEPT. OF MATH.

Attn. Assembly Programmers:

1. Room is left in the command and function tables for additional functions and commands.
2. There are over 110_8 contiguous unused core locations available for new coding.
3. The general disk-read routine in FOCARL, "SWAP", may be used to read over 500_8 core locations of code to perform special functions such as implement dectape, graphics, or four-word floating point routines.
4. There are several unused page zero locations.
5. The most important aspect of FOCARL is that is a program totally different from FOCAL; It is a recompilation, not an overlay.

Hardware EAE

Hardware EAE is standard in FOCARL. That is, it is edited into the source. But, non-EAE coding has been left in also and may be used by making the patch described in the comments in the source at "MP4".

Colpac Programmers:

The PLOT, OPTION SCOPE, ERASE SCOPE, and TYPE/ASK & commands are ignored, so your programs will run in FOCARL provided that they are not too large or generate too many in-core variables.

Compiling FOCARL:

The source for FOCARL is in two parts for ease in editing, etc. The running version of FOCARL is an 18 disk segment file (TSS/8 size disk segments). The first 16 segments are created by compiling the source, as it is, using PAL8. That is, no additions are necessary. The 17th and 18th segments are created by compiling the source with the label, HIGH, defined to be 0000. An example of responses to PAL8 follows:

```
.R PAL8
*FBIN1.BN←FOC14A.PA,FOC14B.PA

ERRORS DETECTED: 0

LINKS GENERATED: 0

.R PAL8
*FBIN2.BN←TTY:,DSK:FOC14A.PA,DSK:FOC14B.PA
HIGH=0000    (RETURN)
CTRL/Z       (WHEN PAL8 STOPS)
CTRL/Z       (LETS PAL8 CONTINUE WITH PASS 2)

ERRORS DETECTED: 0

LINKS GENERATED: 0
```

When PAL8 has finished, FBIN1.BN will be the binary file containing coding which occupies segments 1 through 16 of the TSS/8 disk file. FBIN2.BN will contain coding which occupies segments 17 and 18 of the disk file, FOCARL. Because TSS/8 LOADER loads into core, and because FOCARL requires all 4K of core, several core locations beginning about location 7740_8 will not have the proper contents after loading FBIN1. So, the correct contents, as read from the listing, will have to be hand-patched. FBIN2 will have to be loaded separately from the main file of 16 segments by using LOADER to load FBIN2 into core and then by appending it to the main file using Monitor commands;

```
.SAVE FOCARL 10000 0
.
.
```

The above-described procedure for loading FBIN1 and FBIN2 and creating the SAVE-format file, FOCARL, is awkward, particularly because LOADER loads into core and not, say, into a disk file. This procedure is for use under the TSS/8 timesharing system. There must be a better way...

CONTENTS

SECTION

1. INTRODUCTION

THIS MANUAL DESCRIBES THE PROGRAMMING LANGUAGES COLPAC (CARLETON ON-LINE LANGUAGE FOR PLOTTING AND ARITHMETIC CALCULATION) AND FOCARL (FOR FORMULATING ON-LINE CALCULATIONS IN ALGEBRAIC LANGUAGE AT CARLETON COLLEGE). BOTH ARE ADAPTATIONS OF FOCAL WHICH IS BASED ON JOSS.

COLPAC IS AN 8K VERSION OF FOCAL WHICH RUNS IN AN 8/L COMPUTER. IT FEATURES GRAPHICS AND ACCOMMODATES LARGER PROGRAMS AND MORE VARIABLES THAN FOCAL AND FOCARL.

FOCARL RUNS UNDER THE TSS/8 TIMESHARING SYSTEM (EDUSYSTEM 50), WHICH REQUIRES THAT IT RUN IN ONLY 4K OF CORE, HALF THE WORKING SPACE OF COLPAC. FOCARL FEATURES RANDOM ACCESS DISK DATA STORAGE.

BOTH LANGUAGES SUPERCEDE FOCAL IN COMMAND FLEXIBILITY; LIBRARY COMMANDS, RELATIONAL IF, COMPUTED GOTO, COMPUTED I/O FORMAT CONTROL, OPTIONS (READER, PUNCH, LINEPRINTER, ETC), BREAK FROM FOR LOOP, DECREMENT IN FOR LOOP, AND DOUBLE SUBSCRIPTING ARE STANDARD. COLPAC'S GRAPHICS COMMANDS ARE IGNORED BY FOCARL, SO ANY SMALL COLPAC PROGRAM CAN BE RUN IN FOCARL.

THIS MANUAL DESCRIBES THE SYNTAX AND USE OF COMMANDS FOR BOTH COLPAC AND FOCARL. FOR THE MOST PART, THESE LANGUAGES ARE IDENTICAL IN COMMAND REPERTOIRE. HOWEVER, THERE ARE SOME COMMANDS SPECIFIC TO ONE AND NOT THE OTHER. TO HELP CLARIFY DIFFERENCES BETWEEN THE TWO, THE FOLLOWING CONVENTION HAS BEEN EMPLOYED IN THIS MANUAL:
"*C*" PRECEDES ANY LINES IN THIS MANUAL WHICH ARE SPECIFIC TO COLPAC.
"*F*" PRECEDES THOSE WHICH ARE SPECIFIC TO FOCARL.
IF ONE OF THESE FLAGS APPEARS IN A SECTION HEADING, THEN THAT ENTIRE SECTION IS SPECIFIC TO THE LANGUAGE DENOTED BY THE FLAG.
ALL OTHER REFERENCES TO FOCARL AND COLPAC WHICH APPEAR THROUGHOUT THIS MANUAL APPLY TO BOTH LANGUAGES.

2. ABBREVIATIONS

1) THE ONLY SPACES NECESSARY IN A LINE ARE THE SPACES SEPARATING THE COMMAND (AND SUB-COMMANDS AND OPTIONAL PARAMETERS IN LIBRARY AND OPTION COMMANDS) FROM ITS ARGUMENT AND SEPARATING COMMAND LINES FROM LINE NUMBERS. THE OTHER SPACES SHOWN IN THE COMMAND LINES IN THIS MANUAL ARE INCLUDED FOR EASE OF READING.
2) AN ARITHMETIC EXPRESSION, AS USED IN THE EXPLANATIONS FOLLOWING, MEANS AN EXPRESSION INCLUDING CONSTANTS, VARIABLES, OR FUNCTION VALUES RELATED VIA THE FIVE ARITHMETIC OPERATIONS (+ - * / ^).

```
   EXAMPLES: -7.2  B-A  FSQT(3*2.2^4/17.4)  FSQT(FABS(FSIN(B/2)))^3
```

3) ALL COMMANDS IN FOCARL MAY BE ABBREVIATED TO THEIR INITIAL LETTERS. THUS, ANY WORD WITH THE SAME INITIAL LETTER AS A FOCARL COMMAND WILL BE INTERPRETED AS THAT COMMAND. SEE THE EXAMPLE ON THE NEXT PAGE.

COMMAND	EFFECT
T A, TYPE A, OR TRY A	TYPES THE VALUE OF VARIABLE A
G , GO , OR GOSH	EXECUTES PROGRAM STARTING AT BEGINNING
D A, DO ALL, OR DARNIT ALL	EXECUTES PROGRAM STARTING AT BEGINNING

THERE ARE TWO PAIR OF COMMANDS WITH THE SAME INITIAL LETTERS:

```
GO/GOTO           LOGOUT/LIBRARY
```

```
G WITHOUT AN ARGUMENT IS GO--BEGIN EXECUTION AT LOWEST LINE NUMBER
G WITH AN ARGUMENT (E.G.,G 1.07) IS GOTO--BEGIN EXECUTION AT LINE
    NUMBER SPECIFIED
```

```
L WITHOUT AN ARGUMENT!  LOGOUT--RETURN TO MONITOR
L WITH ARGUMENTS:  LIBRARY COMMAND
    FIRST ARGUMENT SHOULD BE SUB-COMMAND (C,D,S,X)
*F* FIRST ARGUMENT MAY ALSO BE "I" OR "O".
    SECOND ARGUMENT IS FILE NAME
    THIRD ARGUMENT IS ACCOUNT NO. (L X,L C COMMANDS ONLY)
*F* ACCOUNT NO. MAY ALSO BE USED WITH "L I" COMMAND.
*F* FOURTH ARGUMENT IS LINE NUMBER (L X COMMAND ONLY)
```

3. PROGRAM PREPARATION

THE FORM OF A COLPAC OR FOCARL PROGRAM

A DIRECT COMMAND IS ONE WHICH IS EXECUTED IMMEDIATELY AFTER THE CARRIAGE RETURN TERMINATING THE LINE HAS BEEN TYPED. THIS DIFFERS FROM AN INDIRECT COMMAND WHICH BEGINS WITH A LINE NUMBER. A PROGRAM, AS IT IS USUALLY DEFINED, IS A SERIES OF COMMANDS WHICH ARE EXECUTED EITHER COMPLETELY SEQUENTIALLY OR IN AN ORDER UNDER THE CONTROL OF THE PROGRAM ITSELF. PROGRAM CONTROL AND BRANCH STATEMENTS WILL BE EXPLAINED LATER. WE WILL CONTINUE NOW IN OUR EXPLANATION OF INDIRECT PROGRAMMING.

LINE NUMBERS AND GROUP NUMBERS

A LINE NUMBER IN FOCARL HAS A VALUE BETWEEN 1.01 AND 31.99, INCLUSIVE. THE NUMBER HAS TWO DIGITS, NOT BOTH ZERO, TO THE RIGHT OF THE DECIMAL POINT. A SPACE BETWEEN THE LINE NUMBER AND THE COMMAND IS REQUIRED.

A GROUP NUMBER IS AN INTEGER BETWEEN 1 AND 31, INCLUSIVE. IT IS USED TO REFER TO ALL LINES, THE INTEGER PART OF WHOSE LINE NUMBER IS EQUAL TO THAT NUMBER. THUS, GROUP ONE INCLUDES ALL LINES NUMBERED 1.01-1.99; GROUP TWO, LINES 2.01-2.99; ETC. GROUP NUMBERS ARE USED PRIMARILY FOR REFERENCING SUBROUTINES.

DIRECT VS. INDIRECT PROGRAMMING

USING INDIRECT COMMANDS OFFERS SEVERAL ADVANTAGES OVER PROGRAMMING WITH DIRECT COMMANDS. TO INSERT A STEP, ONE NEED ONLY ASSIGN IT A LINE NUMBER BETWEEN THE NUMBERS OF THE OTHER LINES. DIRECT PROGRAMMING REQUIRES THE USER TO RETYPE THE ENTIRE SERIES OF STEPS. IN INDIRECT PROGRAMMING, ONE CAN DELETE OR CHANGE A LINE BY MERELY TYPING ANOTHER LINE WITH THE SAME LINE NUMBER. DIRECT PROGRAMMING AGAIN REQUIRES COMPLETE RETYPING. IN INDIRECT PROGRAMMING, THE USER IS ALSO ABLE TO LIST OUT HIS PROGRAM, AS IT EXISTS, AT ANY TIME, EVEN DURING EXECUTION. DIRECT PROGRAMMING REQUIRES ONE TO LOOK BACK OVER THE TTY PRINT-OUT AND TRY TO SEPARATE THE PROGRAM FROM OUTPUT, FALSE STARTS, ETC. DIRECT PROGRAMMING IS, BY DEFINITION, COMPLETELY SEQUENTIAL, SINCE COMMANDS ARE EXECUTED IN THE ORDER THEY ARE ENTERED. THIS IS THE PRINCIPAL DISADVANTAGE OF DIRECT PROGRAMMING, SINCE IT CANNOT INCLUDE CONDITIONAL STATEMENTS TO ALTER THE ORDER OF EXECUTION. THIS IS WHY DIRECT PROGRAMMING IS SELDOM USED, EXCEPT FOR SHORT, STRAIGHT-FORWARD CALCULATIONS.

MORE THAN ONE COMMAND PER LINE

MULTIPLE COMMANDS MAY BE SPECIFIED ON ONE LINE, SO LONG AS THEY ARE SEPARATED BY SEMI-COLONS. THERE ARE CERTAIN RESTRICTIONS UPON THIS:

1) ANY COMMAND STRING FOLLOWING A FOR COMMAND ON THE SAME LINE WILL BE EXECUTED ONCE FOR EACH ITERATION OF THE FOR COMMAND.

2) ANY COMMAND FOLLOWING A COMMENT STATEMENT ON THE SAME LINE IS ASSUMED BY FOCARL TO BE PART OF THE COMMENT AND WON'T BE EXECUTED.

3) ANY COMMAND FOLLOWING AN ERASE TEXT, MODIFY, LIBRARY DELETE, LIBRARY CALL, OR LIBRARY XTEND COMMAND ON THE SAME LINE WILL NOT BE EXECUTED.

4. EXECUTION COMMANDS: GO, GOTO, DO

GO , DO , OR DO ALL CAUSES EXECUTION OF THE PROGRAM STARTING WITH THE SMALLEST LINE NUMBER
GOTO AB.CD , USED AS A DIRECT COMMAND, STARTS PROGRAM EXECUTION AT LINE AB.CD

DO B AS A DIRECT COMMAND, EXECUTES ONLY GROUP B.
DO AB.CD EXECUTES ONLY LINE AB.CD

5. THE USE OF SPECIAL KEYS IN FOCARL

PANIC BUTTONS

↑C (↑ INDICATES THE CONTROL KEY, MARKED CTRL) IS USED TO STOP EXECUTION OF A PROGRAM. CONTROL STAYS IN FOCARL INDICATED BY AN "*". THE CURRENT PROGRAM IS PRESERVED.
F ALL TEXT I/O RETURNS TO THE TELETYPE.
F %L I/O MODE (DISK DATA FILE I/O) STATUS REMAINS UNCHANGED.

F ↑BS (CONTROL B THEN JUST S) IS USED AS AN ABNORMAL EXIT FROM
F FOCARL. CONTROL RETURNS TO MONITOR, INDICATED BY A "." THE CURRENT
F PROGRAM MAY BE RESTORED BY TYPING "START" AND RETURN, PROVIDED NO
F INTERVENING COMMANDS ARE TYPED (EXCEPT FOR A FEW WHICH ARE DESCRIBED
F IN DEC'S "INTRODUCTION TO PROGRAMMING")
F ↑BS IS ALSO USED TO EXIT FROM OTHER PROGRAMS THAN FOCARL.

WHILE ENTERING PROGRAMS

CARRIAGE RETURN (CR)

TYPING A CARRIAGE RETURN DIRECTS FOCARL TO ANALYZE AND EXECUTE THE LINE JUST TYPED IN. UNTIL A CARRIAGE RETURN IS TYPED, FOCARL MERELY READS IN THE CHARACTERS ONE-BY-ONE AND STORES THEM. FOCARL ALSO GENERATES A LINE FEED IN RESPONSE TO EVERY USER-TYPED CARRIAGE RETURN.

LINE FEED (LF)

LINEFEEDS ARE IGNORED WHEN ENTERING PROGRAMS

RUBOUT

RUBOUT WILL DELETE ONE PREVIOUSLY TYPED CHARACTER FOR EACH TIME IT IS STRUCK, UP TO THE "*" AT THE LEFT MARGIN.

BACKARROW (SHIFT O)

WHEN TYPING IN A PROGRAM LINE OR COMMAND STRING, THIS KEY DELETES THAT PART OF THE LINE TO THE LEFT OF IT.

SEE THE SECTION ON THE MODIFY COMMAND FOR THE USE OF SPECIAL KEYS WHILE MODIFYING A PROGRAM LINE.

