



# DECUS

## PROGRAM LIBRARY

DECUS NO.	FOCAL8-177
TITLE	PS/8 FOCAL, 1971
AUTHOR	David Schneider and Barry Smith
COMPANY	Submitted by: Hartwell H. Whitney, Jr. Oregon Museum of Science and Industry Portland, Oregon
DATE	August 23, 1971
SOURCE LANGUAGE	PAL-8



**PS/8 FOCAL, 1971**

---

CONTENTS	PAGE	CONTENTS	PAGE
INTRODUCTION.....	1	QUOTATION MARKS..["]....	14
CHANGES FROM FOCAL-69.....	2	TRACE.....[?]....	15
COMMANDS.....	3	POWERS-OF-TEN...[E]....	15
ELEMENTARY COMMANDS.....	4	RUBOUT KEY.....	15
ASK.....	4	CONTROL/C, G, L, P, AND Z...15	
COMMENT.....	4	LINE FEED KEY.....	16
DO.....	4	KILL LINE.....[←]....	16
ERASE.....	4	RETURN KEY.....	16
FOR.....	5	ALT MODE.....	16
GOTO.....	5	SPACE BAR.....[ ]....	16
IF.....	5	COMMA.....[, ]....	16
MODIFY.....	5	SEMICOLON.....[;]....	16
QUIT.....	5	FUNCTIONS.....	17
RETURN.....	6	FABS ABSOLUTE VALUE....	17
SET.....	6	FADC ANALOG-DIGITAL....	17
TYPE.....	6	FATN ARCTAN (RADIANS)..	17
WRITE.....	6	FCOS COSINE (RADIANS)..	17
LIBRARY COMMANDS.....	7	FDIS DISPLAY (340)....	17
PS/8 DEVICE NAMES.....	7	FEXP EXPONENTIAL.....	17
LIBRARY CALL.....	7	FITR INTEGER.....	18
LIBRARY DELETE.....	7	FLOG LOG (BASE E)....	18
LIBRARY GOSUB.....	7	FRAN RANDOM NUMBER....	18
LIBRARY LIST.....	8	FSGN SIGN PART.....	18
LIBRARY RUN.....	8	FSIN SINE (RADIANS)....	18
LIBRARY SAVE.....	8	FSQT SQUARE ROOT.....	18
LIBRARY EXIT.....	8	FIN CHARACTER INPUT...18	
FILE COMMANDS.....	9	FOUT CHARACTER OUTPUT..18	
OPEN INPUT.....	10	FDIS DISPLAY (T-4002)..19	
OPEN OUTPUT.....	10	FJOY JOYSTICK (T-4002).19	
OPEN RESTORE INPUT.....10		MISCELLANEOUS.....	20
OPEN RESTORE OUTPUT....10		HIGH SPEED READER/PUNCH.20	
OUTPUT CLOSE.....	10	LINE/GROUP NUMBERS.....20	
SAMPLE FILES PROGRAMS..11		MERGING PROGRAMS.....20	
ARITHMETIC SYMBOLS.....	13	PRECISION.....	21
EXPONENTIATION...[^]....	13	PIP.....	21
MULTIPLICATION...[*]....	13	PROGRAM NAMES & SIZE...21	
DIVISION.....[/]....	13	PUSHDOWN LIST OVERFLOW..21	
ADDITION.....[+]....	13	STRINGS.....	21
SUBTRACTION.....[-]....	13	VARIABLES/SUBSCRIPTING..22	
ENCLOSURES...(<)[ ]<>....13		INITIAL DIALOGUE.....22	
SPECIAL CHARACTERS.....	14	COMMAND EXECUTION TIMES.23	
RETURN/LINE FEED.[!]....	14	TECHNICAL NOTES.....	24
CARRIAGE RETURN.[#]....	14	INDEX FOR 'HELP'.....	27
TAB.....[:]....	14	DECIMAL ASCII CODES.....	28
OUTPUT FORMATTER.[%]....	14	ERROR CODES.....	29
SYMBOL TABLE....[\$]....	14	COMMAND SUMMARY.....	30
		DECTAPE DIRECTORY.....	31



## INTRODUCTION

PS/8 FOCAL, 1971 MAKES THE DEVICE-INDEPENDENT FILE HANDLING FACILITIES OF PS/8 DIRECTLY AVAILABLE TO FOCAL USERS. IT IS A COLLECTION OF PATCHES TO FOCAL, 1969 --- MOST PROGRAMS WRITTEN IN FOCAL, 1969 WILL BE PROCESSED BY PS/8 FOCAL, 1971 (SEE LIST OF DIFFERENCES ON PAGE 2).

PS/8 FOCAL, 1971 SOURCE AND BINARY FILES PLUS DEMONSTRATION PROGRAMS ARE AVAILABLE ON DECTAPE OR PAPER TAPE. ENCLOSE \$5 OR AN EXTRA DECTAPE TO COVER HANDLING AND MAILING. (ADD \$10 FOR DECTAPE VERSION IF YOU DO NOT INCLUDE YOUR OWN PS/8 FORMATTED DECTAPE.) SEND REQUESTS TO:

SOFTWARE DEVELOPMENT GROUP, **OMSI**, PORTLAND, OREGON 97221.

### SUGGESTIONS FOR NOVICES:

THIS MANUAL IS INTENDED AS A REVIEWING AID FOR PERSONS WHO ALREADY HAVE SOME FAMILIARITY WITH FOCAL. A GOOD INTRODUCTION TO FOCAL PROGRAMMING IS CHAPTER 11 OF 'PROGRAMMING LANGUAGES' (1970, \$2); ALSO USEFUL IS THE PS/8 USER'S GUIDE (NO. DEC-08-MEFA-D, \$3). BOTH MAY BE PURCHASED FROM THE DEC PROGRAM LIBRARY, BLDG. 3-5, MAYNARD, MASS. 01754.

TO USE PS/8 FOCAL, 1971 ON DECTAPE SYSTEMS (DISK USERS SEE PAGE 24):

- 1) BOOTSTRAP INTO PS/8 MONITOR;
- 2) AT MULTIPLE DECTAPE INSTALLATIONS, THE AUTHORS SUGGEST THAT THE PS/8 COMMAND 'ASSIGN DTA1: DSK:' BE GIVEN TO PERMIT THE USER TO USE HIS OWN DECTAPE AS THE DEFAULT DEVICE (DSK:); THIS IS REQUIRED BY THE 'HELP' PACKAGE AVAILABLE ON DECTAPE FOR PS/8 FOCAL, 1971;
- 3) TYPE 'RUN DSK: PFOCAL';
- 4) THE INITIAL DIALOGUE WILL ASK IF YOU WISH TO RETAIN STANDARD FEATURES OF PS/8 FOCAL, 1971. A 'YES' WILL PROVIDE 3500 CHARACTERS OF PROGRAM SPACE, 98 VARIABLES, 10 DIGIT PRECISION, FILES, AND TRIG FUNCTIONS. SEE 'INITIAL DIALOGUE' ON PAGE 22 FOR OTHER OPTIONS;
- 5) THE 'LIBRARY EXIT' COMMAND WILL RETURN YOU TO THE PS/8 MONITOR;
- 6) A 'HELP' PACKAGE FOR PS/8 FOCAL, 1971 IS AVAILABLE ON DECTAPE. MOUNT THE HELP TAPE ON DECTAPE DRIVE #1; MOUNT THE PS/8 SYSTEM TAPE ON DRIVE #0. YOU MAY ENTER 'HELP' WHILE WRITING A PS/8 FOCAL PROGRAM BY TYPING THE COMMAND 'L G HELP'. YOUR PROGRAM WILL BE SAVED UNDER THE NAME 'FOCAL.TM'. WHEN YOU DESIRE NO MORE HELP, YOU WILL BE RETURNED TO YOUR OWN PROGRAM. SEE ALSO THE HELP INDEX ON PAGE 27;
- 7) CTRL/P IS PS/8 FOCAL'S BREAK CHARACTER.

THIS MANUAL WAS TYPESET ON AN INEXPENSIVE MATRIX PRINTER MADE BY:

**CENTRONICS DATA COMPUTER CORP.**  
ONE WALL STREET  
HUDSON, NEW HAMPSHIRE 03051  
TELEPHONE (603) 883-0111

**\*\*\*\*\* CHANGES FROM FOCAL, 1969 :**

PROGRAMS WRITTEN IN FOCAL, 1969 SHOULD REQUIRE ONLY ONE CHANGE TO RUN UNDER PS/8 FOCAL, 1971:

THE INPUT DEVICE IS SWITCHED TO THE HIGH SPEED PAPER TAPE READER WITH THE 'OPEN INPUT PTR:' COMMAND AND SWITCHED BACK TO THE TERMINAL WITH 'OPEN INPUT TTY:,ECHO' INSTEAD OF WITH THE '\*' COMMAND OF FOCAL, 1969.