DURING INPUT AND OUTPUT

RUBOUT

IN RESPONSE TO AN INPUT REQUEST,
C IT WILL DELETE CHARACTERS ONE AT A TIME BACK TO THE ":".
F IT WILL BE IGNORED.

BACKARROW (SHIFT O)

IN RESPONSE TO AN ASK COMMAND, DELETES THE VALUE JUST TYPED IN, ALLOWING THE USER TO TYPE IN ANOTHER VALUE.

↑G

USED TO RING THE BELL

ALTMODE

WHEN USED IN RESPONSE TO AN INPUT REQUEST, IT DIRECTS FOCARL TO RETAIN THE FORMER VALUE OF THE VARIABLE.

6. VARIABLES IN FOCARL

VARIABLE NAMES

A VARIABLE NAME IN FOCARL MUST BEGIN WITH A LETTER OTHER THAN F. F IS USED TO DENOTE A FUNCTION CALL AND CANNOT BE USED AS THE FIRST LETTER OF A VARIABLE NAME. AFTER THE INITIAL LETTER, ANY ALPHANUMERIC CHARACTER (A LETTER A-Z OR A DIGIT 0-9) MAY FOLLOW. SOME SHIFT CHARACTERS ARE ALLOWED AS WELL, BUT USUALLY WITH UNDESIRED EFFECTS IN ASK AND TYPE COMMANDS. E.G., ! " # $ %, ETC. FOCARL ONLY REMEMBERS THE FIRST TWO CHARACTERS OF A VARIABLE NAME, SO ANY TWO SIMPLE VARIABLES BEGINNING WITH THE SAME TWO CHARACTERS ARE CONSIDERED TO BE THE SAME.

```
        *SET VIAL=12                  *SET S123=7.02
        *SET VILE=35                  *SET S124=14/3
        *TYPE VIAL,VILE               *TYPE S123,S124
          35   35 *                    4.667  4.667 *
```

SUBSCRIPTED VARIABLES

YOU MAY DIFFERENTIATE BETWEEN TWO VARIABLE NAMES WITH THE SAME FIRST TWO (OR MORE) CHARACTERS BY USING SUBSCRIPTS. VARIABLES MAY HAVE SINGLE OR DOUBLE SUBSCRIPTS. THE LIMITS ON A SINGLE SUBSCRIPT ARE FROM 1 TO 2047. IN A DOUBLE SUBSCRIPTED VARIABLE, THE FIRST SUBSCRIPT IS A POSITIVE INTEGER LESS THAN 32, THE SECOND LESS THAN 64. THE SUBSCRIPT (EITHER SINGLE OR DOUBLE) MUST BE ENCLOSED IN PARENTHESES. IN A DOUBLE SUBSCRIPTED VARIABLE, THE TWO SUBSCRIPTS ARE SEPARATED BY A COMMA. SUBSCRIPTS MAY BE EXPRESSIONS, IN WHICH CASE THEIR VALUES ARE TRUNCATED.

LISTING THE DEFINED VARIABLES

TO GET A LIST OF THE VARIABLES THAT HAVE BEEN DEFINED AND THEIR CURRENT VALUES (STORED IN THE SYMBOL TABLE), TYPE *TYPE $

7. ASSIGNMENT COMMAND: SET

THE SET COMMAND ASSOCIATES A NUMBERIC VALUE WITH A VARIABLE NAME. THE SYNTAX OF THE SET COMMAND IS: SET VA=AE, WHERE VA IS A VARIABLE AND AE IS AN ARITHMETIC EXPRESSION; E.G., A CONSTANT, VARIABLE, OR FUNCTION. EXAMPLE:

```
*
*SET A = 3.02; SET B = 6.17; SET C(13) = -427; TYPE $
A =    3.02000
B =    6.17000
C (13)= -427.00000
*
```

8. ARITHMETIC OPERATIONS AND PRIORITIES

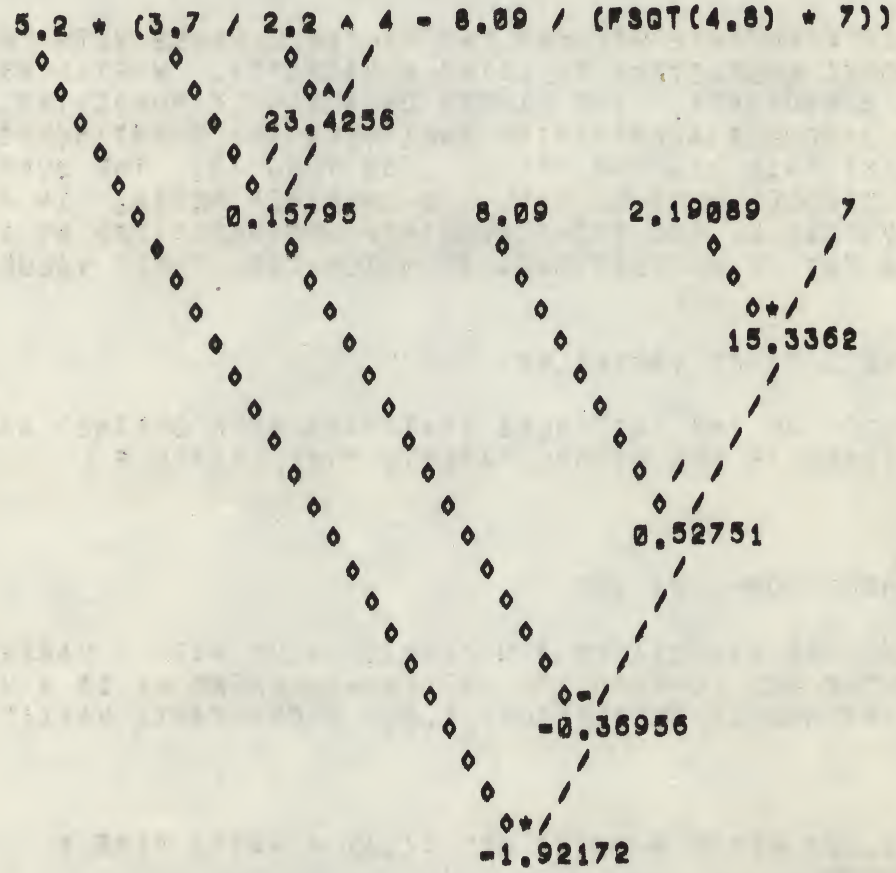
+	ADDITION
-	SUBTRACTION
*	MULTIPLICATION
/	DIVISION
^	EXPONENTIATION

THE ORDER OF PRIORITY OF ARITHMETIC OPERATIONS IS:

1) EXPONENTIATION
2) MULTIPLICATION AND DIVISION (EQUAL PRIORITY)
3) ADDITION AND SUBTRACTION (EQUAL PRIORITY)

EVALUATION PROCEEDS FROM LEFT TO RIGHT ACCORDING TO THE ABOVE PRIORITIES, RESPECTING THE ASSOCIATIONS OF PARENTHESES.

THUS, THE EXPRESSION, 5.2*(3.7/2.2^4-8.09/(FSQT(4.8)*7)) IS EVALUATED AS FOLLOWS:



IN THE DIAGRAM ABOVE, THE LINES GOING DOWN TO THE RIGHT TRACE THE FIRST VALUES TO BE USED BY THE ARITHMETIC OPERATORS (+,-,*,/,^); THE LINES GOING DOWN TO THE LEFT TRACE THE SECOND ARGUMENT OF THE OPERATORS. WHEN THE TWO LINES MEET, THE OPERATION SPECIFIED BY THE OPERATOR LISTED ABOVE THEIR INTERSECTION IS EXECUTED AND THE RESULT TYPED AT THE POINT OF THEIR INTERSECTION.

SEE THE APPENDICES FOR A DESCRIPTION OF THE AVAILABLE FUNCTIONS.

9. INPUT/OUTPUT COMMANDS: ASK, TYPE

ASK

THE ASK COMMAND IS THE INPUT COMMAND OF THE FOCARL LANGUAGE. IT ALLOWS THE USER TO ENTER NUMERICAL VALUES WHICH THE COMPUTER STORES AND REFERENCES AS NAMED VARIABLES. FOCARL TYPES A ":" TO INDICATE THAT IT IS WAITING FOR A NUMERICAL VALUE FOR A VARIABLE TO BE ENTERED. THE USER MUST TYPE A DELIMITER (SEE NEXT PARAGRAPH FOR ALLOWABLE DELIMITERS) FOLLOWING THE NUMBER, BEFORE FOCARL WILL RECOGNIZE THE VALUE. THIS IS NECESSARY BECAUSE FOCARL DOES NOT USE FORMATTED INPUT. IF A MISTAKE IS MADE IN TYPING IN AN INPUT VALUE AND A DELIMITER HAS NOT BEEN TYPED, THE ERROR CAN BE CORRECTED BY:
C STRIKING THE RUBOUT KEY WILL DELETE ONE CHARACTER FOR EACH TIME IT
C IS DEPRESSED, UP TO THE ":".
TYPING A BACKARROW (SHIFT O) DELETES THE ENTIRE RESPONSE AND ALLOWS THE USER TO TYPE IN THE CORRECT VALUE. HITTING THE ALTMODE KEY IN RESPONSE TO THE ASK STATEMENT'S ":" INSTRUCTS FOCARL TO RETAIN THE FORMER VALUE OF THE VARIABLE, INSTEAD OF ACCEPTING AN INPUT VALUE.

THE FOLLOWING CHARACTERS MAY BE USED AS DELIMITERS FOLLOWING A VALUE INPUT IN RESPONSE TO AN ASK COMMAND:

COMMA, COLON, SEMI-COLON, CARRIAGE RETURN, OR A SPACE

ANY OF THE DELIMITERS, EXCEPT A SPACE, MAY BE USED WITHOUT TYPING IN A VALUE TO SPECIFY A VALUE OF ZERO FOR THE VARIABLE. A SPACE MUST BE PRECEDED BY A VALUE, A MINUS SIGN OR A DECIMAL POINT BEFORE IT WILL SERVE AS A DELIMITER.

TYPE

THE TYPE COMMAND IS FOCARL'S OUTPUT COMMAND. IT IS USED TO PRINT OUT THE RESULTS OF THE COMPUTER'S CALCULATIONS SO THEY CAN BE SEEN AND INTERPRETED BY THE USER.
EXAMPLES:

```
*
*SET B=6
*ASK "WHAT ARE THE VALUES OF",!,?A?,!,?B?,!,?C?,!
WHAT ARE THE VALUES OF
A:-2    SPACE USED AS DELIMITER
B:      ALTMODE
C:7.02  COMMA USED AS DELIMITER
*TYPE %4.04,A,B,C,!
-2.000 6.000 7.020
*
*TYPE %4,360,!,1130,!,1401,!,1620,!
  360
 1130
 1401   NOTE THAT MULTIPLE ITEMS CAN BE OUTPUT IN ONE TYPE OR ASK
 1620   COMMAND SO LONG AS THE ITEMS ARE SEPARATED BY COMMAS.
*
```

10. NUMERIC FORMATTING IN THE TYPE STATEMENT

THERE ARE THREE TYPES OF NUMERIC FORMATTING AVAILABLE IN FOCARL: INTEGER, FLOATING-POINT OR DECIMAL, AND EXPONENTIAL OR E FORMAT. THE FORMAT IS SPECIFIED BY USING THE % OPTION.

```
  %B     YIELDS B-PLACE INTEGER OUTPUT;
  %A.BC  YIELDS A-PLACE OUTPUT WITH UP TO BC DIGITS TO THE RIGHT OF
         THE DECIMAL POINT;
  %      YIELDS EXPONENTIAL FORMAT:  A SIX-PLACE DECIMAL FOLLOWED
         BY THE LETTER E, FOLLOWED BY THE POWER OF 10 WHICH THE
         DECIMAL IS TO BE MULTIPLIED BY TO YIELD THE CORRECT VALUE.
```

THE ARGUMENT OF THE % OPTION MAY BE AN ARITHMETIC EXPRESSION.

IF A NUMBER CONTAINS MORE SIGNIFICANT DIGITS TO THE LEFT OF THE DECIMAL POINT THAN IS ALLOWED FOR BY THE INTEGER OR FLOATING-POINT FORMAT, THAT NUMBER WILL BE OUTPUT UNDER A MODIFIED E-FORMAT: ONLY THE NUMBER OF SIGNIFICANT DIGITS (UP TO 6) GIVEN IN THE SPECIFIED FORMAT ARE RETAINED IN THE DECIMAL PART OF THE E-FORMATTED NUMBER.

IF A NUMBER CONTAINS MORE SIGNIFICANT DIGITS TO THE RIGHT OF THE DECIMAL THAN IS ALLOWED UNDER THE OUTPUT FORMAT, THE NUMBER WILL BE ROUNDED TO FIT THE SPECIFIED FORMAT.

ONCE A FORMAT IS SPECIFIED, OUTPUT WILL CONTINUE TO BE TYPED UNDER THIS FORMAT UNTIL ANOTHER FORMAT IS REQUESTED OR A DEFAULT CONDITION CAUSES OUTPUT TO BE TYPED IN E-NOTATION AS EXPLAINED ABOVE.

THE THREE COMMANDS, ASK, SET, AND TYPE, FORM THE BASIS OF ANY PROGRAM. THEY PROVIDE THE ESSENTIAL TASKS OF INPUT, ASSIGNMENT, AND OUTPUT, RESPECTIVELY. IN THE FOLLOWING EXAMPLE, THE GO COMMAND IS USED TO BEGIN EXECUTION OF AN INDIRECT PROGRAM.

```
*
*WRITE
C FOCARL-14

01.13 TYPE "THIS IS A DEMONSTRATION PROGRAM"!
01.15 ASK "WHAT IS THE VALUE OF A",A,!
01.17 SET B=FSQT(A);SET C=A*A;SET D=A^3
01.19 TYPE "  A     SQ. RT. OF A     A^2    A^3"!
01.21 TYPE A,"  ",B,"    ",C,"  ",D,!
*
*GO
THIS IS A DEMONSTRATION PROGRAM
WHAT IS THE VALUE OF A:13
      A     SQ. RT. OF A   A^2    A^3
 13.0000      3.60555     169.000 2197.00
*
*
```

11. ASCII FORMAT INPUT/OUTPUT

EACH CHARACTER ON THE TTY KEYBOARD IS INTERPRETED AND STORED BY THE COMPUTER AS A NUMBER. THE CODE WHICH THE COMPUTER USES IS CALLED ASCII, THE UNITED STATES OF AMERICA STANDARD CODE FOR INFORMATION INTERCHANGE. THIS CODE, OR A MORE COMPACT FORM OF IT, IS USED IN ALL INTER-MACHINE COMMUNICATIONS. THE %A FORMATTING OPTION IN FOCARL ALLOWS THE USER TO WORK WITH BOTH THE NUMERIC AND CHARACTER REPRESENTATIONS OF EACH CHARACTER. FOR A LIST OF THE CHARACTER SET AND THE NUMERIC CODES CORRESPONDING TO EACH CHARACTER, SEE THE APPENDICES.

%A INPUT

SPECIFYING ASCII FORMAT IN AN ASK STATEMENT RESULTS IN TWO IMMEDIATELY NOTICEABLE CHANGES: THE ":" IS NOT TYPED, AS IT IS UNDER A NUMERIC FORMAT, AND ONLY ONE CHARACTER IS ACCEPTED FOR EACH VARIABLE ASKED. WHEN A CHARACTER, FOR EXAMPLE, A, IS TYPED IN RESPONSE TO AN ASK STATEMENT, THE NUMERIC VALUE OF A, 193, IS STORED AS THE VARIABLE.

%A OUTPUT

WHEN %A IS SPECIFIED IN A TYPE STATEMENT, THE COMPUTER WILL TYPE OUT ALL VALUES AS THEIR CHARACTER COUNTERPARTS, IF THEY ARE DEFINED. FOCARL TRIES TO FIND A CHARACTER COUNTERPART FOR THE INTEGER PART OF THE NUMERIC VALUE. THE INTEGERS FROM 128 TO 255 HAVE DIRECT CHARACTER CODES IN FOCARL, ALTHOUGH SOME ARE NON-PRINTING. THE SAME CHARACTER WILL BE PRODUCED BY ANY NUMBER WHICH DIFFERS FROM THE DIRECT-CODED NUMBER BY AN INTEGER MULTIPLE OF 128. FOR EXAMPLE, THE DIRECT CODE FOR A IS 193. THE NUMBERS 65, 321, 449, 3009, AND -3007 WILL ALSO PRODUCE THE LETTER "A" WHEN OUTPUT UNDER %A FORMAT.

NOTE
JUST AS WITH ANY NUMERIC FORMAT, ALL INPUT AND OUTPUT WILL BE HANDLED UNDER THIS FORMAT UNTIL ANOTHER FORMAT IS SPECIFIED.

EVALUATION OF ALPHABETIC RESPONSES TO INPUT

FOCARL WILL ACCEPT AND EVALUATE ALPHABETIC CHARACTER STRINGS TYPED IN RESPONSE TO A NUMERICALLY FORMATTED ASK STATEMENT. THE VALUES ASCRIBED TO THE LETTERS (EXCEPT E) CORRESPOND TO THEIR POSITION IN THE ALPHABET, I.E., A=1, B=2, Z=26, M=13, ETC. E DENOTES EXPONENT, OR POWER OF 10. WHEN E IS THE FIRST CHARACTER OF A STRING TYPED IN RESPONSE TO AN INPUT REQUEST, THE VALUE OF THE STRING IS ZERO. IF E IS THE LAST CHARACTER OF THE STRING, A DELIMITER OTHER THAN A SPACE MUST BE USED, AND THE POWER OF 10 IS ASSUMED TO BE 0. IF THE NUMBER CORRESPONDING TO A LETTER IS GREATER THAN 9, THE TEN'S DIGIT WILL BE CARRIED IF THAT CHARACTER IS PART OF A STRING. A COMPLETE LIST OF THE NUMERIC VALUES OF THE CHARACTERS IS GIVEN IN THE APPENDICES.

12. OPTIONS IN BOTH THE ASK AND TYPE COMMANDS

QUOTE

A CHARACTER STRING ENCLOSED IN QUOTATION MARKS WILL BE REPRODUCED VERBATIM WHEN THAT STATEMENT IS EXECUTED. ^G (BELL) WILL RING THE BELL.

EXCLAMATION POINT

AN EXCLAMATION POINT AS AN ARGUMENT OF A TYPE OR ASK STATEMENT WILL CAUSE FOCARL TO GENERATE A CARRIAGE RETURN/LINE FEED COMBINATION.

NUMBER SIGN (#)

AS AN ARGUMENT OF AN I/O COMMAND WILL GENERATE A CARRIAGE RETURN WITHOUT A LINE FEED.

DOLLAR SIGN ($)

THE $ OPTION WILL GENERATE A PRINT-OUT OF THE SYMBOL TABLE, A LIST OF ALL OF THE DEFINED VARIABLES AND THEIR CURRENT VALUES.
IT MAY BE USED IN AN INDIRECTLY PROGRAMMED STATEMENT; HOWEVER, THERE IS USUALLY LITTLE REASON TO DO SO.
F IF I/O WAS TO DISK (DATA FILE) IT IS CHANGED TO TELETYPE.

QUESTION MARK (?)

A VARIABLE NAME ENCLOSED IN ?'S IN AN I/O STATEMENT WILL BE TYPED OUT. THE ASK STATEMENT WILL TYPE A COLON AFTER THE VARIABLE NAME, AND THEN WAIT FOR THE USER TO TYPE IN THE VALUE; THE TYPE STATEMENT WILL FOLLOW THE VARIABLE NAME WITH ITS CURRENT VALUE. MULTIPLE VARIABLES CAN BE INCLUDED IN THE SAME SET OF ?'S, AS LONG AS THEY ARE SEPARATED BY SPACES OR COMMAS. IN THIS INSTANCE, FOCARL WILL TYPE BOTH THE VARIABLE NAME AND THE DELIMITER BEFORE TYPING THE VALUE (TYPE) OR : (ASK).

NOTE: COMMAS SEPARATING QUOTES, !'S AND #'S FROM ONE ANOTHER MAY BE OMITTED.

13. TRACING

FOCARL PROVIDES AN OPTION TO ALLOW THE USER TO TRACE THE EXECUTION OF HIS PROGRAM. WHEN A ? IS ENCOUNTERED, EXCEPT IN A COMMENT LINE OR WITHIN A QUOTE IN AN INPUT/OUTPUT STATEMENT, FOCARL ENABLES THE TRACE FEATURE. TRACE REMAINS ENABLED UNTIL ANOTHER ? IS ENCOUNTERED. EACH "ODD-NUMBERED" ? ENABLES TRACE, WHILE EACH "EVEN-NUMBERED" OCCURRANCE DISABLES TRACE.
WHEN ENABLED, TRACE BEGINS TYPING EVERY CHARACTER THAT FOCARL ENCOUNTERS AS IT EXECUTES THE PROGRAM.
THE ? WILL BE RECOGNIZED WHEREVER IT APPEARS IN A COMMAND STATEMENT. IT CAN BE IMBEDDED IN A LINE NUMBER OR A VARIABLE NAME, OR BEFORE, FOLLOWING, OR EVEN WITHIN A COMMAND.
AN EXAMPLE OF A PROGRAM TRACE IS GIVEN IN THE APPENDICES.

14. COMMENT

THE COMMENT COMMAND IS USED TO INCLUDE COMMENTS IN A PROGRAM LISTING. ONCE A COMMENT COMMAND IS ENCOUNTERED, THE REMAINDER OF THE LINE IS ASSUMED PART OF THE COMMENT. THUS, COMMENT SHOULD BE THE LAST COMMAND TO APPEAR IN A LINE, SINCE COMMANDS FOLLOWING IT ON THE SAME LINE WON'T BE EXECUTED.

15. *C* THE SCOPE AS AN OUTPUT DEVICE

OPTION SCOPE AND ERASE SCOPE COMMANDS

THERE ARE TWO OUTPUT DEVICES AVAILABLE ON THREE OF THE FOUR 8/L'S: THE TELETYPE PRINTER AND CRT DISPLAY ("SCOPE"). OUTPUTTING ONTO THE SCOPE HAS THE ADVANTAGE OF BEING BOTH FASTER AND QUIETER THAN TTY OUTPUT. FOR THESE REASONS, MUCH OF THE PROGRAMMING AND DEBUGGING DONE IN COLPAC IS DONE USING SCOPE OUTPUT.

UPON ENTERING COLPAC, TELETYPE OUTPUT IS ASSUMED. TO TRANSFER OUTPUT TO THE SCOPE, THE COMMAND OPTION SCOPE IS USED. COLPAC THEN TRANSFERS ALL OUTPUT, INCLUDING CHARACTER ECHO, TO THE SCOPE. COLPAC'S ASTERISK IS PRINTED AT THE LEFT MARGIN OF THE SCREEN.

THE ERASE SCOPE COMMAND IS USED TO CLEAR THE DISPLAY FACE AND RESET THE ASTERISK TO THE TOP OF THE SCREEN. THE ERASE SCOPE COMMAND DOES NOT AFFECT THE PROGRAM OR SYMBOL TABLE STORED IN CORE, AND IT MAY BE PROGRAMMED INDIRECTLY.

SCOPE INFORMATION AND SCALE SIZE

THE 8/L SCOPE IS LABELLED WITH A CARTESIAN COORDINATE SYSTEM IN TWO DIMENSIONS. THE X COORDINATE AXIS (HORIZONTAL) IS 600 UNITS LONG, FROM -300 TO 300; THE Y AXIS (VERTICAL) IS 760 UNITS LONG, FROM -380 TO 380. THE CENTER OF THE SCOPE FACE IS THE ORIGIN, WITH COORDINATES (0,0).

THE SCOPE ON THE 8/L'S IS A "WRAP-AROUND" SCOPE, WHICH MEANS THAT A LINE WHICH GOES OFF THE SCOPE ON ONE SIDE WILL RETURN ON THE OTHER SIDE. THE WRAP-AROUND LENGTH IN THE X-DIRECTION IS APPROXIMATELY 720 UNITS; IN THE Y-DIRECTION, ABOUT 630 UNITS. THE POINT (743,25), FOR EXAMPLE, WOULD BE PLOTTED AT APPROXIMATELY (-280,25).

SCALE SIZE

WHEN PRINTING CHARACTERS OR TEXT STRINGS ON THE SCOPE, IT IS SOMETIMES DESIREABLE TO HAVE THEM ENLARGED FOR EASE IN READING. THE & OPTION OF THE TYPE OR ASK COMMAND IS USED TO CHANGE THE SCALE SIZE. THE FORMAT OF THE COMMAND IS SIMPLY:

```
  TYPE &E       E BEING A NONNEGATIVE NUMBER, THE INTEGER PORTION OF
                WHICH IS TAKEN TO BE THE ENLARGEMENT FACTOR.
```

AN ENLARGEMENT FACTOR OF 0 OR 1 DENOTES NORMAL MAGNIFICATION, 2 DENOTES TWICE THE SIZE, 3, TREBLE THE SIZE, ETC. AN ENLARGEMENT FACTOR OF 1024 PRODUCES A ZERO MAGNIFICATION, THE CHARACTERS BEING PRINTED AS A SINGLE POINT; 1023 IS NORMAL MAGNIFICATION, BUT UPSIDE DOWN, 1022 TWICE THE SIZE AND INVERTED, ETC. ONCE A SCALE SIZE HAS BEEN SPECIFIED, IT WILL CONTINUE UNTIL ANOTHER SIZE IS REQUESTED. TELEPRINTER OUTPUT AND COORDINATE PLOTTING ARE NOT AFFECTED BY A CHANGE IN SCALE SIZE. E MAY BE AN ARITHMETIC EXPRESSION, IN WHICH CASE ITS VALUE IS TRUNCATED TO DETERMINE THE MAGNIFICATION.

16. *C* PLOTTING ON THE CRT DISPLAY, THE "SCOPE"

COLPAC ALLOWS THE USER TO PLOT GRAPHS AND DIAGRAMS ON THE SCOPE BY SPECIFYING THE COORDINATES OF THE POINTS TO BE PLOTTED. THE COORDINATE VALUES, DENOTED BELOW AS X,Y AND A,B, MAY BE ARITHMETIC EXPRESSIONS.

TO PLOT A POINT ON THE SCOPE, THE COMMAND PLOT X,Y IS USED, WHERE X IS AN ARITHMETIC EXPRESSION FOR THE X COORDINATE OF THE POINT, AND Y IS AN ARITHMETIC EXPRESSION FOR THE Y COORDINATE OF THE POINT.

TO MAKE THE POINTS PLOTTED EASIER TO SEE, THE USER MAY WANT TO DRAW A CHARACTER, OFTEN AN "*", AT THE POINT PLOTTED. TO DO THIS, THE COMMAND PLOT X,Y,"TEXT" IS USED. THIS COMMAND WILL PRINT THE TEXT STRING ENCLOSED IN QUOTES BEGINNING AT POINT (X,Y) AND EXTENDING TO THE RIGHT, UNLESS THE & OPTION IS USED TO INVERT PRINTING - THEN IT IS WRITTEN TO THE LEFT.

THE COMMAND PLOT *,X,Y ALLOWS THE USER TO DRAW LINE SEGMENTS ON THE SCOPE. IT DRAWS A LINE SEGMENT CONNECTING THE LAST POINT PLOTTED AND THE POINT (X,Y). TO DRAW AN ISOLATED LINE SEGMENT, THE USER WOULD USE THE TWO COMMANDS PLOT A,B; PLOT *,X,Y . THIS WOULD DRAW A LINE SEGMENT FROM POINT (A,B) TO POINT (X,Y). TO DRAW A SERIES OF CONNECTED LINE SEGMENTS, FOR EXAMPLE, IN A LINE GRAPH, THE USER WOULD PLOT THE INITIAL POINT USING A PLOT X,Y COMMAND. EACH SUBSEQUENT POINT WOULD BE PLOTTED USING A PLOT *,X,Y COMMAND.

PLOT %,X,Y IS USED TO DRAW A CIRCLE ON THE SCOPE. THE CIRCLE IS CENTERED AT THE POINT (X,Y) AND THE LAST PREVIOUSLY PLOTTED POINT IS ASSUMED TO BE ON THE CIRCUMFERENCE. THUS, TO DRAW A CIRCLE CENTERED AT THE ORIGIN AND HAVING A RADIUS OF 25, THE USER MIGHT SAY: *PLOT 0,25; PLOT %,0,0

TO DRAW AN ARC, THE COLPAC COMMAND PLOT #,X,Y,D IS USED, WHERE THE POINT (X,Y) IS THE CENTER OF CURVATURE OF THE ARC, AND D DENOTES THE LENGTH OF THE ARC (IN DEGREES). THE ARC IS PLOTTED IN A CLOCKWISE DIRECTION, BEGINNING AT THE LAST PREVIOULY PLOTTED POINT.

ALL OF THE PLOT COMMANDS ALLOW FOR TEXT OUTPUT FOLLOWING THE ARGUMENTS OF THE PLOT COMMAND. CHARACTER STRINGS ENCLOSED IN QUOTATION MARKS WILL BE WRITTEN VERBATIM, BEGINNING AT THE POINT SPECIFIED IN THE PLOT COMMAND. THE VALUES OF VARIABLES MAY BE PRINTED, AND THE ! (CR/LF) AND # (CR) OPTIONS ARE ALSO ALLOWED. NUMERIC FORMATS CANNOT BE CHANGED WITHIN A PLOT STATEMENT.

WHEN TRYING TO PLOT A THREE-DIMENSIONAL OBJECT IN TWO DIMENSIONS, IT IS OFTEN NECESSARY TO HIDE LINES. THIS CAN BE DONE USING THE COMMAND PLOT X,Y, WHERE NO TEXT STRING FOLLOWS THE COMMA AFTER THE Y. COLPAC WILL NOT PLOT ANYTHING AT THE POINT (X,Y), BUT WILL CONSIDER IT TO BE THE LAST POINT PLOTTED. SUBSEQUENT PLOT COMMANDS MAY REFERENCE THIS POINT FOR DRAWING LINE SEGMENTS, CIRCLES, OR ARCS, EVEN THOUGH IT DOESN'T APPEAR ON THE SCOPE FACE.

17. *C* JOYSTICK

ON THE B/L'S WITH CRT DISPLAYS, THE USER WILL FIND AN INSTRUMENT CALLED A JOYSTICK. IT IS A SINGLE, OMNIDIRECTIONAL CONTROL USED TO GUIDE A DOT OF LIGHT, THE CURSOR, ACROSS THE SCOPE FACE. ACROSS THE END OF THE JOYSTICK CONTROL HOUSING IS AN ORANGE HINGED BAR LABELLED "INTERRUPT". THE INTERRUPT BAR MAY BE USED TO ENTER THE COORDINATES OF THE CURSOR AT THE TIME THE BAR IS PRESSED.