OTHER CHANGES INCLUDE:

- 1) EXTENDED LIBRARY FEATURES INCLUDING DEVICE-INDEPENDENT CHAINING AND SUBROUTINE CALLS BETWEEN PROGRAMS.
- 2) FILE READING AND WRITING COMMANDS, 10 DIGIT PRECISION, OR STANDARD TRIG FUNCTIONS MAY BE DELETED BY THE INITIAL DIALOGUE TO ALLOW UP TO 222 VARIABLES WITH 5500 CHARACTER PROGRAMS; WITH ALL FEATURES RETAINED, 98 VARIABLES AND 3500 CHARACTERS REMAIN. SEE PAGE 22.
- 3) COMPUTED LINE NUMBERS; SEE PAGE 20.
- 4) TABULATION.
- 5) IMPROVED DISTRIBUTION AND 'RANDOM' INITIALIZATION OF 'FRAN()', THE PSEUDO-RANDOM NUMBER FUNCTION.
- 6) UNLIMITED LINE LENGTHS.
- 7) CHARACTER MANIPULATIONS HANDLED WITH 'FIN()' AND 'FOUT()'.  
 8) FSGN(0)=0 [FSGN(0)=1 IN FOCAL, 1969]
- 9) 'ERASE ALL' AND 'MODIFY' DO NOT ERASE VARIABLES.
- 10) IN ADDITION TO THE FDIS() DISPLAY FUNCTION FOR DEC'S 340 DISPLAY, VERSIONS ARE AVAILABLE WITH DISPLAY AND JOYSTICK FUNCTIONS FOR THE TEKTRONIX T-4002 GRAPHICS TERMINAL.
- 11) 3 SPECIAL VARIABLES ARE PROTECTED FROM THE 'ERASE' COMMAND --- '!', '"', AND '#'. '#' IS SET TO 1 BY THE INITIAL DIALOGUE IF ALL OPTIONAL FEATURES OF PS/8 FOCAL ARE DELETED AND TO 2310 IF STANDARD FEATURES ARE RETAINED; SEE 'INITIAL DIALOGUE', PAGE 22.
- 12) PS/8 FOCAL'S STARTING ADDRESS IS 10200 (FIELD 1).

**----- CTRL/P IS THE BREAK CHARACTER -----**

NOTE: PS/8 FOCAL DATA FILES ARE COMPATIBLE WITH EDIT AND TECO-8; HOWEVER, PS/8 FOCAL PROGRAM FILES ARE SAVED AS CORE IMAGES.

## COMMANDS

-----

\*\*\* DIRECT COMMANDS ARE GIVEN WHILE FOCAL IS IN COMMAND MODE. THEY ARE TYPED WITHOUT LINE NUMBERS AND FOCAL EXECUTES THEM AS SOON AS THE RETURN BUTTON IS HIT --- FOR EXAMPLE:

```
*TYPE 3+4, !           (FOCAL OUTPUTS THE VALUE OF 3+4)
  7. 0000
*
```

(OR SEVERAL COMMANDS MAY BE GIVEN IN SEQUENCE:)

```
*SET X=3
*SET Y=2
*TYPE 3+X+Y, !
  8. 0000
*
```

\*\*\* INDIRECT COMMANDS ARE USED FOR LONGER PROGRAMS. THEY ARE TYPED FOLLOWING LINE NUMBERS BETWEEN 1. 01 AND 31. 99 NOT INCLUDING INTEGERS AND MAY BE EXECUTED BY A DIRECT 'GO':

```
*2. 1 SET X=3
*2. 2 SET Y=2
*2. 3 TYPE 3+X+Y, !
*GO
  8. 0000
*
```

### COMMAND FORMAT :

THE GENERAL FORM OF EACH IS GIVEN FOLLOWED BY EXAMPLES IN WHICH:

⟨ ⟩ ENCLOSE REQUIRED TERMS. [ ] ENCLOSE OPTIONAL TERMS.  
 ( ) ENCLOSE COMMENTS. ONE LETTER ABBREVIATIONS MAY BE USED FOR COMMAND WORDS. X REPRESENTS A VARIABLE. E1, E2 AND E3 ARE ARITHMETIC EXPRESSIONS. L1, L2 AND L3 ARE LINE NUMBERS. G1 IS EITHER A LINE OR GROUP NUMBER. L1, L2, L3 AND G1 MAY BE REPLACED WITH ANY ARITHMETIC EXPRESSION NOT BEGINNING WITH THE LETTER 'A'.

MOST COMMANDS MUST BE FOLLOWED BY A SPACE.

## ELEMENTARY COMMANDS

---

### ASK

\*ASK ["PRINT OUT IN QUOTES"] X[, Y, Z]           <INPUT VALUES FOR VARIABLES>

ASK Y                   <EXPECTS A VALUE TO BE INPUT FOR Y>  
 A Z                    <A IS THE ABBREVIATION FOR 'ASK'>  
 A B2, X, NUMBER       <EXPECTS THREE VALUES TO BE INPUT>  
 A "HOW MANY?" "M      <TYPES 'HOW MANY' THEN SETS M EQUAL TO THE RESPONSE>

\* \* \* SPACE, COMMA OR CARRIAGE RETURN WILL TERMINATE EACH NUMBER INPUT. TYPE A '←' (BACK ARROW) TO ERASE A PARTIALLY ENTERED NUMBER WHICH MAY THEN BE RE-ENTERED. ALT MODE WILL CAUSE FOCAL TO LEAVE THE VARIABLE WITH ITS FORMER VALUE. IF 'ASKING' FROM A FILE --- A SPACE, COMMA OR OTHER DELIMITER MUST PRECEDE A NEGATIVE SIGN, '-', OR THE NUMBER WILL BE READ AS POSITIVE. RESPONDING TO INITIAL DIALOGUE WITH 'YES' OR '4,5' WILL REMEDY THIS BY OUTPUTTING A LEADING SPACE WHEN USING 'TYPE'.

### COMMENT

\*COMMENT               <LINES BEGINNING WITH A 'C' WILL BE IGNORED>

COMMENT SORT           <COMMENTS ARE USED TO DESCRIBE PROGRAMS>  
 C                      <ABBREVIATION>  
 CONTINUE               <DUMMY LINE>

### DO

\*DO [LINE OR GROUP NUMBER]                   <SUBROUTINE CALL>

DO 7. 24                <TELLS FOCAL TO EXECUTE LINE 7. 24 THEN RETURN  
 D 4. 48                TO THE COMMAND FOLLOWING THE 'DO'>  
 DO 7                    <EXECUTES GROUP 7>  
 DO ALL                 <STARTS EXECUTION OF PROGRAM AT FIRST LINE>  
 D G1; D 7. 6            <EXECUTES LINE OR GROUP G1 THEN LINE 7. 6>

\* \* \* DO NOT USE AN ARITHMETIC EXPRESSION STARTING WITH 'A' AS A COMPUTED LINE NUMBER OR IT WILL 'DO ALL'.

### ERASE

\*ERASE [LINE OR GROUP NUMBER]               <ZEROS VARIABLES IF NO NUMBER GIVEN>

ERASE 2. 17             <DELETES LINE 2. 17 FROM INDIRECT PROGRAM>  
 E 3                    <DELETES ALL GROUP 3 LINES>  
 ERASE                 <ZEROS ALL VARIABLES>  
 ERASE ALL             <ERASES ENTIRE INDIRECT PROGRAM>

\* \* \* 'ERASE' MUST BE THE LAST COMMAND ON A LINE. 'ERASE <LINE OR GROUP NUMBER>' WILL CAUSE FOCAL TO RETURN TO COMMAND MODE AFTER ERASING.



**FOR**

\*FOR X=E1[, E2], E3; [COMMAND TO BE REPEATED (E3-E1)/E2+1 TIMES]

FOR P=1, 7; TYPE 3 (WILL TYPE A '3' SEVEN TIMES)  
 FOR J=1, 5; DO 17. 1 (WILL DO LINE 17. 1 FIVE TIMES)  
 F X=2, 2, 8; T X\*10 (WILL TYPE '20 40 60 80')  
 FOR Z=1, Y; DO 7 (WILL DO GROUP 7, Y TIMES)

\* \* \* THE 'FOR' COMMAND IS REPEATED --- ADDING 1 (OR E2) TO E1 EACH CYCLE UNTIL E1 EXCEEDS E3. EXECUTION THEN GOES TO THE NEXT LINE.