THE COMMAND TO ENABLE THE JOYSTICK IS:

JOYSTICK ALPHA,BETA   ALPHA AND BETA BEING ANY VARIABLES

WHEN THE JOYSTICK COMMAND IS EXECUTED, THE CURSOR APPEARS ON THE SCOPE FACE IN A POSITION RELATIVE TO THE POSITION OF THE JOYSTICK. MOVING THE JOYSTICK WILL CAUSE THE CURSOR TO MOVE ACROSS THE FACE OF THE SCOPE. THE JOYSTICK WILL REMAIN ENABLED UNTIL THE INTERRUPT BAR IS PRESSED. PRESSING THE INTERRUPT BAR CAUSES THE COORDINATES OF THE CURSOR TO BE STORED AS THE VARIABLES ALPHA AND BETA: ALPHA STORES THE X-COORDINATE, BETA STORES THE Y-COORDINATE.

THERE ARE TWO OTHER JOYSTICK COMMANDS IN COLPAC:

JOYSTICK *, X,Y   AND JOYSTICK

JOYSTICK *,X,Y DOES NOT WAIT FOR THE INTERRUPT BAR TO BE PRESSED, BUT STORES THE COORDINATES OF THE CURSOR AT THE MOMENT THE COMMAND IS EXECUTED.

JOYSTICK IS A CONDITIONAL BRANCH STATEMENT. IT CHECKS THE INTERRUPT "FLAG" TO SEE IF THE INTERRUPT BAR HAS BEEN PRESSED SINCE THE LAST JOYSTICK COMMAND. IF IT HAS BEEN PRESSED, JOYSTICK CAUSES THE REMAINDER OF THE LINE TO BE EXECUTED. IF IT HAS NOT BEEN DEPRESSED, IT CAUSES EXECUTION TO CONTINUE WITH THE NEXT LINE.

18. GOTO COMMAND

GOTO IS AN UNCONDITIONAL BRANCH STATEMENT. WHENEVER A GOTO AB.CD COMMAND IS ENCOUNTERED, THE PROGRAM CONTINUES EXECUTION BEGINNING AT LINE AB.CD UNTIL THE END OF THE PROGRAM OR UNTIL PROGRAM EXECUTION IS TERMINATED BY ONE OF THE FOLLOWING:

```
    ERROR (ERROR MESSAGE IS TYPED)
*F* LIBRARY CALL
*C* LIBRARY CALL EXECUTED INDIRECTLY IS EXACTLY LIKE A LIBRARY EXTEND
*F* LIBRARY DELETE
    LOGOUT
    QUIT
    RETURN WITHOUT A CORRESPONDING DO COMMAND
    ^C OR ^BS PROGRAM INTERRUPT
    THE COMPUTER "CRASHES"
```

AB.CD MAY BE AN ARITHMETIC EXPRESSION SO LONG AS A LINE WITH THAT NUMBER EXISTS IN THE PROGRAM.

19. IF COMMANDS

THE IF STATEMENT IS CALLED A CONDITIONAL BRANCH STATEMENT. THAT IS, CERTAIN CONDITIONS (VALUES OF VARIABLES) MUST EXIST BEFORE IT WILL TRANSFER CONTROL (BRANCH) TO ANOTHER PART OF THE PROGRAM. THERE ARE TWO TYPES OF IF STATEMENTS IN FOCARL: VALUATIONAL AND RELATIONAL.

VALUATIONAL IF

THE FORMAT OF THE VALUATIONAL IF STATEMENT IS:

```
IF (AR) AB.CD, EF.GH, IJ.KL
```

(AR IS AN ARITHMETIC EXPRESSION, AND AB.CD, EF.GH, AND IJ.KL ARE LINE NUMBERS.) THE IF STATEMENT CHECKS FOR THREE CONDITIONS: <0, =0, >0. IF THE VALUE OF THE ARITHMETIC EXPRESSION IS NEGATIVE, IT BRANCHES TO LINE AB.CD; IF ZERO, TO LINE EF.GH; IF POSITIVE, TO LINE IJ.KL

LESS THAN THREE ARGUMENTS CAN BE GIVEN FOLLOWING AN IF COMMAND.

```
EX:  02.31 IF (A-6) 2.39; TYPE "A=>6"!
     IF A<6, ((A-6)<0), IT BRANCHES TO LINE 2.39; OTHERWISE, THE TYPE
     COMMAND WILL BE EXECUTED.
EX:  04.30 IF (47); TYPE "HOWDY"
     THE TYPE COMMAND WILL ALWAYS BE EXECUTED, SINCE NO BRANCH IS
     INDICATED.
```

RELATIONAL IF

THE FORMAT OF THE RELATIONAL IF STATEMENT IS:

IF (E1@E2);(COMMAND STRING)

E1 AND E2 ARE ARITHMETIC EXPRESSIONS, AND @ DENOTES ONE OF THE FOLLOWING RELATIONS BETWEEN THE VALUES OF THE TWO EXPRESSIONS:

```
< LESS THAN      > GREATER THAN      = EQUAL TO     # NOT EQUAL TO
<= OR =< LESS THAN OR EQUAL TO      <# OR #< LESS THAN OR NOT EQUAL TO
=> OR >= GREATER THAN OR EQUAL TO  #> OR ># GREATER THAN OR NOT EQUAL TO
<> OR >< LESS THAN OR GREATER THAN (I.E., NOT EQUAL TO)
<#> OR <># OR #>< OR >#< OR ><# OR #<> LESS THAN OR GREATER THAN
<=> OR =<> OR <>= OR =>< OR >=< OR ><= LESS THAN OR EQUAL TO OR
GREATER THAN  (I.E., ALWAYS TRUE).
```

THE RELATIONAL IF COMMAND EVALUATES THE TWO EXPRESSIONS AND THEN COMPARES THEM ACCORDING TO THE INDICATED RELATION. IF THE STATEMENT IS TRUE, THE COMMAND STRING FOLLOWING THE SEMICOLON IS EXECUTED. IF THE STATEMENT IS NOT TRUE, EXECUTION CONTINUES WITH THE NEXT LINE OF THE PROGRAM.
A DELIMITER (SEMICOLON OR SPACE) NEED NOT SEPARATE THE OUTSIDE RIGHT PARENTHESIS IN A RELATIONAL IF COMMAND FROM THE COMMAND STRING WHICH FOLLOWS. FOR EXAMPLE:

```
*
*IF (N8<Ø)IF (N8^2=4)TYPE N8,!
 -2.ØØØ
*
*ASK X,X; IF (X=ØHI)TYPE "HOWDY"!
:HI
HOWDY
*
*
```

THE Ø(ALPHABETIC STRING) OPTION OF EVALUATING ALPHABETIC RESPONSES TO AN INPUT REQUEST MAY BE USED IN AN IF STATEMENT. SEE LINE 4.1 OF THE EXAMPLE OF A PROGRAM TRACE IN THE APPENDICES.

2Ø. QUIT COMMAND

THE QUIT COMMAND TERMINATES PROGRAM EXECUTION.

21. DO AND RETURN

THE DO COMMAND IS USED TO SUSPEND SEQUENTIAL EXECUTION OF COMMANDS, EXECUTE INTERVENING COMMANDS, AND THEN CONTINUE EXECUTION AT THE POINT SUSPENDED. THE INTERVENING COMMANDS CAN BE A SINGLE LINE OR AN ENTIRE GROUP AS SPECIFIED BY THE ARGUMENT OF THE DO COMMAND. THE DO COMMAND WITHOUT AN ARGUMENT, USED AS A DIRECT COMMAND, WILL BE INTERPRETED AS A DO ALL COMMAND BY FOCARL. THIS COMMAND CAUSES EXECUTION TO BEGIN AT THE SMALLEST LINE NUMBER JUST LIKE THE GO COMMAND.

THE USER MAY INCLUDE ARITHMETIC EXPRESSIONS AS ARGUMENTS OF DO COMMANDS, SO LONG AS THE LINE OR GROUP NUMBERS REFERENCED EXIST IN THE PROGRAM.

DO N (N HAS A VALUE WHICH CORRESPONDS TO A GROUP NUMBER)
THIS COMMAND WILL BEGIN EXECUTION OF GROUP N AT THE LOWEST LINE NUMBER AND WILL PROCEED THROUGH THE GROUP SEQUENTIALLY, WITH THE FOLLOWING EXCEPTIONS:

1) A BRANCH STATEMENT TO A NON-SEQUENTIAL LINE WITHIN THE GROUP WILL BE EXECUTED NORMALLY, ALTERING SEQUENTIAL EXECUTION.
2) A DO N.AB TO A LINE WITHIN GROUP N WILL BE EXECUTED CORRECTLY.
3) A BRANCH COMMAND TO A LINE OUTSIDE THE GROUP WILL RESULT IN FOCARL EXECUTING ONLY THAT LINE, BEFORE RETURNING TO THE NEXT LINE IN GROUP N AND CONTINUING EXECUTION THERE. HOWEVER, IF THE LINE BRANCHED TO CONTAINS A DO OR A GOTO COMMAND, THAT COMMAND WILL BE EXECUTED BEFORE FOCARL RETURNS TO GROUP N. THIS "CHAINING" EFFECT CAN CONTINUE TO MULTIPLE LEVELS. THE DO WILL EXECUTE ONLY ONE LINE WHEN A BRANCH COMMAND DIRECTS IT OUTSIDE THE GROUP SPECIFIED, BUT THAT LINE WILL BE EXECUTED COMPLETELY BEFORE RETURNING TO THE DO GROUP, GROUP N.

DO L.AB (L.AB IS A LINE NUMBER)
THIS COMMAND CAUSES THE LINE L.AB TO BE EXECUTED. IF LINE L.AB CONTAINS A DO STATEMENT, THIS WILL BE EXECUTED COMPLETELY BEFORE RETURNING TO LINE L.AB. IF LINE L.AB CONTAINS A BRANCH STATEMENT, ONLY THE LINE SPECIFIED BY THE BRANCH COMMAND WILL BE EXECUTED. IF, HOWEVER, THIS LINE CONTAINS A DO OR BRANCH COMMAND, IT WILL ALSO BE EXECUTED BEFORE CONTROL RETURNS TO LINE L.AB.

THE DO COMMAND CAN ALSO BE USED TO DO RECURSION BY INCLUDING A DO N COMMAND INTERNAL TO GROUP N. FOR EXAMPLE, THE PROGRAM LINE:

```
02.20 IF (BC*4-AL) 3.1,2.27;DO 2
```

RETURN

THE RETURN COMMAND IS USED IN CONJUNCTION WITH A DO STATEMENT. A RETURN COMMAND SIGNALS THE END OF EXECUTION OF A DO COMMAND. EXECUTION CONTINUES WITH THE NEXT SEQUENTIAL COMMAND, OR, IF THE DO WAS NESTED INTERNAL TO ANOTHER DO COMMAND OR FOR ITERATION, EXECUTION CONTINUES AS THEY DIRECT.

A RETURN COMMAND, IF ENCOUNTERED WHILE NOT EXECUTING A DO COMMAND, WILL FUNCTION AS A QUIT COMMAND, TERMINATING PROGRAM EXECUTION.

22. FOR AND BREAK

THE FOR COMMAND IS USED TO PERFORM ITERATIONS. THE SYNTAX OF A FOR COMMAND IS:

```
FOR VA=VI,VF,IN; (VI, VF, IN CAN BE ARITHMETIC EXPRESSIONS)
```

WHEN THE FOR COMMAND IS EXECUTED, THE VARIABLE, DENOTED VA, IS SET EQUAL TO THE INITIAL VALUE, VI. WITH VA=VI, THE COMMAND STRING FOLLOWING THE ";" IS EXECUTED. THEN THE INCREMENT, IN, IS ADDED TO VA AND THIS VALUE IS CHECKED AGAINST THE FINAL VALUE, VF.
IF VA<VF AND THE INCREMENT IS NEGATIVE, FOCARL CONTINUES EXECUTION WITH THE NEXT PROGRAM LINE.
IF VA>VF AND THE INCREMENT IS NEGATIVE, THE COMMAND STRING IS EXECUTED AGAIN, THE INCREMENT ADDED TO VA AND ITS VALUE CHECKED AGAIN. THIS CYCLE CONTINUES UNTIL:
VA=VF, WHEN THE COMMAND STRING IS EXECUTED ONCE MORE BEFORE CONTINUING EXECUTION WITH THE NEXT LINE; OR,
VA<VF, WHICH CONTINUES EXECUTION WITH THE LINE FOLLOWING THE FOR COMMAND, UNLESS THE FOR COMMAND WAS NESTED WITHIN ANOTHER FOR STATEMENT.

IF VA<VF AND THE INCREMENT IS POSITIVE, THE COMMAND STRING IS EXECUTED, VA IS INCREMENTED BY IN, AND ITS VALUE AGAIN COMPARED WITH VF. THIS CYCLE CONTINUES UNTIL VA>VF. THEN EXECUTION IS DIRECTED TO THE NEXT SEQUENTIAL LINE, UNLESS THE FOR LOOP JUST COMPLETED WAS NESTED WITHIN ANOTHER FOR LOOP.

WITH NESTED FOR LOOPS, THE COMPLETION OF THE ITERATIONS OF THE INNER FOR LOOP IS THE COMPLETION OF ONLY ONE EXECUTION OF THE COMMAND STRING OF THE OUTER FOR LOOP. THUS, THE INNER FOR STATEMENT WILL BE DIRECTED THROUGH ALL OF ITS ITERATIONS FOR EACH ITERATION OF THE OUTER LOOP. THUS, THE STATEMENT:

```
FOR I=1,4,1;FOR J=1,3,1;TYPE "*"
```

WILL CAUSE THE TYPE COMMAND TO BE EXECTUTED 12 TIMES.
IF NO INCREMENT IS SPECIFIED IN THE FOR STATEMENT, FOCARL ASSUMES AN INCREMENT OF 1. THUS, THE STATEMENT COULD HAVE BEEN WRITTEN:

```
FOR I=1,4;FOR J=1,3;TYPE "*"
```

BREAK

THE BREAK COMMAND IS USED TO EXIT EARLY FROM A FOR LOOP. A BREAK COMMAND ENCOUNTERED WITHIN NESTED FOR LOOPS WILL CAUSE EXIT FROM ONLY THE MOST INTERNAL LOOP. FOR A SAMPLE PROGRAM ILLUSTRATING THE BREAK COMMAND, SEE THE APPENDICES.

23. TEXT EDITING COMMANDS: WRITE, ERASE, MODIFY

WRITE

THE WRITE COMMAND IS USED FOR LISTING OUT THE USER'S PROGRAM. THERE ARE THREE OPTIONS TO THE WRITE COMMAND:

```
WRITE AB.CD              PRINTS OUT LINE AB.CD (IF IT EXISTS)
WRITE B                  PRINTS OUT ALL OF GROUP B (IF IT EXISTS)
WRITE  OR WRITE ALL      LISTS OUT THE ENTIRE PROGRAM
```

THE WRITE COMMAND CAN BE USED EITHER AS A DIRECT OR AN INDIRECT COMMAND, BUT THERE IS LITTLE REASON TO USE IT AS AN INDIRECT COMMAND.
NO ERROR OCCURS IF THE LINE OR GROUP IS NOT FOUND (I.E., DOES NOT EXIST).