**GO**

\*GO [LINE NUMBER] (STARTS EXECUTING PROGRAM AT DESIGNATED LINE)

GO (STARTS EXECUTING INDIRECT PROGRAM AT FIRST LINE)  
 G (ABBREVIATION)  
 GOTO 3. 1 (STARTS EXECUTING PROGRAM AT LINE 3. 1)  
 GO ? ('TRACE FEATURE' -- PRINTS PROGRAM DURING EXECUTION UNTIL NEXT OCCURRENCE OF '?')  
 G X (STARTS EXECUTING PROGRAM AT LINE X)

**IF**

\*IF (E1)L1[, L2, L3] (GOES TO LINE L1, L2, OR L3 DEPENDING ON WHETHER E1 IS -, 0, OR +)

IF (X)2. 7, 1. 2, 3. 8 (FOCAL GOES TO LINE 2. 7, 1. 2 OR 3. 8 IF X IS -, 0 OR +)  
 I (H-5)1. 3, 1. 5, 1. 7 ('IF' CHECKS THE VALUE INSIDE THE PARENTHESES)  
 I (B-3)2. 8, 7. 9; 0 ('QUITS' EXECUTION IF (B-3) IS POSITIVE)  
 I (Z)2. 7; T 21 (IF Z IS NEGATIVE THEN GO TO 2. 7; ELSE 'TYPE 21')  
 I (G)2. 1, Y, 2. 4 (IF G=0 THEN GOES TO LINE Y)

**MODIFY**

\*MODIFY <LINE NUMBER> (EDIT LINE)

'MODIFY 3. 72' FOLLOWED BY A CARRIAGE RETURN AND A CHARACTER 'X' WILL OUTPUT LINE 3. 72 THROUGH ITS FIRST 'X'. YOU MAY THEN:

- 1) INPUT THE REST OF THE REVISED LINE AND HIT 'RETURN'
- 2) USE 'RUB OUT' TO ERASE SINGLE CHARACTERS
- 3) HIT 'LINE FEED' TO KEEP REMAINDER OF OLD LINE
- 4) HIT CTRL/L TO GO TO NEXT OCCURRENCE OF 'X'
- 5) HIT CTRL/BELL TO CHANGE SEARCH CHARACTER; THEN HIT NEW CHARACTER
- 6) HIT '<' TO KILL LINE OVER TO LEFT MARGIN

**QUIT**

\*QUIT (TERMINATES PROGRAM EXECUTION)

Q (ABBREVIATION)

**RETURN**

\*RETURN           <CAUSES ESCAPE FROM A SUBROUTINE TO THE COMMAND  
                  FOLLOWING THE CALLING 'DO' OR 'LIBRARY GOSUB'>

R                   <ABBREVIATION>

**SET**

\*SET Y=<NUMBER, VARIABLE OR EXPRESSION>       <SETS VARIABLE VALUES>

SET Y=37           <CAUSES 'Y' TO TAKE THE VALUE 37>

S A=110/P+32       <CAUSES 'A' TO ASSUME THE VALUE OF 110/P+32>

S Y=Y+1            <SETS THE NEW VALUE OF Y EQUAL TO 1 PLUS OLD VALUE>

S Z=#              <SETS Z EQUAL TO VALUE OF PROTECTED VARIABLE '#'>

S #=0              <ONLY WAY TO ZERO PROTECTED VARIABLE '#'>

S X=FIN(<)         <SETS X TO DECIMAL ASCII CODE OF NEXT CHARACTER INPUT>

S DUMMY=FOUT(203) <OUTPUTS LETTER 'K' AND SETS DU=203>

**TYPE**

\*TYPE [NUMBERS, VARIABLES, EXPRESSIONS, TEXT IN QUOTES, !, #, :, \$ OR %]

TYPE 3            <OUTPUTS A 3 ON THE TERMINAL OR OTHER OUTPUT DEVICE>

T X               <OUTPUTS THE VALUE OF X>

T X/Y-1           <OUTPUTS THE VALUE OF THE EXPRESSION X/Y-1>

T "HI THERE"      <OUTPUTS THE GREETING 'HI THERE'>

T !               <OUTPUTS A CARRIAGE RETURN/LINE FEED>

T "A!" "B!" "C!" <OUTPUTS CHARACTERS 'A', 'B' AND 'C' VERTICALLY>

T #               <OUTPUTS A CARRIAGE RETURN WITHOUT LINE FEED>

T (#)             <OUTPUTS VALUE OF SYSTEM STATUS VARIABLE, '#'>

T "ANS: ", 26+5   <OUTPUTS 'ANS: 31'>

T !:8, "\*"        <OUTPUTS RETURN/LINE FEED THEN A '\*' IN POSITION 8>

T \$               <OUTPUTS THE SYMBOL TABLE>

T %10. 07         <SETS THE OUTPUT FORMAT TO ALLOW UP TO 10 DIGITS  
                  TO BE PRINTED, OF WHICH 7 MAY BE DECIMAL PLACES>

T %               <FORMATS OUTPUT IN POWERS-OF-TEN NOTATION>

\* \* \* THE INITIAL DIALOGUE'S STANDARD FEATURES OUTPUTS A LEADING SPACE BEFORE EACH NUMBER IS 'TYPED' --- THIS PERMITS NEGATIVE NUMBERS TO BE READ AS NEGATIVE FROM A FILE WRITTEN IN THIS MANNER. FEATURE 6 PERMITS OUTPUTTING A '=' AS IN FOCAL, 1969 (NOT RECOMMENDED FOR WRITING FILES).

**WRITE**

\*WRITE [GROUP OR LINE NUMBER]       <LISTS INDIRECT PROGRAM>

WRITE 2. 27       <OUTPUTS LINE 2. 27 OF THE INDIRECT PROGRAM>

W 2               <OUTPUTS GROUP 2>

WRITE ALL         <OUTPUTS ENTIRE INDIRECT PROGRAM>

W A               <ABBREVIATION FOR 'WRITE ALL'>

W                 <ABBREVIATION FOR 'WRITE ALL'>

W G1              <OUTPUTS GROUP OR LINE GIVEN BY 'G1'>

\* \* \* DO NOT USE AN ARITHMETIC EXPRESSION STARTING WITH 'A' AS A COMPUTED LINE NUMBER OR IT WILL 'WRITE ALL'. 'WRITE' MUST BE THE LAST COMMAND ON A LINE.

## LIBRARY COMMANDS

### PS/8 DEVICE NAMES :

SYS: SYSTEM DEVICE (DSK: IN DISK SYSTEM; DTA0: IN DECTAPE SYSTEM)  
 DSK: THE DISK IN DISK SYSTEMS; DECTAPE #0 IN DECTAPE SYSTEMS  
 (THE AUTHORS SUGGEST THAT DECTAPE SYSTEMS ASSIGN DTA1:DSK:)  
 (<'DSK:' IS ASSUMED IF A DEVICE IS NOT SPECIFIED>)  
 DTA0: - DTA7: DECTAPE DRIVES  
 LTA0: - LTA7: LINCTAPE DRIVES  
 MTA0: - MTA7: MAGTAPE DRIVES  
 PTR: HIGH SPEED READER  
 PTP: HIGH SPEED PUNCH  
 LPT: LINE PRINTER  
 TTY: TERMINAL

REFERENCING A NON-EXISTENT DEVICE MAY CAUSE THE COMPUTER TO ENTER AN INFINITE LOOP WAITING FOR THE DEVICE. RECOVERY IS MADE BY USING THE SWITCH REGISTER --- LOAD ADD 10200 AND START.

### LIBRARY CALL

\*LIBRARY CALL [DEVICE:]<PROGRAM NAME> (LOADS SAVED PROGRAM)  
 LIBRARY CALL CHISQR (LOADS PROGRAM 'CHISQR.FC' FROM DSK:)  
 L C DTA3:PRGRAM (LOADS 'PRGRAM.FC' FROM DECTAPE DRIVE #3)  
 L C TEST1 (LOADS 'TEST1.FC' FROM DSK:)

\* \* \* FOCAL RETURNS TO COMMAND MODE FOLLOWING A 'LIBRARY CALL'.

### LIBRARY DELETE

\*LIBRARY DELETE [DEVICE:]<PROGRAM NAME> (DELETES FILE FROM DEVICE)  
 LIBRARY DELETE TTEST (DELETES PROGRAM 'TTEST.FC' FROM DSK:)  
 L D DTA7:PROG (DELETES 'PROG.FC' FROM DECTAPE DRIVE #7)  
 L D HOLD.FD (DELETES DATA FILE 'HOLD.FD')

\* \* \* LIBRARY DELETE ASSUMES '.FC' FILE NAME EXTENSION; OTHERS MUST BE SPECIFIED. 'LIBRARY DELETE' CLOSSES ANY OPEN OUTPUT FILES.

### LIBRARY GOSUB

\*LIBRARY GOSUB [DEVICE:]<PROGRAM NAME> [GROUP OR LINE NUMBER]  
 LIBRARY GOSUB TEXT 13.7 (LINE 13.7 OF 'TEXT.FC' BECOMES SUBROUTINE WHICH  
 RETURNS TO THE COMMAND FOLLOWING 'GOSUB')  
 L G SUMSQR (TREATS WHOLE 'SUMSQR.FC' PROGRAM AS SUBROUTINE)  
 L G DTA3:X.Y 7 (TREATS GROUP 7 OF 'X.Y' LIKE A 'DO' SUBROUTINE)  
 L G BOBO G1 ('GOSUB' TO LINE OR GROUP EQUAL TO VALUE OF G1.  
 DO NOT USE EXPRESSIONS BEGINNING WITH 'A'.)

\* \* \* IF THE VERSION OF A PROGRAM ISSUING A 'GOSUB' IS NOT SAVED, ANY OPEN OUTPUT FILE IS CLOSED AND THE PROGRAM IS SAVED AS 'FOCAL.TM' BEFORE EXECUTION OF THE 'GOSUB'.