ERASE

THE ERASE COMMAND IS USED TO DELETE INFORMATION.
THERE ARE FIVE OPTIONS TO THE ERASE COMMAND:

```
    ERASE AB.CD DELETES LINE AB.CD FROM YOUR PROGRAM (IF IT EXISTED)
    ERASE B     DELETES ALL OF GROUP B (IF IT EXISTED)
    ERASE       ERASES THE SYMBOL TABLE (I.E., SETS ALL VARIABLES=0)
    ERASE ALL   ERASES ENTIRE PROGRAM, SYMBOL TABLE, AND SCOPE
*C* ERASE SCOPE ERASES SCOPE -- IGNORED BY FOCARL
```

THE OPTIONS, ERASE AND ERASE SCOPE, ARE THE ONLY ONES WHICH CAN BE PROGRAMMED INDIRECTLY. ERASE IS OFTEN INCLUDED AT THE BEGINNING OF A PROGRAM, BEFORE ANY VARIABLES HAVE BEEN DEFINED, TO SET ALL VARIABLES TO ZERO, ELIMINATING ANY "CARRY-OVER" FROM THE PREVIOUS PROGRAM.
NO ERROR OCCURS IF THE LINE OR GROUP SPECIFIED IS NOT FOUND IN ERASE TEXT COMMANDS.

MODIFY

THE MODIFY COMMAND IS USED FOR CORRECTING A PROGRAM LINE WITHOUT MAKING THE USER RETYPE THE ENTIRE LINE. THE MODIFY COMMAND CANNOT BE PROGRAMMED INDIRECTLY, SINCE IT NECESSITATES SUPERVISORY INTERACTION BY THE USER. THE FORMAT OF THE MODIFY STATEMENT IS:

```
MODIFY AB.CD
```

WHERE AB.CD IS THE NUMBER OF THE LINE TO BE MODIFIED. FOCARL WAITS FOR THE USER TO TYPE IN THE CHARACTER HE WANTS TO SEARCH THE LINE FOR. IT WILL TYPE OUT THE LINE UNTIL IT FINDS THE SEARCH CHARACTER. IF IT DOESN'T FIND IT, IT WILL TYPE OUT THE ENTIRE LINE. ONCE IT FINDS THE SEARCH CHARACTER, IT WILL WAIT FOR THE USER TO PERFORM ONE OF THE FOLLOWING OPTIONS:

1) TO TYPE IN NEW TEXT FOLLOWING THE SEARCH CHARACTER
2) HIT THE RUBOUT KEY TO DELETE THE LAST CHARACTER TYPED
3) TYPE BACKRROW (SHIFT O) TO DELETE THAT PART OF THE LINE THAT HAS BEEN TYPED OUT. THIS DOES NOT DELETE THE LINE NUMBER.
4) HIT THE RETURN KEY TO TERMINATE THE LINE AT THE SEARCH CHARACTER, DELETING THE REMAINDER OF THE LINE

5) TYPE FORM-FEED (^L; CTRL L) TO CONTINUE TYPING OUT THE LINE UNTIL ANOTHER OCCURRENCE OF THE SEARCH CHARACTER
6) TYPE BELL (^G) TO CHANGE SEARCH CHARACTER, AND THEN TYPE THE NEW SEARCH CHARACTER
7) TYPE LINE FEED TO SAVE THE REMAINDER OF THE LINE

F THE MODIFY COMMAND RETURNS TEXT I/O TO THE TELETYPE.

24. LIBRARY COMMANDS

DISK STORAGE OF USER PROGRAMS

FOR MOST PURPOSES, DISK STORAGE IS TO BE TREATED AS ONLY TEMPORARY STORAGE. THIS IS A RESULT OF TWO FACTORS: THE LARGE NUMBER OF SYSTEM AND DEPARTMENT FILES WHICH ARE PERMANENTLY STORED ON DISK; AND THE UNPREDICTABILITY OF THE COMPUTER. THE FIRST FACTOR MEANS THAT THERE ARE FEWER AVAILABLE DISK SEGMENTS FOR USER PROGRAMS, WHICH NECESSITATES A PRIORITY RANKING OF USER PROGRAMS STORED ON THE SYSTEM, AND PERIODIC PURGING OF OLD AND LOW PRIORITY FILES.
THE SECOND FACTOR MUST BE CONSIDERED AS ONE OF THE HAZARDS OF USING A COMPUTER. THE DISK, THE FASTEST MEDIUM OF EXTERNAL STORAGE, IS ALSO THE MOST VULNERABLE. INFORMATION STORED ON IT CAN BE LOST IF THE SYSTEM CRASHES. THEREFORE, IF YOU WANT TO SAVE A PROGRAM FOR LATER USE, MAKE A PAPER TAPE COPY OF IT.

FOCARL LIBRARY COMMANDS

THE LIBRARY COMMANDS IN FOCARL ARE USED TO REFERENCE DISK FILES. EACH DISK FILE HAS A NAME AND AN EXTENSION AND RESIDES ON A PARTICULAR SECTION OF DISK, DISTINGUISHED BY AN ACCOUNT NUMBER. THE FILE NAME IS FROM ONE TO SIX ALPHANUMERIC CHARACTERS.
F FOCARL APPENDS AN EXTENSION OF .FRL TO FILES IT CREATES WHEN
F STORING PROGRAMS ON DISK. DATA FILES ARE GIVEN .DAT EXTENSIONS.
C COLPAC APPENDS AN EXTENSION OF .COL TO FILES IT CREATES.

THE SYNTAX OF A LIBRARY COMMAND IS:

```
L <SUB-COMMAND> <FILE NAME> [ACCT #]
```

THE [AND] DESIGNATE OPTIONAL PARAMETERS.
F FOCARL ALLOWS THE USER TO SPECIFY A LINE NUMBER FOLLOWING THE
F ACCOUNT NUMBER. IT MAY BE AN ARITHMETIC EXPRESSION AND IS PRECEDED
F BY AN "L". IT HAS MEANING IN THE LIBRARY EXTEND COMMAND ONLY.

THE SIX AVAILABLE LIBRARY SUB-COMMANDS ARE:
CALL--TO LOAD A PROGRAM STORED ON DISK INTO THE USER'S CORE
DELETE--TO DELETE A PROGRAM STORED ON DISK
SAVE--TO STORE A COPY OF A USER'S PROGRAM ON THE DISK
XTEND--TO CHAIN FROM A PROGRAM IN CORE TO A PROGRAM STORED ON DISK
F OUTPUT--TO CREATE A DATA FILE ON DISK OR OPEN AN ALREADY EXISTING
F DATA FILE FOR OUTPUT.
F INPUT--TO OPEN A DATA FILE TO READ THE INFORMATION BACK INTO CORE.

F LIBRARY OUTPUT AND INPUT WILL BE EXPLAINED MORE FULLY IN THE
F SECTION: DATA FILES IN FOCARL.

LIBRARY SAVE   L S <NAME>

THIS COMMAND STORES A COPY OF THE PROGRAM PRESENTLY IN THE USER'S CORE IN A DISK FILE WITH THE GIVEN NAME. THIS FILE IS STORED ON THE LIBRARY OF THE ACCOUNT THAT THE USER IS LOGGED IN UNDER.
F A DOLLAR SIGN FOLLOWING THE NAME TELLS FOCARL TO DELETE ANY FILE
F HAVING THE NAME SPECIFIED IF IT ALREADY EXISTS ON THE CURRENT ACCOUNT
F BEFORE SAVING THE PROGRAM. THIS IS SIMILAR TO BASIC'S "REPLACE".
F EX: *L S PROGRM$

LIBRARY CALL   L C <NAME> [ACCT #]

THE CALL COMMAND IS USED TO COPY A PROGRAM STORED ON DISK INTO THE USER'S CORE.
C COLPAC LOOKS FOR A FILE OF THE GIVEN NAME, AND CHECKS THAT IS
C EITHER A COLPAC OR A FOCARL PROGRAM FILE. IT THEN ERASES THE
C PROGRAM THAT IS IN THE S/L AND READS IN THE PROGRAM FROM DISK.
C PROGRAMMED INDIRECTLY, LIBRARY CALL ACTS EXACTLY LIKE A LIBRARY
C EXTEND.
F FOCARL CAN ONLY READ IN FOCARL PROGRAM FILES. IT ERASES THE PROGRAM
F THAT IS IN CORE AND THE VARIABLES BEFORE DOING SO.

LIBRARY CALL CAN ALSO BE USED TO CALL A PROGRAM FROM THE PSEUDO LIBRARY, ACCOUNT 3, OR FROM AN ACCOUNT OTHER THAN THE ACCOUNT THE USER IS CURRENTLY LOGGED IN UNDER. AN ASTERISK FOLLOWING THE FILE NAME IS USED TO READ A PROGRAM FROM THE PSEUDO LIBRARY.

EX: *L C PROGRM*

TO CALL A PROGRAM THAT IS STORED UNDER ANOTHER ACCOUNT NUMBER, FOLLOW THE NAME OF THE FILE BY A SPACE AND THE ACCOUNT NUMBER IT IS STORED UNDER. EXAMPLE: *L C SNOBAL 301

LIBRARY DELETE L D <NAME>

THIS COMMAND DELETES THE DISK FILE WITH THE GIVEN NAME FROM THE LIBRARY OF THE ACCOUNT THE USER IS LOGGED IN UNDER.
F PROGRAM EXECUTION HALTS UPON COMPLETION OF THIS COMMAND.

LIBRARY EXTEND L X <NAME> [ACCT #] [LAE] (AE IS AN ARITHMETIC EXPRESSION)

THE LIBRARY EXTEND COMMAND IS USED IN "CHAINING" FROM ONE FOCARL PROGRAM TO ANOTHER. BECAUSE OF THE LIMITED CORE AVAILABLE FOR PROGRAM AND VARIABLE STORAGE, THE USER MAY BE UNABLE TO EXECUTE HIS ENTIRE PROGRAM AS ONE IF IT USES A LARGE NUMBER OF VARIABLES. THE LIBRARY EXTEND COMMAND, INCLUDED AS A LINE IN THE PROGRAM, PERFORMS THREE OPERATIONS WHEN IT IS EXECUTED:

```
     1) IT RETAINS THE SYMBOL TABLE, BUT ERASES THE PROGRAM IN CORE
     2) IT LOADS THE PROGRAM SPECIFIED FROM DISK INTO THE USER'S CORE
*C*  3) IT BEGINS EXECUTION AT THE LOWEST LINE NUMBER.
*C*     EXECUTED DIRECTLY, LIBRARY EXTEND IS EXACTLY LIKE LIBRARY CALL.
*F*  3) IT BEGINS EXECUTION AT THE LINE NUMBER SPECIFIED.  IF NO LINE
*F*     NUMBER IS SPECIFIED, EXECUTION STARTS AT THE LOWEST LINE NUMBER
*F*     EXAMPLE OF LIBRARY EXTEND WITH A LINE NUMBER SPECIFIED:
*F*     *L X PROG L30.22
```

IF THE PROGRAM TO WHICH ONE IS EXTENDING IS TOO LARGE TO FIT IN CORE SIMULTANEOUSLY WITH ALL THE VARIABLES EXISTING AT THAT TIME, PROGRAM EXECUTION HALTS.

THE EXTEND COMMAND CAN ALSO BE USED TO REFERENCE PROGRAMS ON OTHER ACCOUNTS BY FOLLOWING THE NAME OF THE FILE BY THE ACCOUNT NUMBER OR BY AN ASTERISK. EXAMPLE:

```
04.05 L X BTLSH2 301
```

THE FOLLOWING EXAMPLES ILLUSTRATE THE USE OF THE LIBRARY COMMANDS.

```
*
*L C PART1
*WRITE
C FOCARL-14

01.10 TYPE "WE WILL BEGIN HERE!",!
01.20 ASK ?A B C?,!
01.30 TYPE "NOW WE WILL XTEND!"!!!
01.40 L X PART2
01.50 TYPE "HERE WE COME BACK TO THE FIRST PART TO QUIT!",!
01.60 QUIT
*L C PART2
*WRITE
C FOCARL-14

03.20 TYPE "THIS IS THE PROGRAM WE XTEND TO.",!
03.30 TYPE "SECOND LINE OF XTEND PROGRAM!",!
03.40 TYPE "A IS ",A,"  B IS ",B,"  C IS ",C,!
03.50 L X PART1 L1.5
*L C PART1
*GO
WE WILL BEGIN HERE!
A :-2.5 B :13.74 C:.089
NOW WE WILL XTEND!


THIS IS THE PROGRAM WE XTEND TO.
SECOND LINE OF XTEND PROGRAM!
A IS -2.50000  B IS  13.7400  C IS  0.08900
HERE WE COME BACK TO THE FIRST PART TO QUIT!
*
```

25. LOGOUT COMMAND

THE LOGOUT COMMAND IS USED TO EXIT FROM THE FOCARL LANGUAGE AND TO RETURN TO TSS/8 MONITOR.

```
*C* YOU CAN NOW RUN ANY PROGRAM WHICH DOES NOT LOAD INTO THE 8/L, OR
*C* EVEN LOGOUT THE 8/L TERMINAL FROM THE TIMESHARING SYSTEM, AND RETURN
*C* TO COLPAC (AFTER LOGGING BACK IN IF YOU LOGGED OUT OF THE SYSTEM)
*C* BY TYPING "R COLPAC:R" AND RETURN AFTER MONITOR'S "."  YOUR PROGRAM
*C* WILL STILL BE INTACT AND YOUR VARIABLES WILL BE UNCHANGED FROM
*C* THE TIME WHEN YOU LOGGED OUT OF COLPAC.  IF OUTPUT WAS TO THE SCOPE
*C* WHEN YOU LOGGED OUT IT WILL STILL BE TO THE SCOPE.
*F* FOCARL USERS CAN RE-ENTER FOCARL BY SIMPLY TYPING "START" AND RETURN
*F* AFTER MONITOR'S "."  YOUR PROGRAM, VARIABLES AND I/O STATUS WILL
*F* BE THE SAME AS WHEN YOU LOGGED OUT.  NOTE:  THIS WILL NOT WORK IF
*F* YOU RUN ANOTHER PROGRAM BETWEEN LOGGING OUT OF FOCARL AND TYPING
*F* "START" BECAUSE YOUR CORE CONTENTS WOULD BE DESTROYED.  ONLY A FEW
*F* MONITOR COMMANDS SUCH AS "TIME", "TALK", AND "USER" CAN BE EXECUTED
*F* WHICH WILL NOT AFFECT YOUR CORE CONTENTS.
```

26. *F* DISK DATA FILES IN FOCARL

IN ADDITION TO STORING FOCARL PROGRAMS ON THE DISK, IT IS ALSO POSSIBLE, USING FOCARL, TO CREATE AND ACCESS DATA FILES ON DISK. THE FILES ARE SET UP AS ONE-DIMENSIONAL ARRAYS OF NUMBERS, WITH THE NUMBERS REFERENCED VIA THEIR LOCATION IN THE FILE.

OPENING A DATA FILE FOR OUTPUT--LIBRARY OUTPUT

THE LIBRARY OUTPUT COMMAND IS USED TO CREATE A DATA FILE OR TO OPEN AN ALREADY EXISTING DATA FILE IF THE NAME IS FOLLOWED BY A DOLLAR SIGN.
THE FORMAT OF THE COMMAND IS SIMPLY: L O <FILE NAME>
IF THERE ARE NO DISK SEGMENTS AVAILABLE FOR A NEW DISK FILE AN ERROR WILL RESULT. IF THERE IS A FILE WITH THE SPECIFIED NAME ALREADY IN EXISTENCE AND A DOLLAR SIGN DOES NOT FOLLOW THE NAME AN ERROR WILL ALSO RESULT.

WRITING ON AN OUTPUT FILE

ONCE AN OUTPUT FILE HAS BEEN INITIALIZED, IT CAN BE USED FOR STORING DATA. DATA IS OUTPUT TO THE DISK BY USING THE %L FORMAT OPTION OF THE TYPE COMMAND. THE FORMAT OF THE COMMANDS ARE:

```
  TYPE %L<ALPHA>,<BETA>,<THETA>,...
```

WHERE <ALPHA> SPECIFIES THE STARTING LOCATION (ON DISK) FOR STORAGE OF THE OUTPUT VALUES <BETA>,<THETA>, AND SO ON

```
  TYPE %L,<BETA>,<GAMMA>,...
```

WHERE <BETA> IS WRITTEN IN THE LOCATION FOLLOWING THE LAST NUMBER OUTPUT TO DISK, <GAMMA> IS STORED IN THE LOCATION FOLLOWING <BETA>, AND SO ON.

OPENING A DATA FILE FOR INPUT--LIBRARY INPUT

BEFORE ONE MAY RECALL INFORMATION FROM A DATA FILE, IT MUST BE OPENED AS AN INPUT FILE. THE LIBRARY INPUT COMMAND (L I <NAME> [ACCT #]) IS USED TO INITIALIZE AN EXISTING DATA FILE AS AN INPUT FILE. IF THE FILE DOES NOT EXIST, FOCARL WILL RETURN AN ERROR MESSAGE. THE INPUT ACCESS POINTER IS RESET TO ZERO (THE FIRST LOCATION IN THE DATA FILE).

READING FROM AN INPUT FILE

ONCE A DATA FILE HAS BEEN INITIALIZED, INFORMATION MAY BE READ FROM IT BY SPECIFYING THE %L FORMAT IN AN ASK STATEMENT. ANALAGOUS TO THE TYPE COMMANDS:

```
  ASK %L<ALPHA, <BETA>, <THETA>, ...
```

WILL READ VALUES STARTING FROM LOCATION <ALPHA> OF THE DISK FILE AND STORE THEM AS THE VARIABLES <BETA>, <THETA>, AND SO ON.

```
  ASK %L, <GAMMA>, <DELTA>
```

WILL READ IN THE NEXT TWO SEQUENTIAL LOCATIONS AND STORE THEIR VALUES AS THE VARIABLES <GAMMA> AND <DELTA>, RESPECTIVELY.

NOTE: <ALPHA> IS AN ARITHMETIC EXPRESSION. <BETA>, ETC. ARE ARITHMETIC EXPRESSIONS IN THE TYPE STATEMENT, BUT MUST BE VARIABLES IN THE ASK STATEMENT.

NOTES ON USING DATA FILES AND %L FORMAT

SPECIFYING %L IN EITHER A TYPE OR AN ASK STATEMENT SELECTS DISK FOR BOTH INPUT AND OUTPUT. UNLESS AN EXPRESSION FOLLOWS THE "L" IN %L, THE ACCESS POINTERS ARE NOT AFFECTED.
ONCE %L IS SPECIFIED, ALL I/O IS ASSUMED TO BE DISK UNTIL ANOTHER FORMAT IS USED, SUCH AS %, %A.BC, OR %A.
WHEN SPECIFYING A LOCATION IN A DISK I/O STATEMENT, NO COMMA SEPARATES %L AND ITS FIRST ARGUMENT, NOR IS A SPACE REQUIRED BETWEEN THEM.

WHEN INPUTTING FROM AND OUTPUTTING TO THE SAME DATA FILE, EACH COMMAND KEEPS ITS OWN POINTER AS TO THE LOCATION IT WILL NEXT REFERENCE ON DISK.
SPECIFYING %, %A.BC, OR %A IN AN ASK OR TYPE STATEMENT WILL CHANGE I/O FROM DISK DATA FILE I/O TO TTY, PUNCH, LINEPRINTER, READER, OR NON-PRINTING DEPENDING ON THE CURRENT I/O DEVICES SELECTED USING THE OPTION COMMAND.

THE EXAMPLE ON THE NEXT PAGE SHOULD HELP CLARIFY SOME OF THE AMBIGUITIES THAT HAVE ARISEN IN THE PRECEDING EXPLANATION.

```
*
*
*WRITE
C FOCARL-14

01.05 TYPE "THIS IS A DEMONSTRATION PROGRAM, ILLUSTRATING THE",!
01.07 TYPE "USE OF DISK DATA FILES.",!,!
01.09 TYPE "THE PROGRAM STORES THE SQUARES OF THE FIRST 100 P)SITIVE ",!
01.11 TYPE "INTEGERS ON DISK AND RETRIEVES THEM AS REQUESTED"
01.12 TYPE " BY THE USER.",!,!
01.15 L O T1;C THIS CREATES DATA FILE AND INITIALIZES IT FOR OUTPUT
01.18 L I T1;C THIS INITIALIZES FILE FOR INPUT
01.20 FOR I=1,100;TYPE %L,I^2;C WRITES FIRST 100 SQUARES ON DATA FILE
01.30 ASK %,"NUMBER YOU WANT THE SQUARE OF",QU,!
01.32 IF (QU) 1.97,1.97,1.33
01.33 IF (QU-FITR(QU)) 1.95,1.35,1.95
01.35 IF (QU-101) 1.38;TYPE "VALUE TOO LARGE",!;GOTO 1.3
01.38 ASK %LQU,AN;C THIS GETS THE VALUE FROM DISK
01.40 TYPE %5,"THE SQUARE OF ",QU," IS ",AN,!
01.43 ASK "AGAIN? <NO=.5>",AG,!;IF (AG-.5) 1.3,1.9,1.3
01.90 L D T1;C DELETES DATA FILE, SINCE IT IS TRIVIAL, AND HALTS US
01.95 TYPE "VALUE IS NOT AN INTEGER.  TRY AGAIN!",!;GOTO 1.3
01.97 TYPE "PROGRAM WORKS FOR POSITIVE INTEGERS <=100",!;GOTO 1.3
*
*
*
*GO
THIS IS A DEMONSTRATION PROGRAM, ILLUSTRATING THE
USE OF DISK DATA FILES.

THE PROGRAM STORES THE SQUARES OF THE FIRST 100 POSITIVE
INTEGERS ON DISK AND RETRIEVES THEM AS REQUESTED BY THE USER.

NUMBER YOU WANT THE SQUARE OF:33
THE SQUARE OF     33 IS   1089
AGAIN? <NO=.5>:0
NUMBER YOU WANT THE SQUARE OF:99
THE SQUARE OF     99 IS   9801
AGAIN? <NO=.5>:

NUMBER YOU WANT THE SQUARE OF:0
PROGRAM WORKS FOR POSITIVE INTEGERS <=100
NUMBER YOU WANT THE SQUARE OF:33.4
VALUE IS NOT AN INTEGER.  TRY AGAIN!
NUMBER YOU WANT THE SQUARE OF:201
VALUE TOO LARGE
NUMBER YOU WANT THE SQUARE OF:97
THE SQUARE OF     97 IS   9409
AGAIN? <NO=.5>:.5
*
*
```

27. *F* COMPUTED FILE NAMES

COMPUTED FILE NAMES PROVIDE FLEXIBILITY IN FILE REFERENCING. FILE NAMES MAY BE CONSTRUCTED FROM UP TO SIX EXPRESSIONS, OR COMBINATIONS OF CHARACTERS AND EXPRESSIONS.

EXAMPLES:

```
*L C (20)                 -IS THE SAME AS L C T, FILE NAME "T"
*L C (20,21,22)           -IS THE SAME AS L C TUV, FILE NAME "TUV"
*L C (1,2,3,4,5,6)        -FILE NAME "ABCDEF"
*L C (1+1,2+2,3+3)        -FILE NAME "BDF"
*L C (X+4,3*I)            -IF X=1, AND I=2, FILE NAME "EF"
*L C FI(1,2)2B            -FILE NAME "FIAB2B"
*L C FI(0L,0E)*           -FILE NAME "FILE" FROM THE PSEUDO LIBRARY
```

ONE MIGHT ASK THE USER TO GIVE THE NAME OF A DATA FILE.

EX:

```
01.10 ASK %A,"6-CHARACTER FILE NAME?",A,B,C,D,E,G,!
01.20 LIBRARY OUTPUT (A,B,C,D,E,G)S;C USE FILE IF IT ALREADY EXISTS.
01.30 TYPE "DONE"!;QUIT
```

THIS PROGRAM WOULD OPEN THE FILE HAVING THE NAME THE USER SPECIFIES. THE "S" TELLS FOCARL TO USE THE FILE HAVING THE NAME GIVEN IF IT ALREADY EXISTS ON THE SAME ACCOUNT. IF NOT, A FILE BY THAT NAME IS CREATED.

THE EXPRESSION FOR A CHARACTER IS TAKEN TO BE THE DECIMAL VALUE OF THE SIX-BIT CHOPPED ASCII CODE FOR THE CHARACTER. SPACES ARE SQUISHED OUT OF FILE NAMES.
NOTE: *L C (A,B,C,D) WILL CALL THE PROGRAM WITH THE NAME "AB" IF A=1, B=2, AND C=D=32, SINCE 32 IS EQUIVILENT TO A SPACE. THAT IS, SPACES MAY BE USED TO PAD OUT A FILE NAME.
ALL LIBRARY COMMANDS ACCEPT COMPUTED FILE NAMES IN FOCARL.

28. OPTION COMMANDS

OPTIONS: TELETYPE, LINEPRINTER, PUNCH, READER, NON-PRINTING, FORMFEED
SCOPE.

OPTION TELETYPE (O T)

O T IS USED TO DESIGNATE THE TTY AS THE INPUT/OUTPUT DEVICE. THE TTY IS THE OUTPUT DEVICE ASSUMED BY FOCARL UPON ENTRY. O T WILL RESTORE TTY PRINTING FROM ANY OF THE OTHER DEVICE DESIGNATIONS AND FROM THE O N COMMAND.

F NOTE: THE SELECTION OF A DISK DATA FILE FOR I/O IS NOT AFFECTED BY
F   O T NOR ANY OF THE OTHER OPTIONS. THE % OPTION MUST BE USED
F   TO RESTORE NUMERICAL I/O TO THE TTY.

OPTION LINEPRINTER (O L)

C O L IS AVAILABLE IN COLPAC ONLY IF THE USER TYPES "R COLPAC:LPT*"
C WHEN HE INITIALLY RUNS COLPAC. THIS CAUSES AN OVERLAY TO BE LOADED
C INTO THE 8/L ALONG WITH COLPAC WHICH PERMITS USE OF THE VIDEOJET
C LINEPRINTER CONNECTED TO THE COMPUTER CENTER 8/L. THIS FEATURE IS
C ONLY USEFUL IF YOU ARE RUNNING COLPAC AT THAT TERMINAL.
F O L DESIGNATES THE LINEPRINTER AS OUTPUT DEVICE.

OPTION PUNCH (O P)

C O P CAUSES THE CURRENT PROGRAM TO BE PUNCHED ON THE HIGH-SPEED PAPER
C TAPE PUNCH IN THE COMPUTER CENTER. SO, UNLESS YOU PHONE THE
C COMPUTER CENTER AND ASK SOMEONE TO MAKE SURE THE PUNCH IS TURNED ON,
C IT IS BEST NOT TO USE THIS OPTION UNLESS YOU ARE RUNNING AT THE
C COMPUTER CENTER 8/L TERMINAL AND CAN CHECK THE PUNCH YOURSELF.
F O P DESIGNATES THE HIGH SPEED PUNCH AS OUTPUT DEVICE.

OPTION READER (O R)

C O R CAUSES COLPAC TO ERASE THE CURRENT PROGRAM IN CORE AND TO READ
C A NEW PROGRAM FROM THE HIGH-SPEED READER IN THE COMPUTER CENTER.
C AS WHEN USING O P, IT IS BEST TO USE THIS OPTION ONLY FROM THE 8/L
C TERMINAL IN THE COMPUTER CENTER, SINCE ONE'S PAPER TAPE MUST BE
C MOUNTED, ETC.
F O R AS A DIRECT COMMAND WILL READ IN A PROGRAM FROM THE HIGH SPEED
F READER. IT DOES NOT ERASE THE CURRENT PROGRAM FIRST. WHEN THE
F END OF THE PAPER TAPE IS REACHED, INPUT IS RESTORED TO THE TTY.
F AS AN INDIRECT COMMAND, ANY REQUEST FOR DATA BY MEANS OF AN ASK
F STATEMENT (PROVIDED THAT DISK DATA I/O IS NOT SELECTED) WILL
F READ FROM THE HIGH SPEED READER AND THE ASK COMMAND'S ":" WILL
F NOT BE TYPED. BE SURE TO INCLUDE A DELIMITER (SPACE, COMMA, OR RETURN)
F BETWEEN DATA VALUES ON THE PAPER TAPE. IF THE END OF THE TAPE IS
F REACHED WHILE IN INDIRECT MODE AN ERROR USUALLY RESULTS.

OPTION NON-PRINTING (O N)

O N IS USED TO UNDUPLEX THE TTY TO FACILITATE READING IN SLOW-SPEED PAPER TAPES. O T MUST BE USED TO RESTORE ECHOING (DUPLEX).

OPTION FORMFEED (O F)

O F IS USED TO GENERATE LEADER/TRAILER CODE FOR PUNCHING PAPER TAPES. IF THE LINEPRINTER IS DESIGNATED AS OUTPUT DEVICE AT THE TIME, A FORMFEED WILL RESULT.

OPTION SCOPE (O S)

C O S IS USED TO TRANSFER ALL OUTPUT, INCLUDING CHARACTER ECHO, ONTO
C THE SCOPE.
F O S IS IGNORED BY FOCARL.

F NOTE: ASSIGNABLE DEVICES ARE RELEASED WHEN FOCARL TYPES ITS *.

29. USE OF HIGH AND LO-SPEED PUNCH AND READER FOR PROGRAM SAVE AND RESTORE.

HIGH-SPEED PAPER TAPE PUNCH

TO PUNCH A COPY OF YOUR PROGRAM ON THE HIGH-SPEED PUNCH, YOU MUST FIRST BE SURE THAT IT IS TURNED ON. IF YOU ARE IN THE LAB, WALK OVER AND CHECK THAT IT IS TURNED ON, AND THAT THERE IS A FOLD OR TWO OF LEADER TAPE PUNCHED. IF YOU ARE AT A REMOTE TERMINAL, CALL EXT. 413 AND HAVE THE CONSULTANT OR LAB ASSISTANT READY THE PUNCH FOR YOU.
TYPE THE COMMAND O P (OPTION PUNCH).
F THEN TYPE THE COMMAND, WRITE.
IF NO ONE ELSE HAD THE PUNCH ASSIGNED, YOUR PROGRAM SHOULD BEGIN PUNCHING OUT.
F WHEN IT IS DONE TYPE O T (OPTION TELETYPE).
PUNCH A FOLD OR TWO OF TRAILER BEFORE YOU TEAR THE TAPE OFF TO ENSURE THAT YOU HAVE YOUR ENTIRE PROGRAM.
F THIS MAY BE DONE BY TYPING O F (OPTION FORMFEED) ONCE OR TWICE
F BEFORE TYPING O T.
THEN TURN THE PUNCH OFF. IF YOU ARE AT A REMOTE TERMINAL, TELL THE CONSULTANT YOUR NAME AND/OR THE PROGRAM NAME, SO HE CAN LABEL THE TAPE. ALSO INFORM HIM WHERE TO LEAVE THE TAPE SO THAT YOU CAN FIND IT WHEN YOU COME TO PICK IT UP.

```
*F* EX:   *O P;O F;W;O F;O T        THESE COMMANDS WILL PUNCH THE
*F*                                 PROGRAM WITH LEADER/TRAILER.
```

HIGH-SPEED PAPER TAPE READER

MOUNT THE TAPE IN THE READER SO THAT THE FOLLOWING CONDITIONS ARE SATISFIED:

1) THE ARROWS PRINTED ON THE TAPE ARE VISIBLE (ON THE TOP OF THE TAPE)
2) THE ARROWS POINT TO THE LEFT AS YOU FACE THE READER
3) LEADER TAPE, NOT YOUR PROGRAM, IS OVER THE READ HEAD

TYPE THE COMMAND, O R (OPTION READER). THE TAPE SHOULD BEGIN READING. WHEN IT HAS BEEN READ IN COMPLETELY, FOCARL WILL TYPE AN *. PUSH THE WHITE BUTTON ABOVE THE READ HEAD TO RUN THE TRAILER THROUGH THE READER OR LIFT THE METAL PLATE AND REMOVE THE TAPE. IF YOU HAVE ANY DIFFICULTY, ASK THE CONSULTANT OR LAB ASSISTANT FOR HELP.
NOTE ABSOLUTELY DO NOT READ OILED PAPER TAPE IN THE HIGH-SPEED READER!!