## LIBRARY LIST

\*LIBRARY LIST [DEVICE:] [FIRST FILE NAME TO BE LISTED]  
 (LISTS PROGRAM AND DATA FILE NAMES WITH FOCAL  
 EXTENSIONS '.FC'=PROGRAM; '.FD'=DATA FILES)

LIBRARY LIST (LISTS FOCAL .FC AND .FD FILES SAVED ON DSK:)  
 L L (ABBREVIATION)  
 LIBRARY LIST DTA3: (LISTS FILES SAVED ON DECTAPE DRIVE #3)  
 L L DTA6:TEST (STARTS LISTING WITH 'TEST.FC')

\* \* \* FILE LENGTH IS PRINTED IN BLOCKS; 1 BLOCK IS ABOUT 500 CHARACTERS.

## LIBRARY RUN

\*LIBRARY RUN [DEVICE:] <PROGRAM NAME> [LINE NUMBER]  
 (BEGINS EXECUTION OF NAMED PROGRAM)

LIBRARY RUN JOHN (LOADS 'JOHN.FC' THEN BEGINS PROGRAM EXECUTION)  
 L R DTA2:ZONK (RUNS 'ZONK.FC' FROM DECTAPE DRIVE #2)  
 L R POP 22.81 (STARTS EXECUTING 'POP.FC' AT LINE 22.81)  
 L R ZOWIE E1 (STARTS EXECUTING 'ZOWIE.FC' AT LINE E1)

\* \* \* PROGRAMS CONTAINING 'RUNS' SHOULD BE SAVED BEFORE EXECUTION.

## LIBRARY SAVE

\*LIBRARY SAVE [DEVICE:] <PROGRAM NAME>

LIBRARY SAVE PROG (SAVES INDIRECT PROGRAM PRESENTLY IN THE TEXT  
 BUFFER AREA UNDER THE NAME 'PROG.FC' ON DSK:)  
 L S DTA6:ZAAP (SAVES 'ZAAP.FC' ON DECTAPE DRIVE #6)  
 L S PRGNAM (SAVES 'PRGNAM.FC' ON DSK:)

\* \* \* OLD 'PRGNAM' IS DELETED WHEN NEW 'PRGNAM' IS SAVED. THIS  
 COMMAND CLOSES OPEN OUTPUT FILES.

## LIBRARY EXIT

\*LIBRARY EXIT (LEAVES FOCAL; RETURNS TO PS/8 MONITOR)  
 L E (ABBREVIATION)

## FILE COMMANDS

THE EXPERIENCED PROGRAMMER MAY READ AND WRITE PS/8 COMPATIBLE DATA FILES WITH MANY DEVICES. SEE PS/8 DEVICE NAMES ON PAGE 7.

PS/8 FOCAL ASSUMES .FC AND .FD AS NAME EXTENSIONS FOR PROGRAM AND DATA FILES RESPECTIVELY. DATA FILES ARE SAVED IN STANDARD PS/8 ASCII FORMAT AND ARE COMPATIBLE WITH EDIT AND TECO-8. PROGRAM FILES ARE SAVED IN CORE IMAGE FORMAT AND MAY BE TRANSFERRED BY PIP ONLY WITH THE '/I' OPTION. TO PRODUCE AN ASCII FILE CONTAINING A FOCAL PROGRAM, OPEN AN OUTPUT FILE: 'WRITE ALL' THEN 'OUTPUT CLOSE'.

THE TERMINAL (TTY:) IS PS/8 FOCAL'S NORMAL INPUT AND OUTPUT DEVICE. THE 'OPEN INPUT' COMMAND PERMITS SELECTING ANOTHER DEVICE FOR INPUT (DEVICES LISTED ON PAGE 7). 'OPEN OUTPUT' DOES THE SAME FOR OUTPUT. BE SURE TO CLOSE ONE OUTPUT FILE BEFORE OPENING ANOTHER.

'ECHO' ADDED TO THE INPUT COMMANDS ECHOES THE INPUT DATA ON THE OUTPUT DEVICE. WHEN ADDED TO OUTPUT COMMANDS, 'ECHO' CAUSES THE OUTPUT TO BE ECHOED ON THE TERMINAL.

'OUTPUT CLOSE' EMPTIES THE OUTPUT BUFFER ONTO THE DEVICE AND SAVES THE FILE IF THE DEVICE IS FILE STRUCTURED (DISK OR MAGNETIC TAPE).

THE 'OPEN INPUT TTY:,ECHO' COMMAND WILL RESTORE THE TERMINAL AS THE INPUT DEVICE WITH ECHO. AFTER AN ERROR OCCURS OR 'CTRL/P' IS STRUCK, THE TERMINAL IS RESTORED AS BOTH INPUT AND OUTPUT DEVICE.

READING AND WRITING CAN BE RESUMED WITH PREVIOUSLY OPENED NON-TTY: DEVICES THROUGH 'OPEN RESTORE INPUT' AND 'OPEN RESTORE OUTPUT' COMMANDS.

DIRECT FILE COMMANDS MAY BE EXECUTED AS MENTIONED UNDER 'MERGING PROGRAMS' OR A SERIES OF DIRECT COMMANDS MAY BE EXECUTED AUTOMATICALLY FROM A FILE. FOR EXAMPLE, IF YOU WISHED TO GET LISTINGS OF SEVERAL PROGRAMS, YOU COULD PREPARE THE FILE 'LISTAL.FD' USING EDIT OR TECO-8:

```

TYPE !!!!"LISTING OF TEST1-TEST3:"!!!!
LIBRARY CALL TEST1
WRITE ALL
TYPE !!!!
LIBRARY CALL TEST2
WRITE ALL
TYPE !!!!
LIBRARY CALL TEST3
WRITE ALL
TYPE !!!!
OPEN INPUT TTY:,ECHO

```

THEN, BY GIVING THE DIRECT FOCAL COMMAND 'OPEN INPUT LISTAL', ALL 3 PROGRAMS WOULD BE OUTPUT WITHOUT YOUR INTERVENTION.

NOTE: WHEN WRITING FOCAL DATA FILES, IT IS NECESSARY TO INCLUDE A SPACE, COMMA, CARRIAGE RETURN OR OTHER DELIMITER PRECEDING A MINUS SIGN, OTHERWISE THE NUMBER WILL APPEAR POSITIVE WHEN 'ASKED'. A PRECEDING SPACE WILL BE AUTOMATICALLY 'TYPED' IF THE INITIAL DIALOGUE RECEIVES AN ANSWER OF 'YES' OR '4,5'.

**OPEN INPUT**

\*OPEN INPUT [DEVICE:][FILE NAME],ECHO] (PREPARES A FILE TO BE READ WITH 'ASK' OR 'FIN()')

OPEN INPUT BLOOP (OPENS 'BLOOP.FD' FILE FROM DSK:)  
 O I DTA1:RED,ECHO (OPENS 'RED.FD' FILE FROM DECTAPE DRIVE #4 AND WILL ECHO ON OUTPUT DEVICE WHILE READING 'RED')  
 O I PTR: (PS/8 WILL TYPE AN '~' ON THE TERMINAL; HIT THE SPACE BAR AFTER PAPER TAPE IS LOADED IN READER.)  
 OPEN INPUT TTY:,ECHO (RESTORES TERMINAL TO NORMAL STATUS)

\* \* \* CTRL/Z IS THE LAST CHARACTER IN A FOCAL DATA FILE. ATTEMPTS TO READ PAST THE END-OF-FILE WILL CAUSE A '?' TO BE OUTPUT ON THE TERMINAL AND WILL RESTORE THE TERMINAL AS INPUT DEVICE.

**OPEN OUTPUT**

\*OPEN OUTPUT [DEVICE:][FILE NAME],ECHO] (PREPARES A FILE TO BE WRITTEN WITH 'TYPE', 'FOUT(A)' OR ECHOING FROM INPUT FILE)

OPEN OUTPUT ABC (OPENS 'ABC.FD' TO BE WRITTEN ON THE DSK:)  
 O O DTA7:ZZ (OPENS 'ZZ.FD' FILE ON DECTAPE DRIVE #7)  
 O O PTP:,ECHO (ECHOES ON TTY: WHILE OUTPUTTING TO PUNCH; BE SURE TO GIVE AN 'OUTPUT CLOSE' WHEN DONE)

\* \* \* OPEN OUTPUT FILES WILL BE CLOSED IF THE COMMAND 'LIBRARY SAVE' OR 'LIBRARY DELETE' IS GIVEN OR IF 'LIBRARY GOSUB' IS GIVEN BY A VERSION OF A PROGRAM THAT HAS NOT BEEN SAVED.