LO-SPEED PAPER TAPE PUNCH AND READ

TO PUNCH A SLOW SPEED PAPER TAPE PROGRAM COPY TYPE "O F;W;O F" AFTER FOCARL'S "*", PRESS "ON" ON THE PUNCH, AND TYPE RETURN.
TO READ A SLOW SPEED PAPER TAPE, ERASE ANY OLD PROGRAM BY TYPING "E A" AND RETURN, INSERT THE TAPE IN THE READER, TYPE "O N", AND MOVE THE READER SWITCH TO "START". THE TAPE SHOULD BEGIN READING IN. WHEN IT IS FINISHED, TYPE "O T".
C "O S" MAY BE TYPED INSTEAD OF "O N", AND THE PROGRAM WILL BE LISTED
C ON THE SCOPE AS IT IS READ IN.

30. ERROR MESSAGES, COMPUTED LINE AND GROUP NUMBERS, RANDOM NUMBER GENERATOR INITIALIZATION

ERROR MESSAGES

WHEN FOCARL DETECTS A SYNTAX ERROR IN PROGRAMMING OR IS UNABLE TO EXECUTE A COMMAND, IT WILL TYPE AN ERROR MESSAGE. THE ERROR MESSAGE HAS THE FORM:

```
*C* ERROR (NUMBER) @ (LINE NUMBER)
*F* ?(ERROR NUMBER) @ (LINE NUMBER)
```

IF THE ERROR WAS DETECTED WHILE EXECUTING A PROGRAM, OR SIMPLY:

```
*C* ERROR (NUMBER)
*F* ?(ERROR NUMBER)
```

IF THE ERROR OCCURRED WHILE TYPING OR EXECUTING A DIRECT COMMAND. A LIST OF ERROR DIAGNOSTICS IS GIVEN IN THE APPENDICES.

COMPUTED LINE AND GROUP NUMBERS

COMPUTED LINE AND GROUP NUMBERS (I.E., ARITHMETIC EXPRESSIONS) MAY BE SUBSTITUTED FOR FIXED LINE NUMBERS IN ALL COMMANDS (IF, DO, GOTO, AND WRITE) EXCEPT ERASE AND MODIFY.
EXAMPLE: `GOTO 4*H+FITR(U-.5)+0.2`

F COMPUTED LINE NUMBERS MAY ALSO BE USED IN THE LIBRARY EXTEND COMMAND.

F RANDOM NUMBER GENERATOR INITIALIZATION

F THE RANDOM NUMBER GENERATOR IS "RANDOMIZED" DURING THE INITIAL
F DIALOGUE.

31. COMMAND SUMMARY

COMMAND	OPTION	SYNTAX	ACTION
ASK	%A	A %A,B	ACCEPTS ONLY ONE CHARACTER FOR B, STORES ASCII VALUE OF CHARACTER
F	%LA,C		READS VALUE FROM LOCATION A IN DISK
F			DATA FILE INTO VARIABLE C
	!	A !	GENERATES CR/LF COMBINATION
	#	A #	GENERATES A CARRIAGE RETURN (CR)
	"	A "A IS",A	VERBATIM PRINTING OF CHARACTER STRINGS
	?	A ?A?	TYPES OUT VARIABLE NAME, ASKS FOR VALUE
	$	A $	GENERATES LISTING OF SYMBOL TABLE
C	&	A &AE	SETS THE SCOPE CHARACTER SCALE SIZE
F	&		FOCARL IGNORES THE & OPTION
BREAK		BREAK	CAUSES EARLY EXIT FROM A FOR LOOP
COMMENT		C THIS IS...	DENOTES NON-EXECUTED TEXT
DO		DO	BEGINS EXECUTION AT LOWEST LINE NUMBER
		DO 3	EXECUTES GROUP 3, BEGINNING AT LOWEST LINE NUMBER OF GROUP 3
		DO 1.23	EXECUTES LINE 1.23
ERASE		ERASE	ERASES THE SYMBOL TABLE
		ERASE 3	ERASES GROUP 3
		ERASE 2.51	ERASES LINE 2.51
		ERASE ALL	ERASES ENTIRE PROGRAM AND SYMBOL TABLE
C		ERASE SCOPE	ERASES SCOPE
F		ERASE SCOPE	IS IGNORED BY FOCARL
FOR		FOR I=1,5,1;T "*" FOR J=1,6;T %2,J	PERFORMS ITERATIONS OF COMMAND STRING FOLLOWING ";"
GO		GO	STARTS EXECUTION AT LOWEST LINE NUMBER
GOTO		GOTO 1.03	BEGINS (CONTINUES)EXECUTION AT LINE 1.03
IF		IF (A-6) 1.3,1.5 1.4 N Z P	CONDITIONAL BRANCH STATEMENT: (A-6)<0 BRANCHES TO 1.3 (A-6)=0 BRANCHES TO 1.5 (A-6)>0 BRANCHES TO 1.4
		IF (B-13)4.2,5.1;S B=A	
		IF (AE REL AE);COMMAND STRING	EXECUTES COMMAND STRING IF RELATION IS TRUE. AE IS ANY ARITHMETIC EXPRESSION. REL IS A RELATIONAL OPERATOR OR COMBINATION OF THEM.

```
COMMAND               SYNTAX      ACTION

*C* JOYSTICK     J                     EXECUTES REMAINDER OF LINE IF INTERRUPT
*C*                                    BAR HAS BEEN PRESSED; ELSE, EXECUTION
*C*                                    CONTINUES WITH NEXT LINE
*C*                J X,Y               DISPLAY CURSOR; READ VALUES FOR X AND Y
*C*                                    WHEN INTERRUPT BAR IS PRESSED
*C*                J *,X,Y             READS X AND Y; DOESN'T WAIT FOR INTERRUPT
*C*                                    BAR TO BE PRESSED
*F*                                    FOCARL IGNORES JOYSTICK COMMANDS

LIBRARY CALL     L C TEMP                  LOADS A PROGRAM FROM DISK INTO CORE
                 L C DEMO*                 LOOKS ON PSEUDO LIBRARY FOR FILE
                 L C PRGRAM 1234           LOOKS ON ACCT. 1234 FOR THE FILE
LIBRARY DELETE   L D TEST                  DELETES THE PROGRAM TEST FROM DISK
LIBRARY EXTEND   L X TEMP                  CHAINS FROM PROGRAM IN CORE TO THE
                                           PROGRAM CALLED TEMP ON DISK
*F*              L X TEMP L1.7             EXTEND TO PROGRAM; START AT LINE 1.7
*F*              L X GOODY 5555 L31.95        GET PROGRAM FROM ACCOUNT 5555
*F*                                           AND BEGIN EXECUTION AT LINE 31.95
*F*              L X TRIAL2* L3*X+R*.1        ARITHMETIC EXPRESSION FOR LINENO
*F* LIBRARY INPUT    L I TRY1              INITIALIZES DATA FILE FOR INPUT
*F* LIBRARY OUTPUT   L O TRY1              CREATES A DATA FILE ON DISK AND
*F*                                        INITIALIZES IT FOR OUTPUT
*F*              L O TRY2S                 OPENS AN ALREADY EXISTING FILE FOR
*F*                                        OUTPUT OR CREATES A NEW FILE IF
*F*                                        NONE ALREADY EXISTS.
*F*                                        OR CREATES A NEW FILE
LIBRARY SAVE     L S TEST                  STORES THE PROGRAM IN CORE ON DISK
*F*              L S PROGS                 DELETES OLD FILE IF IT ALREADY
*F*                                        EXISTS BEFORE DOING SAVE
LOGOUT           L OR LOGOUT               EXITS FROM FOCARL, RETURNS USER
                                           TO TSS/8 MONITOR

MODIFY           MODIFY 1.37               MODIFY A PROGRAM LINE

OPTION FORMFEED          OUTPUT LEADER/TRAILER OR FORMFEED ON THE LINEPRINTER
OPTION LINEPRINTER       SELECT LINEPRINTER FOR OUTPUT
OPTION NON-PRINTING      UNDUPLEX TTY
*C* OPTION PUNCH         PUNCH PROGRAM ON HIGH-SPEED PUNCH
*F* OPTION PUNCH         SELECT HIGH SPEED PUNCH FOR OUTPUT
*C* OPTION READER        READ A PROGRAM FROM HIGH-SPEED PUNCH
*F* OPTION READER        READ A PROGRAM OR DATA FROM HIGH-SPEED READER
*C* OPTION SCOPE         SELECT SCOPE FOR OUTPUT
*F* OPTION SCOPE         FOCARL IGNORES O S
OPTION TELETYPE          SELECT TTY FOR I/O

*C* PLOT         P X,Y               PLOT THE POINT HAVING COORDINATES X,Y
*C*              P X,Y,"TEXT"        PRINT TEXT STRING AT POINT X,Y
*C*              P X,Y,              PLOT A HIDDEN LINE
*C*              P *,X,Y             DRAW A LINE SEGMENT
*C*              P %,X,Y             DRAW A CIRCLE
*C*              P #,X,Y,D           DRAW AN ARC WITH (X,Y) AT CENTER, D=DEGREES
*F* PLOT                             IS IGNORED BY FOCARL
```

COMMAND	OPTION	SYNTAX	ACTION
QUIT		QUIT	STOPS PROGRAM EXECUTION
RETURN		RETURN	DENOTES END OF A DO SUBROUTINE
SET		S A=3.4/7.1	ASSIGNS VALUE OF RIGHT SIDE TO VARIABLE
TYPE	%	TYPE %,A	NUMERIC OUTPUT IN EXPONENTIAL FORMAT
	%B	TYPE %4,A	NUMERIC OUTPUT IN INTEGER FORMAT
	%A.BC	T %6.Ø4,A	OUTPUT IN DECIMAL FORMAT
	%A	T %A,L	ASCII CHARACTER OUTPUT
F	%L	T %LA,B	OUTPUT TO A DISK DATA FILE THE VALUE B
F			AT LOCATION A
	!	T !	GENERATES CARRIAGE RETURN/LINE FEED
	#	T #	GENERATES A CARRIAGE RETURN
	$	T $	GENERATES A PRINT-OUT OF THE SYMBOL
			TABLE UNDER THE EXISTING FORMAT
	"	T "HI"	VERBATIM PRINTING OF CHARACTER STRINGS
	?	T ?A?	TYPES OUT VARIABLE NAME, THEN VALUE
C	&	T &AE	SETS THE SCOPE CHARACTER SCALE SIZE
F	&		IGNORED BY FOCARL
WRITE		WRITE ALL	TYPES OUT A LIST OF THE ENTIRE PROGRAM
		WRITE 3	LISTS GROUP 3
		WRITE 2.72	LISTS LINE 2.72

APPENDIX

A. FUNCTIONS

B. ASCII AND NUMERIC CODES

C. EXAMPLE OF A PROGRAM TRACE

D. PROGRAM ILLUSTRATING THE BREAK COMMAND

E. DIAGNOSTICS

APPENDIX A -- FUNCTIONS

IN THE LIST OF FUNCTION FORMS AND COMMENTS WHICH FOLLOWS,
(A) DENOTES A NUMBER, A VARIABLE, OR AN ARITHMETIC EXPRESSION.
E.G., FABS(BC), FSQT(7), FITR(SR*4^3/(L*2.1)), FABS(FCOS(1.57))

PERMANENT FUNCTIONS

SQUARE ROOT	FSQT(A)	A=>0;OTHERWISE, ERROR
ABSOLUTE VALUE	FABS(A)	ABSOLUTE VALUE OF A
SIGN OF THE NUMBER	FSGN(A)	YIELDS +1 IF (A)=>0;-1 IF (A)<0
INTEGER PART	FITR(A)	YIELDS INTEGER PORTION OF NUMBER, NOT GREATEST INTEGER <=(A). FITR(7.4)=7 FITR(-3.5)=-3
RANDOM NUMBER	FRAN(A)	HERE VALUE OF (A) DOES NOT INFLUENCE THE VALUE OF THE FUNCTION CALL. GENERATES A NUMBER BETWEEN 0 AND 1

OPTIONAL FUNCTIONS

EXPUNENTIAL	FEXP(A)	RAISES E (2.71828) TO THE (A) POWER. -616<=(A)<=616 AND (A) MAY BE A NON-INTEGER.
NATURAL LOGARITHM	FLOG(A)	FLOG(2)=.69315 FLOG(2.71828)=1.00
ARCTANGENT	FATN(A)	
COSINE	FCOS(A)	(A) IS ANGLE IN RADIANS
SINE	FSIN(A)	(A) IS ANGLE IN RADIANS

C THESE FUNCTIONS ARE STANDARD IN COLPAC

F RETAINING OPTIONAL FUNCTIONS

F LEGITIMATE RESPONSES TO THE QUESTION, "FUNCTIONS?", ARE:

F AL OR ALL	RETAINS ALL OPTIONAL FUNCTIONS
F N, NONE OR RETURN	RETAINS NO OPTIONAL FUNCTIONS
F FEXP EXP OR E	RETAINS EXPONENTIAL FUNCTION
F FLOG LOG OR L	RETAINS NATURAL LOGARITHM
F FATN ATN OR A	RETAINS ARCTANGENT
F FCOS COS OR C	RETAINS BOTH SINE AND COSINE
F FSIN SIN OR S	RETAINS BOTH SINE AND COSINE
F BACKARROW (SHIFT/O)	DELETES PREVIOUSLY TYPED RESPONSES.

F TO RETAIN SEVERAL OF THE FUNCTIONS, SEPARATE THE NAMES WITH COMMAS.
F (E.G., FUNCTIONS? E,FLOG,ATN (RETURN))
F ALL RESPONSES MUST BE FOLLOWED BY A CARRIAGE RETURN. IF THE
F RESPONSE WAS NOT UNDERSTOOD, FOCARL WILL ASK FUNCTIONS? AGAIN.
F WHEN A CORRECT RESPONSE IS GIVEN, FOCARL TYPES AN *. A BACKARROW
F (SHIFT O) DELETES THE PREVIOUS RESPONSE.