**OPEN RESTORE INPUT**

\*OPEN RESTORE INPUT[,ECHO] (RESUMES INPUT FROM FILE)

OPEN RESTORE INPUT (RESUMES INPUTTING DATA FROM A PREVIOUSLY OPENED INPUT FILE AFTER USING THE TTY: WITH AN 'OPEN INPUT TTY:,ECHO')

O R I (SAME)  
 O R I,E (SAME PLUS ECHO)

**OPEN RESTORE OUTPUT**

\*OPEN RESTORE OUTPUT[,ECHO] (RESUMES OUTPUT TO FILE)

OPEN RESTORE OUTPUT (RESUMES OUTPUTTING ON PREVIOUSLY OPENED OUTPUT DEVICE AFTER USING TTY:)

O R O (SAME)  
 O R O,ECHO (SAME PLUS ECHO)

**OUTPUT CLOSE**

\*OUTPUT CLOSE (ENDS FILE WRITING AND SAVES OUTPUT FILE IF DEVICE IS FILE STRUCTURED --- DISK OR MAGNETIC TAPE)

O C (ABBREVIATION)

# FILE AND CHARACTER HANDLING PROGRAMS

---

## 'WORDS'

GROUP 1 SETS UP THE 'WORDS' FILE. GROUP 2 SCANS IT, ECHOING ONLY THE WORDS WHICH START WITH THE SEARCH LETTER. OBSERVE THE PROGRAM LISTING AND EXECUTION; ALSO NOTE COMPUTED LINE NUMBER 'P' USED IN LINE 2.60 TO MINIMIZE TIME IN THE CHARACTER INPUT LOOP:

```
*WRITE ALL
C-PS/8 FOCAL, 1971
```

```
01.05 COMMENT: 'WORDS' 7/3/71 BY J. HUISMAN OF OMSI
01.10 OPEN OUTPUT DTA2:WORDS; OPEN OUTPUT TTY.; ERASE
01.20 TYPE !"ENTER WORDS; FOLLOW EACH WITH A SPACE:!"
01.30 TYPE "(ENTER A SPACE BY ITSELF TO CLOSE FILE)"!!
01.40 TYPE :5"? "; OPEN RESTORE OUTPUT, ECHO
01.50 IF (-FABS(FIN()-160))1.5; OPEN OUTPUT TTY:
01.60 TYPE !"MORE? "; OPEN RESTORE OUTPUT, ECHO
01.70 IF (-FABS(FIN()-160))1.5; OUTPUT CLOSE; GOTO 2.7
```

```
02.10 TYPE !"ENTER SEARCH LETTER: "; SET L=FIN(); T !!!
02.20 TYPE "WORDS STARTING WITH "; S D=FOUT(L); T "!"!5
02.30 OPEN INPUT DTA2:WORDS; SET S=160; SET P=2.6
02.40 SET C=FIN(); IF (FABS(C-S)), 2.7; I (-FABS(L-C))2.6
02.50 OPEN RESTORE INPUT, ECHO; T !:5; SET DUMMY=FOUT(C)
02.60 I (-FABS(FIN()-S))P; OPEN RESTORE INPUT; G 2.4
02.70 OPEN INPUT TTY.; ECHO; TYPE !"DONE. !!!" QUIT
*GO
```

```
ENTER WORDS; FOLLOW EACH WITH A SPACE;
(ENTER A SPACE BY ITSELF TO CLOSE FILE)
```

```
? KINCAID
MORE? BUTLER
MORE? CRAWFORD
MORE? KAY
MORE? POLLARD
MORE? KLATT
MORE?
```

```
DONE.
```

```
*GOTO 2.1
```

```
ENTER SEARCH LETTER: K
```

```
WORDS STARTING WITH K:
```

```
KINCAID
KAY
KLATT
```

```
DONE.
```

```
*
```

## ' ALPHA '

' ALPHA ' ALPHABETIZES A ' WORDS. FD ' FILE PREPARED BY THE ' WORDS. FC ' PROGRAM ON PAGE 11.

NOTE THAT THESE PROGRAMS ARE EXAMPLES ONLY AND NOT RECOMMENDED FOR GENERAL USE SINCE ERROR CHECKING THAT WOULD MAKE THEM USER-ORIENTED WAS EXCLUDED FOR THE SAKE OF BREVITY.

\*WRITE ALL

C-PS/8 FOCAL, 1971

01. 05 COMMENT: ' ALPHA ' 8/1/71 BY B. SMITH OF OMSI

01. 10 OPEN INPUT DTA2:WORDS

01. 15 OPEN OUTPUT DTA2:WORDS, ECHO; ERASE

01. 20 DO 10; IF <LE>, 3. 3

01. 25 SET LE<1>=LE; FOR J=1, LE; SET B<J>=A<J>; SET A<J>=0

01. 30 DO 10; IF <LE>, 3. 1

01. 40 IF <LE-LE<1>>1. 45; SET N=LE; GOTO 1. 5

01. 45 SET N=LE<1>

01. 50 FOR I=1, N; DO 2

01. 60 GOTO 1. 3

02. 10 IF <A<I>-B<I>>2. 4, 2. 3

02. 15 FOR J=1, LE<1>; SET D=FOUT<B<J>>; SET B<J>=0

02. 20 TYPE " "; SET I=N+1; DO 1. 25; RETURN

02. 30 IF <I-N>; DO 2. 4; TYPE " "; RETURN

02. 40 FOR J=1, LE; SET D=FOUT<A<J>>; SET A<J>=0

02. 50 SET I=N+1; SET CHANGE=CHANGE+1; TYPE " "

03. 10 FOR I=1, LE<1>; SET D=FOUT<B<I>>

03. 20 TYPE " "

03. 30 OUTPUT CLOSE; TYPE !; IF <CHANGE>, 3. 4.

03. 40 OPEN INPUT TTY:, ECHO; TYPE !, "DONE. ", !!; QUIT

10. 10 SET LE=0

10. 20 SET A<LE+1>=FIN<>; IF <A<LE+1>-154>10. 3.

10. 30 IF <A<LE+1>-160>10. 4, 10. 5

10. 40 SET LE=LE+1; GOTO 10. 2

10. 50 SET A<LE+1>=0

\*GO

791 AND QUITE 124 123 ALL NUMERICS ORDERED ALPHABETICS USEFULLY

791 AND 124 123 ALL NUMERICS ORDERED ALPHABETICS QUITE USEFULLY

791 124 AND 123 ALL NUMERICS ORDERED ALPHABETICS QUITE USEFULLY

124 791 123 ALL AND NUMERICS ALPHABETICS ORDERED QUITE USEFULLY

124 123 791 ALL AND ALPHABETICS NUMERICS ORDERED QUITE USEFULLY

123 124 791 ALL ALPHABETICS AND NUMERICS ORDERED QUITE USEFULLY

123 124 791 ALL ALPHABETICS AND NUMERICS ORDERED QUITE USEFULLY

DONE.

\*



## SYMBOLS

-----

### ARITHMETIC SYMBOLS :

EXPONENTIATION      [^]      <INDICATES AN INTEGER EXPONENT>  
 TYPE 3^2            <OUTPUTS A '9' --- 3 TO THE SECOND POWER>  
 SET X=Y^Z           <SETS X EQUAL TO Y RAISED TO THE INTEGRAL Z POWER>  
 TYPE 2^3.5          <PRINTS AN '8' --- THE .5 IS IGNORED>

MULTIPLICATION      [\*]      <OUTPUTS AN '8'>  
 TYPE 2\*4            <SETS THE NEW VALUE OF X EQUAL TO X TIMES Z>  
 SET X=X\*Z

DIVISION             [/]      <OUTPUTS A '3'>  
 TYPE 6/2            <SETS X EQUAL TO Y DIVIDED BY Z>  
 SET X=Y/Z

ADDITION             [+]      <OUTPUTS A '4'>  
 TYPE 2+2            <SETS X EQUAL TO Y PLUS Z>  
 SET X=Y+Z

SUBTRACTION         [-]      <OUTPUTS THE VALUE OF 3-X>  
 TYPE 3-X            <SETS X EQUAL TO Y MINUS Z>  
 SET X=Y-Z

ENCLOSURES           <>, [ ], AND <> MAY BE USED IN MATCHED PAIRS  
 TO ENCLOSE QUANTITIES WHICH ARE TO BE OPERATED  
 ON AS A UNIT.

## SPECIAL CHARACTERS :

RETURN/LINE FEED [!]  
 TYPE !!!!!,X,!! (OUTPUTS 4 CARRIAGE RETURN/LINE FEEDS, THE VALUE OF X, AND THEN 2 MORE CARRIAGE RETURN/LINE FEEDS.)

CARRIAGE RETURN [#]  
 SEE WHAT THE '#/' DOES TO LINE 10.35 FIRST WRITTEN THEN EXECUTED:  
 (NOTE THAT THE '/' AND THE '=' ARE SUPERIMPOSED)  
 10.35 TYPE "X=5",#," /",!!!!  
 X#5

TAB [:]  
 TAB OUTPUTS ENOUGH SPACES SO THAT THE NEXT CHARACTER WILL BE OUTPUT AT THE POSITION ON THE LINE INDICATED BY THE VALUE AFTER THE ':'. TAB WILL HAVE NO EFFECT IF THE POSITION SPECIFIED HAS BEEN EXCEEDED. SEE WHAT THE ':/' DOES TO LINE 11.37 FIRST WRITTEN THEN EXECUTED:  
 11.37 TYPE "NAME---":25,"ROGER":40,"AGE = ",2\*13,!!!!  
 NAME--- ROGER AGE = 26.0000

OUTPUT FORMATTER [%]  
 SEE WHAT THE '%/' DOES TO LINE 12.30 FIRST WRITTEN THEN EXECUTED:  
 12.30 SET X=123.456; TYPE %6.04 X, !; TYPE % 4.02 X, !; T % 3.0 X, !; T %, X, !  
 123.456  
 123.5  
 124  
 0.1234560000E+03  
 '%4.02' PERMITS FOUR DIGITS TO BE OUTPUT INCLUDING UP TO TWO DECIMAL PLACES; 'TYPE %' ALONE GIVES POWERS-OF-TEN NOTATION. '%/' DOES NOT CHANGE CALCULATION PRECISION; THAT IS DONE WITH THE INITIAL DIALOGUE.

SYMBOL TABLE [\$]  
 TYPE \$ (OUTPUTS THE LIST OF DEFINED VARIABLES NOT INCLUDING PROTECTED VARIABLES. BE SURE TO USE AN OUTPUT FORMAT (%) THAT WILL NOT MASK PRECISION. SUBSCRIPTS OVER 99 WILL NOT OUTPUT ACCURATELY.)

QUOTATION MARKS ["]  
 WORDS MAY BE TYPED BY ENCLOSING THEM IN QUOTES.  
 SEE WHAT THE '"' DOES TO LINE 14.65 FIRST WRITTEN THEN EXECUTED:  
 14.65 TYPE "THE ANSWER IS ",3+8,!!!!  
 THE ANSWER IS 11.0000

TRACE [?]  
 THE FIRST TIME FOCAL READS A '?' (EXCEPT WITHIN QUOTES) IT WILL  
 START OUTPUTTING THE PROGRAM WHILE IT IS BEING EXECUTED. THE NEXT  
 TIME IT MEETS A '?' IT WILL STOP OUTPUTTING THE PROGRAM.  
 SEE WHAT THE '?' DOES TO LINE 15.60 FIRST WRITTEN THEN EXECUTED:  
 15.60 SET X=3;? SET Y=5;TYPE X/Y,!?  
 SET Y=5;TYPE X/Y, 0.6000!  
 A 'GO ?' COMMAND WILL CAUSE THE WHOLE PROGRAM TO BE 'TRACED'.

AN ATTEMPT TO READ PAST A CTRL/Z END-OF-FILE CHARACTER WILL OUTPUT  
 A '?' ON THE TERMINAL AND RESTORE THE TERMINAL AS THE INPUT DEVICE.

POWERS-OF-TEN [E OR &]  
 'E' MAY BE CHANGED TO '&' IN THE INITIAL DIALOGUE.  
 SEE WHAT THE 'E' DOES TO LINE 6.80 FIRST WRITTEN THEN EXECUTED:  
 06.80 TYPE 1E1,7E3,1.23E-2,3.76E217,!!!!  
 10.0000 7000.00 0.0123 0.376000E+218

RUBOUT KEY  
 WHEN WRITING A PROGRAM, ONE CHARACTER WILL BE DELETED FROM THE CURRENT  
 LINE WHEN THE RUBOUT KEY IS STRUCK; RUBOUT WILL ECHO AS A '\ ' --- THUS:  
 'PLWEA\\SE' BECOMES 'PLEASE'.

CTRL/C (PRESSING 'CTRL' AND 'C' AT THE SAME TIME)  
 WILL RETURN USER TO THE PS/8 MONITOR WHEN FOCAL IS USING PS/8  
 ROUTINES --- THE 'LIBRARY EXIT' COMMAND IS PREFERRED FOR  
 LEAVING PS/8 FOCAL, 1971. CTRL/P IS THE BREAK CHARACTER.

CTRL/G [BELL] (PRESSING 'CTRL' AND 'G' AT THE SAME TIME)  
 USED WITH THE 'MODIFY' COMMAND TO CHANGE THE SEARCH CHARACTER.

CTRL/L (PRESSING 'CTRL' AND 'L' AT THE SAME TIME)  
 USED WITH THE 'MODIFY' COMMAND TO SKIP TO THE NEXT SEARCH CHARACTER.

CTRL/P (PRESSING 'CTRL' AND 'P' AT THE SAME TIME)  
 IS PS/8 FOCAL'S BREAK CHARACTER --- INTERRUPTS PROGRAM EXECUTION AND  
 CAUSES FOCAL TO PRINT ITS '\*'.

CTRL/Z (PRESSING 'CTRL' AND 'Z' AT THE SAME TIME)  
 IS THE LAST CHARACTER IN A PS/8 FOCAL DATA FILE. ATTEMPTS TO READ  
 PAST THE END-OF-FILE WILL CAUSE A '?' TO BE TYPED ON THE TERMINAL AND  
 WILL RESTORE THE TERMINAL AS THE INPUT DEVICE.

## LINE FEED

USE WITH THE 'MODIFY' COMMAND TO RETAIN REMAINDER OF MODIFIED LINE.

## KILL LINE [←]

WHEN WRITING A PROGRAM, '←' DELETES EVERYTHING OVER TO LEFT MARGIN.  
IN RESPONSE TO 'ASK' COMMAND, '←' CANCELS THE NUMBER BEING ENTERED.

## RETURN KEY

USED WITH THE 'MODIFY' COMMAND TO DELETE REMAINDER OF MODIFIED LINE.  
'RETURN' IS A LEGAL SYMBOL FOR SEPARATING DATA BEING REQUESTED BY  
AN 'ASK' COMMAND.

## ALT MODE KEY

IN RESPONSE TO AN 'ASK' COMMAND, ALT MODE CAUSES THE VARIABLE  
TO RETAIN ITS PREVIOUS VALUE.

## SPACE KEY

IT IS A LEGAL SYMBOL FOR SEPARATING DATA REQUESTED BY AN 'ASK'.

## COMMA [, ]

IT IS A LEGAL SYMBOL FOR SEPARATING DATA IN 'TYPE' AND 'ASK' COMMANDS.  
SEE WHAT THE ',' DOES TO LINE 26. 40 FIRST WRITTEN THEN EXECUTED:  
26. 40 TYPE 2, 3, 5/6, 12^3, !!!!  
2. 0000 3. 0000 0. 8333 1728. 00

## SEMICOLON [; ]

SEPARATES COMMANDS WHEN PLACED TOGETHER ON ONE LINE.  
SEE WHAT THE ';' DOES TO LINE 27. 42 FIRST WRITTEN THEN EXECUTED:  
27. 42 SET X=17; TYPE 3\*X; SET X=3/7; TYPE " ", X+3, !!!!  
51. 0000 3. 4286

## FUNCTIONS

---

FOR MOST FUNCTIONS, A SAMPLE LINE IS SHOWN AND THEN EXECUTED:

'(A)' REPRESENTS THE ARGUMENT OF THE FUNCTIONS

NOTE: ARCTAN, LOG AND EXP FUNCTIONS ARE PRECISE TO LESS THAN 7 DIGITS.  
ARCTAN'S PRECISION IS MUCH REDUCED FOR ARGUMENTS NEAR ZERO.  
SINE AND COSINE ARE PRECISE TO 10 DIGITS.

FABS(A) ABSOLUTE VALUE  
EXAMPLE (WRITTEN THEN EXECUTED):  
03. 40 TYPE FABS(-3),FABS(2),!!!!  
3. 0000 2. 0000

FADC(A) ANALOG TO DIGITAL INPUT FUNCTION  
04. 22 SET X=FADC(A);TYPE !!!!

FATN(A) ARCTANGENT ARGUMENT RANGE:  $0 < A < 10^6$   
05. 25 TYPE FATN(1),!!!!;COM: OUTPUTS RADIAN ANGLE WHOSE TANGENT = 1. 00  
0. 7854

----- TANGENT 'FTAN(A)' IS NOT AVAILABLE  
TANGENT IS COMPUTED AS THE QUOTIENT (SINE/COSINE)  
05. 55 TYPE FSIN(1)/FCOS(1),!!!!;COMMENT: OUTPUTS TANGENT OF 1 RADIAN  
1. 5574

FCOS(A) COSINE ARGUMENT RANGE:  $-1 * 10^4 < A < 10^4$  (RADIANS)  
06. 27 TYPE FCOS(1),FCOS(3/2),!!!!;COM: OUTPUTS COSINES OF 1 & 1. 5 RADIANS  
0. 5403 0. 0707

FDIS(A) DISPLAY FUNCTION FOR DEC TYPE 340 DISPLAY  
EXAMPLE:  
07. 20 SET DUMMY=FDIS(X,Y)  
WOULD CAUSE A POINT TO BE PLOTTED AT POSITION (X,Y) ON DISPLAY

DISPLAY AND JOYSTICK FUNCTIONS FOR PS/8 FOCAL HAVE ALSO BEEN WRITTEN  
FOR THE TEKTRONIX T-4002 GRAPHICS TERMINAL. SEE PAGE 19.