APPENDIX B -- ASCII AND NUMERIC CODES

```
C FOCARL-14

01.10 T "THIS PROGRAM GENERATES A TABLE OF THE NUMERIC CODES OF",!

01.20 T "ALPHABETIC CHARACTERS UNDER THE POSSIBLE FORMATS",!

01.30 T "CHARACTER           %A             ASCII CODE        NUMERIC ",!
01.40 T "                (DECIMAL)           (OCTAL)         (DECIMAL)",!
01.50 F I=161,175;D 2;D 3.05;T !
01.52 F I=176,185;S CO=-176;D 2;D 3.05;D 3.06
01.54 F I=186,192;D 2;D 3.05;T !
01.56 F I=193,218;S CO=-192;D 2;D 3.05;D 3.06
01.58 F I=219,223;D 2;D 3.05;T !
01.60 G 2.25

02.10 C THIS DECODES DECIMAL TO OCTAL <BASE 10 TO BASE 8>
02.11 I (I-197) 2.15,2.13,2.15
02.13 S CO=-197;C THIS SETS A VALUE OF 0 FOR E UNDER NUMERIC
02.15 S HI=FITR(I/64);S H=HI+176
02.17 S MD=FITR((I-(HI*64))/8);S M=MD+176
02.19 S LO=I-(HI*64+MD*8);S L=LO+176
02.21 RETURN
02.25 T "SPACE    ";S I=160;D 2;D 3.09
02.27 T "LINE FEED";S I=138;D 2;D 3.09
02.29 T "RETURN   ";S I=141;D 2;D 3.09
02.31 T "BELL     ";S I=135;D 2;D 3.09
02.33 T "RUBOUT   ";S I=255;D 2;D 3.09
02.35 T "LEADER",!,"TRAILER  ";S I=128;D 2;D 3.09
02.99 QUIT

03.05 T "    ",%A,I,"                ",%3,I,"                 ",%A,H,M,L
03.06 T "                 ",%2,I+CO,!
03.09 T "        ",%3,I,"                ",%A,H,M,L,!
*GO
THIS PROGRAM GENERATES A TABLE OF THE NUMERIC CODES OF
ALPHABETIC CHARACTERS UNDER THE POSSIBLE FORMATS
CHARACTER          %A            ASCII CODE        NUMERIC
               (DECIMAL)          (OCTAL)         (DECIMAL)
    !             161               241
    "             162               242
    #             163               243
    $             164               244
    %             165               245
    &             166               246
    '             167               247
    (             168               250
    )             169               251
    *             170               252
    +             171               253
    ,             172               254
    -             173               255
    .             174               256
    /             175               257
```

0	176	260	0
1	177	261	1
2	178	262	2
3	179	263	3
4	180	264	4
5	181	265	5
6	182	266	6
7	183	267	7
8	184	270	8
9	185	271	9
:	186	272	
;	187	273	
<	188	274	
=	189	275	
>	190	276	
?	191	277	
@	192	300	
A	193	301	1
B	194	302	2
C	195	303	3
D	196	304	4
E	197	305	0
F	198	306	6
G	199	307	7
H	200	310	8
I	201	311	9
J	202	312	10
K	203	313	11
L	204	314	12
M	205	315	13
N	206	316	14
O	207	317	15
P	208	320	16
Q	209	321	17
R	210	322	18
S	211	323	19
T	212	324	20
U	213	325	21
V	214	326	22
W	215	327	23
X	216	330	24
Y	217	331	25
Z	218	332	26
[	219	333	
◆	220	334	
]	221	335	
▲	222	336	
▼	223	337	
SPACE	160	240	
LINE FEED	138	212	
RETURN	141	215	
BELL	135	207	
RUBOUT	255	377	
LEADER			
TRAILER	128	200	

APPENDIX C -- EXAMPLE OF A PROGRAM TRACE

```
C FOCARL-14

Ø1.1Ø C THIS PROGRAM WILL FIND THE ROOTS OF A QUADRATIC EQUATION
Ø1.15 T "TYPE IN THE VALUES OF A,B,AND C, WHERE A IS THE",!
Ø1.16 T "COEFFICIENT OF X^2, B THE COEFFICIENT OF X, AND C",!
Ø1.17 T "IS A CONSTANT.",!,!
Ø1.2Ø A ?A B C ?,!,!
Ø1.23 S RD=B^2-4*A*C
Ø1.25 IF (RD)  3.Ø5,1.3,2.Ø5
Ø1.3Ø T "EQUATION HAS ONLY ONE ROOT: ",%6.Ø4,-(B/2),!,!
Ø1.32 G 4.1

Ø2.Ø5 T "EQUATION HAS TWO REAL ROOTS: "
Ø2.1Ø S SR=FSQT(RD)
Ø2.2Ø T -(B-SR)/2,"  ",-(B+SR)/2,!,!
Ø2.3Ø G 4.1

Ø3.Ø5 T "EQUATION HAS TWO IMAGINARY ROOTS:",!
Ø3.Ø7 S RA= FSQT(-RD)
Ø3.1Ø T -B/2," + ",RA," * I "
Ø3.2Ø S RA=-RA; T "   ",!;D 3.1;T !,!
Ø3.3Ø G 4.1

Ø4.1Ø A "AGAIN <Y OR N> "L,!,!;IF (L-ØN) 1.2,4.2,1.2
Ø4.2Ø QUIT
*
*GO
TYPE IN THE VALUES OF A,B,AND C, WHERE A IS THE
COEFFICIENT OF X^2, B THE COEFFICIENT OF X, AND C
IS A CONSTANT.

A :1 B :2 C :3

EQUATION HAS TWO IMAGINARY ROOTS:
-1.ØØØØØ +  2.82843 * I
-1.ØØØØØ + -2.82843 * I

AGAIN <Y OR N> :Y

A :1 B :2 C :1

EQUATION HAS ONLY ONE ROOT: - 1.ØØØØ

AGAIN <Y OR N> :Y

A :37 B :75.9 C :-236

EQUATION HAS TWO REAL ROOTS:  62.9Ø73  -138.8Ø7

AGAIN <Y OR N> :N

*
```

```
*
*GO ?

C C T "TYPE IN THE VALUES OF A,B,AND C, WHERE A IS THE!

T "COEFFICIENT OF X^2, B THE COEFFICIENT OF X, AND C!

T "IS A CONSTANT.!
,!

A :1 :2 :1 ,!
,!

S RD=B^2-4*A*C
IF (RD)  3.05,1.3,T "EQUATION HAS ONLY ONE ROOT: %6.04,-(B/2),- 1.0000!
,!

G 4.1
A "AGAIN <Y OR N> ,:Y !
,!
;IF (L-0N) 1.2,4.2,1.2
A :1 :2 :3 ,!
,!

S RD=B^2-4*A*C
IF (RD)  3.05,T "EQUATION HAS TWO IMAGINARY ROOTS:!

S RA= FSQT(-RD)
T -B/2,- 1.0000" + RA,  2.8284" * I S RA=-RA; T "    !
;D 3.1;T -B/2,- 1.0000" + RA,- 2.8284" * I T !
,!

G 4.1
A "AGAIN <Y OR N> ,:N !
,!
;IF (L-0N) 1.2,4.2,QUIT
*
```

APPENDIX D -- PROGRAM ILLUSTRATING THE BREAK COMMAND

COLPAC, 1970

```
01.10 C THIS IS A PROGRAM TO ILLUSTRATE THE BREAK COMMAND IN COLPAC
01.15 E S
01.20 FOR A=2,5;T %2,!;DO 2
01.30 TYPE &3,!,"THAT'S ALL FOLKS",!;F I=10,17;DO 1.6
01.40 T &;Q
01.60 D 1.9;T &I;P 0,-320,"<";P 0,-320,">"
01.90 F J=1,8;S W=0

02.10 T &2,"FIRST LOOP  A = ",A,!
02.20 FOR B=5,11,2;DO 3

03.20 IF (2*A+B>15);BREAK;GOTO 5.1
03.30 T "  SECOND LOOP  B = ",B,!
03.35 IF (B=11);T " INNER LOOP COMPLETED",!
03.40 RETURN

05.10 TYPE &2,"*****BREAK FROM INNER LOOP!*****",!


*
*
*GO

FIRST LOOP  A =    2
  SECOND LOOP  B =    5
  SECOND LOOP  B =    7
  SECOND LOOP  B =    9
  SECOND LOOP  B =   11
 INNER LOOP COMPLETED

FIRST LOOP  A =    3
  SECOND LOOP  B =    5
  SECOND LOOP  B =    7
  SECOND LOOP  B =    9
*****BREAK FROM INNER LOOP!*****

FIRST LOOP  A =    4
  SECOND LOOP  B =    5
  SECOND LOOP  B =    7
*****BREAK FROM INNER LOOP!*****

FIRST LOOP  A =    5
  SECOND LOOP  B =    5
*****BREAK FROM INNER LOOP!*****

THAT'S ALL FOLKS
*
```

APPENDIX E -- DIAGNOSTICS

ERROR MESSAGES FOR COLPAC, 1970

ERROR 00	MANUAL START
ERROR 01	CONTROL/C RECOVERY OR LINE PRINTER ERROR
ERROR 02	DIRECT COMMAND OR TEXT LINE TOO LONG
ERROR 03	TOO MUCH TEXT--VARIABLES ERASED IF NOT 12K
ERROR 04	TOO MANY VARIABLES
ERROR 05	PUSH-DOWN OVERFLOW IN 'FOR' OR 'IF'
ERROR 06	PUSH-DOWN LIST OVERFLOW
ERROR 07	TELETYPE INPUT BUFFER OVERFLOW
ERROR 08	INVALID COMMAND
ERROR 09	ILLEGAL LINE NUMBER ON TYPED TEXT LINE
ERROR 10	ILLEGAL LINE NUMBER OR FORMAT
ERROR 11	ILLEGAL CHARACTER IN TEXT
ERROR 12	ILLEGAL SCALE
ERROR 13	LEFT OF "=" OR MISSING "=" IN 'FOR' OR 'SET'
ERROR 14	ERROR IN 'FOR'
ERROR 15	BAD ARGUMENT IN 'IF'
ERROR 16	ILLEGAL ARGUMENT IN 'ERASE'
ERROR 17	'ERASE' NOT DIRECT COMMAND
ERROR 18	'MODIFY' DID NOT FIND THE SPECIFIED LINE
ERROR 19	'MODIFY' NOT DIRECT COMMAND
ERROR 20	'GOTO' OR 'GO' DID NOT FIND THE SPECIFIED LINE
ERROR 21	'DO' DID NOT FIND THE SPECIFIED LINE OR GROUP
ERROR 22	ILLEGAL OPTION IN 'OPTION'
ERROR 23	MORE 'BREAK' COMMANDS THAN 'FOR' COMMANDS
ERROR 24	NO ARGUMENT
ERROR 25	NISSING OPERATOR
ERROR 26	EXCESS LEFT PARENTHESES
ERROR 27	MISSING LEFT PARENTHESES
ERROR 28	MIS-MATCHED PARENTHESES
ERROR 29	ILLEGAL VARIABLE NAME
ERROR 30	ILLEGAL SUBSCRIPT
ERROR 31	NEGATIVE OR ZERO ARGUMENT TO 'FLOG'
ERROR 32	DIVISION BY ZERO
ERROR 33	NEGATIVE NUMBER TO 'FSQT'
ERROR 34	EXPONENT TOO LARGE
ERROR 35	NUMERIC OVERFLOW OR UNDERFLOW
ERROR 36	LITERAL NUMBER TOO LARGE
ERROR 37	HIGH SPEED READER OUT OF TAPE, OR COLPAC MONITOR HALTED IN THE 8/I
ERROR 38	BAD FORMAT IN 'PLOT' OR 'JOYSTICK'
ERROR 39	NO DEGREES FOR ARC 'PLOT'
ERROR 40	ILLEGAL LIBRARY COMMAND: FILE NOT FOUND FOR CALL OR DELETE; FILE ALREADY EXISTS FOR SAVE; NOT A COLPAC OR FOCARL FILE; OR DECTAPE DRIVE, PUNCH OR READER NOT AVAILABLE (CAN'T BE ASSIGNED)

ERROR DIAGNOSTICS FOR FOCARL-VERSION 14

CODE	MEANING
?00.00	CONTROL/C RECOVERY
?01.49	ILLEGAL LINE NUMBER USED
?01.93	NONEXISTENT GROUP REFERENCED BY "DO"
?01./3	NONEXISTENT LINE NUMBER REFERENCED BY "DO"
?02.42	BAD LINE NUMBER
?02.96	STORAGE WAS FILLED BY PUSH-DOWN LIST
?03.18	NONEXISTENT LINE USED AFTER "GOTO" OR "IF"
?03.43	ILLEGAL COMMAND USED
?05.<7	ILLEGAL USE OF FUNCTION OR NUMBER
?06.03	BAD ARGUMENT IN "FOR", "SET", OR "ASK"
?06.28	BAD SUBSCRIPT
?06.82	STORAGE IS FILLED BY VARIABLES
?07.10	MISSING OPERATOR IN EXPRESSION
?07.22	EMPTY PARENTHESES
?07.28	NO OPERATOR IS USED BEFORE PARENTHESIS
?07.99	NO ARGUMENT GIVEN AFTER FUNCTION CALL
?07.16	ILLEGAL FUNCTION NAME, DOUBLE OPERATORS, EXCESS PARENTHESES, OR EMPTY PARENTHESES
?08.60	PARENTHESES DO NOT MATCH
?10.15	INVALID CHARACTER TYPED
?10.33	STORAGE FILLED BY TEXT
?11./6	LINEPRINTER OR PUNCH IS HUNG OR DEVICE NOT AVAILABLE. (I.E., CAN'T ASSIGN DEVICE)
?11.<4	END OF TAPE ON H.S. READER
?12.05	IMAGINARY SQUARE ROOT REQUESTED
?16.96	INITIAL DIALOGUE ERROR
?20.42	NEGATIVE OR ZERO ARGUMENT FOR LOG
?21.42	NEGATIVE OR ZERO ARGUMENT FOR LOG
?22.78	TOO MANY "BREAKS"
?22.18	TOO MANY NESTED "FOR" LOOPS
?23.37	LITERAL NUMBER IS TOO LARGE
?24.04	LIBRARY PROGRAM IS TOO LARGE OR ILLEGAL OPTION
?24.07	BAD ARGUMENT IN "IF"
?24.10	BAD ARGUMENT IN "ERASE"
?24.30	LIBRARY EXTEND ERROR: TOO MANY VARIABLES
?24.49	LINE NOT FOUND IN "MODIFY"
?24.87	ILLEGAL USE OF "MODIFY" OR "ERASE" -MUST BE A DIRECT COMMAND
?24.12	ERROR LEFT OF "=" OR MISSING "=" IN "FOR" OR "SET"
?24.13	CANNOT SAVE PROGRAM: DISK FULL!
?24./5	EXCESS RIGHT TERMINATOR IN "FOR" OR "SET"
?24.<7	ILLEGAL TERMINATOR IN "FOR"
?25.14	ILLEGAL FILE NAME OR TOO MANY CHARACTERS
?25.72	BAD ACCOUNT NUMBER FOR FILE
?25.93	ILLEGAL LIBRARY COMMAND
?25.17	DISC FULL OR PROTECTION ERROR ON SAVE OR DATA OUTPUT
?25.<4	FILE NOT FOUND FOR CALL, DELETE, OR DATA INPUT OR PROTECTION VIOLATION

```
?26.03  NOT FOCARL FILE (.FRL) OR DATA FILE (.DAT)
?26.16  FILE ALREADY IN EXISTENCE FOR PROGRAM SAVE OR DATA OUTPUT
?26.26  BAD DISK FILE ACCESS NUMBER.  IT MUST BE NON-NEGATIVE
        AND REASONABLY SMALL.
?26.57  CANNOT DELETE DISK FILE: PROTECTED
?26.72  CANNOT EXTEND OUTPUT FILE: PROTECTED OR DISK FULL
?26.80  INPUT OR OUTPUT DATA FILE NOT INITIALIZED (OPENED)
?26.<5  UNRECOVERABLE DISK ERROR!
?28.!0  EXPONENT TOO LARGE
?30.51  DIVISION BY ZERO REQUESTED
?31./0  NUMERIC OVERFLOW OR UNDERFLOW
?31.<7  UNAVAILABLE FUNCTION USED
```