FEXP(A) EXPONENTIAL  
08. 24 TYPE FEXP(1),FEXP(2. 17),!!!!;COM: NATURAL BASE TO THE POWER (A)  
2. 7183 8. 7583

## FITR(A) INTEGER

09.18 TYPE FITR(3/2),FITR(23.719),FITR(-2.99),!!!!  
 1.0000 23.0000- 2.0000

WARNING: LACK OF PRECISION IN FOCAL, 1969'S FLOATING POINT PACKAGE  
 MAY CAUSE FITR(A) TO PRODUCE ROUNDING ERRORS WITH NUMBERS OF  
 MORE THAN FOUR OR FIVE DIGITS. YOU MAY PROGRAM AROUND THIS  
 PROBLEM BY ADDING EMPIRICAL CONSTANTS TO THE FUNCTION ARGUMENT.

## FLOG(A) NATURAL LOGARITHM

10.14 TYPE FLOG(1),FLOG(4.237),FLOG(10),!!!!  
 0.0000 1.4439 2.3026

## FRAN() RANDOM NUMBER

11.22 TYPE FRAN(),FRAN(),FRAN(),FRAN(),FRAN(),!  
 0.5444 0.7732 0.3680 0.8852 0.8446

THE PSEUDO-RANDOM NUMBERS PRODUCED ARE PART OF A VERY LONG AND WELL  
 DISTRIBUTED BUT DETERMINISTIC SERIES. YOU WILL USUALLY OBSERVE AN  
 ENTIRELY DIFFERENT SERIES EACH TIME FOCAL IS INITIALIZED. HOWEVER,  
 FRAN() DOES NOT GENERATE TRULY RANDOM NUMBERS.

## FSGN(A) SIGN

12.34 T %1.0 FSGN(2.78), " ",FSGN(-299), " ",FSGN(0), " ",FSGN(-1),!!!!  
 1 -1 0 -1

FSIN(A) SINE ARGUMENT RANGE:  $-1 \times 10^4 < A < 10^4$  (RADIAN)

13.52 TYPE FSIN(1),FSIN(0),FSIN(37-2.22),!!!!  
 0.8415 0.0000- 0.2207

## FSQT(A) SQUARE ROOT OF A POSITIVE NUMBER

14.40 TYPE FSQT(4),FSQT(391),FSQT(.0038953),!!!!  
 2.0000 19.7737 0.0624

## FIN() CHARACTER INPUT FUNCTION

ASKS FOR A CHARACTER FROM THE TERMINAL (OR FILE); EXAMPLE:

09.14 SET A=FIN()

SETS 'A' EQUAL TO THE DECIMAL ASCII CODE OF THE CHARACTER INPUT.

## FOUT(A) CHARACTER OUTPUT FUNCTION

09.25 SET DUMMY=FOUT(A)

OUTPUTS THE CHARACTER WHOSE DECIMAL ASCII CODE EQUALS 'A'.

IF AN 'L' WERE INPUT IN RESPONSE TO LINE 9.14, 'A' WOULD BE SET  
 EQUAL TO 204. THEN LINE 9.25 WOULD CAUSE AN 'L' TO BE OUTPUT.  
 THE 'A' ON LINE 9.25 IS A DUMMY VARIABLE. SEE DECIMAL ASCII CODES  
 FOR FIN() AND FOUT(A) ON PAGE 28.

## DISPLAY AND JOYSTICK FUNCTIONS

---

FOR THE TEKTRONIX T-4002 GRAPHIC COMPUTER TERMINAL

### FDIS(I, X, Y) --- DISPLAY

COORDINATES:             $0 \leq X < 1024$              $0 \leq Y < 1024$

I=-1 PLOT A POINT AT (X, Y)  
 I= 0 DRAW A DARK VECTOR TO (X, Y)  
 I=+1 IF IN GRAPHICS MODE, DRAW A BRIGHT VECTOR TO (X, Y); OTHERWISE ENTER GRAPHICS MODE AND DRAW A DARK VECTOR TO (X, Y).

### FJOY(I) ----- JOYSTICK

FJOY(I) SETS VARIABLES XJ AND YJ EQUAL TO THE X AND Y LOCALATIONAL ADDRESSES IN THE INTERACTIVE GRAPHICS UNIT.

I=-1 SETS XJ AND YJ EQUAL TO LOCALATIONAL ADDRESSES.  
 I= 0 TURNS ON JOYSTICK. WAITS FOR KEYBOARD INPUT BEFORE READING CURSOR COORDINATES.  
 I=+1 ASSUMES JOYSTICK PREVIOUSLY TURNED ON BY A FOUT(17); READS CURSOR IMMEDIATELY.

### GRAPHICS FOUT(I) CODES

1 .. HOME  
 17 .. TURNS ON JOYSTICK FOR FJOY(1) APPLICATION  
 24 .. ERASE SCREEN (REQUIRES .5 SECOND DELAY)  
 30 .. ENTER INCREMENTAL PLOT MODE  
 31 .. LEAVE GRAPHICS MODE

### INCREMENTAL PLOT MODE

FOUT(I) CODES:

DIRECTION	N	NE	E	SE	S	SW	W	NW
BRIGHT POINT	32	33	34	35	36	37	38	39
DARK POINT	40	41	42	43	44	45	46	47

### PROGRAM: 'SKETCH'

(FOLLOWS JOYSTICK ON DISPLAY)

```
01. 10 SET I=0; FOR K=0, . 2, FOUT(24); C ERASE AND PAUSE
01. 20 SET DUMMY=FJOY(0); C WAITS FOR A KEY TO BE HIT
01. 30 SET DUMMY=FDIS(0, XHOLD, YHOLD)+FDIS(I, XJ, YJ)
01. 40 SET XHOLD=XJ; SET YHOLD=YJ; SET I=1; GOTO 1. 2
```

NOTE: THE VERSION OF PS/8 FOCAL USING FDIS AND FJOY FOR THE TEKTRONIX T-4002 PROVIDES 10%-20% FEWER VARIABLES.

## MISCELLANEOUS

---

EXPRESSIONS --- ARITHMETIC EXPRESSIONS INCLUDE ANY EVALUATABLE GROUP OF NUMBERS OR VARIABLES SUCH AS: A7, 2+3\*X OR 22/7

HIGH SPEED PAPER TAPE READER CAN READ IN FOCAL PROGRAMS:

- 1) TYPE 'OPEN INPUT PTR:'
- 2) AFTER '^' IS TYPED OUT, HIT SPACE BAR TO READ TAPE
- 3) AFTER '?' IS TYPED OUT, HIT 'CTRL/P'

USING YOUR TERMINAL'S LOW SPEED READER, YOU CAN INPUT PROGRAM TAPES BY FIRST TURNING OFF THE ECHO WITH 'OPEN INPUT TTY:,'; READING IN THE TAPE; AND THEN TURNING ON THE ECHO WITH CTRL/P OR 'OPEN INPUT TTY:,'E'.

THE FOLLOWING 3 LINE PROGRAM WILL ASK FOR 10 NUMBERS FROM THE READER:

```
10.60 OPEN INPUT PTR:
      (PS/8 WILL NOW OUTPUT A '^' ON THE TERMINAL AND WAIT FOR YOU
      TO SIGNIFY THAT THE TAPE IS LOADED BY HITTING THE SPACE BAR.)
10.65 FOR X=1,10) ASK A(X)
10.70 OPEN INPUT TTY:,'ECHO; COMMENT --- RESTORES TERMINAL TO NORMAL
```

HIGH SPEED PUNCH MAY BE USED TO PUNCH PROGRAM TAPES:

- 1) TYPE 'OPEN INPUT TTY:,'OPEN OUTPUT PTP:,'W'
- 2) THEN TYPE 'OUTPUT CLOSE;OPEN INPUT TTY:,'E'

LINE NUMBERS --- MAY RANGE FROM 1.01 THROUGH 31.99 BUT DO NOT INCLUDE WHOLE NUMBERS. VARIABLES (OR ARITHMETIC EXPRESSIONS NOT STARTING WITH 'A') OF THE CORRECT VALUE MAY BE USED IN PLACE OF LINE NUMBER REFERENCES (EXAMPLE: 'GOTO X'). GROUP NUMBERS ARE INTEGERS FROM 1 THROUGH 31 AND REFERENCE GROUPS OF LINES IN 'DO', 'WRITE', AND 'ERASE' COMMANDS.

MERGING PROGRAMS 'A' AND 'B' IS DONE WITH A SERIES OF COMMANDS WHICH CONVERT 'A' INTO A PS/8 EDITOR COMPATIBLE DATA FILE AND THEN BRING IT IN 'ON TOP' OF PROGRAM 'B':

- 1) LIBRARY CALL A
- 2) OPEN INPUT TTY:
- 3) OPEN OUTPUT A (STEPS 3-4 WILL NOT OUTPUT ON TERMINAL)
- 4) OUTPUT CLOSE;OPEN INPUT TTY:,'ECHO
- 5) LIBRARY CALL B
- 6) OPEN INPUT A (WAIT FOR '?' ON TERMINAL; THEN HIT 'CTRL/P')
- 7) LIBRARY DELETE A.FD

THE MERGED PROGRAM WILL NOW BE IN YOUR PROGRAM BUFFER.



PRECISION --- IF 10 DIGIT PRECISION IS NOT RETAINED THROUGH THE INITIAL DIALOGUE, YOU WILL HAVE 20% MORE VARIABLES BUT ONLY 6 DIGIT PRECISION.

PIP --- A PS/8 SYSTEM PROGRAM WHICH CAN LIST YOUR ENTIRE DEVICE DIRECTORY. TYPE 'R PIP' TO PS/8 MONITOR THEN 'DEVICE:/L' TO LIST DIRECTORY. WHEN TRANSFERRING FOCAL PROGRAM FILES WITH PIP, BE SURE TO USE THE '/I' OPTION AS THEY ARE CORE IMAGE FILES --- NOT ASCII!!!

PROGRAM NAMES --- MAY HAVE UP TO SIX ALPHABETIC OR NUMERIC CHARACTERS. FOCAL ASSUMES '.FC' OR '.FD' EXTENSIONS FOR PROGRAM AND DATA FILE NAMES.

PROGRAM SIZE --- WITH STANDARD FEATURES, PS/8 FOCAL PROGRAMS MAY BE 7 BLOCKS LONG (ABOUT 3500 CHARACTERS). IF FILES ARE DELETED THROUGH THE INITIAL DIALOGUE, 11 BLOCK PROGRAMS ARE POSSIBLE.

PUSHDOWN LIST OVERFLOW --- WHEN A PROGRAM USES A 'DO' OR 'LIBRARY GOSUB' COMMAND, FOCAL RECORDS THE LOCATION IN ITS PUSHDOWN LIST SO THAT IT CAN 'RETURN' TO THE NEXT COMMAND AFTER COMPLETING THE 'DO' SUBROUTINE. IF TOO MANY SUBROUTINES ARE REQUESTED, THE LIST WILL OVERFLOW. TO REMEDY THIS SITUATION, REDUCE THE NUMBER OF BLOCKS OR SUBROUTINES IN PROGRAM.

STRINGS --- ARE ANY SERIES OF CHARACTERS SUCH AS:

```
HELLO
122.5
$99.95
NOW IS THE TIME FOR ALL GOOD MEN ....
TEST34
```

THE ABOVE STRINGS WERE TYPED BY SURROUNDING THEM WITH QUOTATION MARKS AS IN LINE 22.25:  
22.25 TYPE " HELLO"!

STRINGS MAY ALSO BE PRINTED OUT CHARACTER BY CHARACTER USING THE 'FOUT(>)' FUNCTION; INPUT OF STRINGS MAY BE HANDLED WITH THE 'FIN(>)' FUNCTION. LIST THE INITIAL DIALOGUE FOR AN EXAMPLE.

THE ASK COMMAND MAY BE USED TO INPUT AND COMPARE 6 OR 10 CHARACTER STRINGS (DEPENDING ON PRECISION RETAINED IN INITIAL DIALOGUE). IT IS BEST TO RETAIN FEATURE '8' IN DIALOGUE WHEN USING THIS TECHNIQUE.

```
22.80 ASK "ANSWER YES OR NO? ", AN; COMMENT 'YES OR NO' SUBROUTINE
22.82 IF (AN-0YES)22.84, 22.86
22.84 IF (AN-0NO)22.8, 22.88, 22.8
22.86 SET X=2; RETURN
22.88 SET X=1; RETURN
```

```
*DO 22.8
ANSWER YES OR NO? YEP
ANSWER YES OR NO? NOPE
ANSWER YES OR NO? YES
```

PROGRAM CONTROL WOULD THEN RETURN TO THE COMMAND FOLLOWING THE SUBROUTINE CALL ('DO' OR 'LIBRARY GOSUB') WITH X EQUAL TO 2.

## VARIABLES

VARIABLES --- SOME EXAMPLES INCLUDE: A, X, Z7, P2, AN, QU, PI, NUMBER (ONLY THE FIRST TWO LETTERS WILL BE USED) --- VARIABLES MAY NOT BEGIN WITH AN 'F' OR A DIGIT.

SUBSCRIPTING --- VARIABLES MAY HAVE UNIQUE SUBSCRIPTS FROM -2048 THROUGH 2047 ALTHOUGH THE TOTAL NUMBER OF VARIABLES IS STILL LIMITED BY CHOICES MADE IN THE INITIAL DIALOGUE. NOTE THAT THE SYMBOL TABLE WILL ONLY ACCURATELY OUTPUT SUBSCRIPTS BETWEEN 0 AND 99.

PROTECTED VARIABLES --- '!', '" AND '#' CANNOT BE 'TYPED', 'ASKED' OR ZEROED WITH THE 'ERASE' COMMAND DIRECTLY BUT ARE OTHERWISE NORMAL VARIABLES AND MAY BE USED AS SUCH IN 'SET', 'IF', 'GOTO', 'FOR' AND OTHER COMMANDS. '#', THE SYSTEM STATUS VARIABLE, IS 'SET' TO SPECIAL CODES BY THE INITIAL DIALOGUE.

## INITIAL DIALOGUE

INITIAL DIALOGUE --- ASK FOCAL'S INITIAL DIALOGUE FOR 'HELP' WHICH WILL LIST OPTIONS OTHER THAN STANDARD FEATURES. BY NOT RETAINING FILES (FEATURE #4) PROGRAM SPACE IS INCREASED FROM ABOUT 3500 CHARACTERS (7 BLOCKS) UP TO ABOUT 5500 CHARACTERS (11 BLOCKS). IF FEATURES 5 AND 6 ARE BOTH REQUESTED, 6 WILL BE IGNORED.

THE SYSTEM STATUS VARIABLE, '#', LIKE '!' AND '"' CANNOT BE 'TYPED' 'ASKED' OR 'ERASED' BUT IS OTHERWISE A NORMAL VARIABLE. THE SYSTEM STATUS VARIABLE IS 'SET' TO THE PRODUCT OF THE CODES REPRESENTING FEATURES CHOSEN IN THE INITIAL DIALOGUE.

FEATURE	CODE
'0' NO OPTIONAL FEATURES .....	1
'1' FSIN AND FCOS FUNCTIONS .....	2
'2' FSIN, FCOS, FLOG, FEXP AND FATN .....	6
'3' EXTENDED PRECISION (10 DIGITS) .....	5
'4' FILE COMMANDS .....	7
'5' LEADING SPACE IN 'TYPE' .....	11
'6' '=' AUTOMATICALLY PRINTED BY 'TYPE' ..	13
'7' '::' AUTOMATICALLY PRINTED BY 'ASK' ..	17
'8' '&' IN POWERS-OF-TEN NOTATION .....	-1
'9' FEATURES 2, 3, 4 .....	210
'YES' STANDARD FEATURES 2, 3, 4, 5 .....	2310

THE NUMBER OF ORDINARY VARIABLES PERMITTED DEPENDS ON WHETHER EXTENDED PRECISION OR EXTENDED FUNCTIONS ARE RETAINED:

	NO EXTENDED FUNCTIONS	FSIN & FCOS ONLY (1)	FSIN, FCOS, FLOG FLOG, & FEXP (2)
6 DIGIT PRECISION:	222	196	118
10 DIGIT PRECISION:	185	163	98

## COMMAND EXECUTION TIMES

6 & 10 DIGIT PRECISION VERSIONS OF PS/8 FOCAL, 1971 WERE USED TO TEST THE COMMAND EXECUTION TIMES IN MILLISECONDS ON A PDP-8 USING A TU55 DECTAPE FOR INPUT AND HIGH SPEED TERMINAL FOR OUTPUT. TIMES INCREASE AS LINES ARE REFERENCED TOWARD THE END OF A PROGRAM AND VARIABLES ARE USED FROM THE END OF THE SYMBOL TABLE.

COMMAND	6 DIG	10 DIG	COMMAND	6 DIG	10 DIG
ASK X	13.42	13.57	S X=H-H	6.98	7.02
A X	13.08	13.09	S X=G*G	7.75	8.55
A X, X	25.69	25.96	S X=H*H	8.47	9.98
COMMENT	1.50	1.50	S X=G/G	8.71	10.34
C	0.51	0.51	S X=H/H	8.76	10.39
DO 1, 3	16.06	18.38	S X=444444/444444	15.48	17.12
D 1, 3	15.91	18.22	S X=444444/333333	15.56	17.35
FOR X=1, 1;	14.94	15.01	S X=1E10/1E-10	50.11	79.77
F X=1, 1;	14.60	14.67	S X=1E100/1E-100	357.57	657.90
F X=1, 2;	18.13	18.23	S X=FABS()	6.09	6.15
F X=1, 10;	50.27	50.63	S X=FABS(G)	7.31	7.37
GO 1, 3	14.43	16.74	S X=FABS(4)	9.59	9.66
G 1, 3	14.27	16.59	S X=FABS(H)	7.35	7.40
G Z	9.45	10.28	S X=FABS(444444)	10.69	10.74
IF (-1)1, 3, 1, 3, 1, 3	20.87	23.22	S X=FSQT(G)	22.79	31.77
I (-1)1, 3, 1, 3, 1, 3	20.69	23.04	S X=FSQT(4)	25.07	34.06
I (L)1, 3, 1, 3, 1, 3	16.78	19.10	S X=FSGN(G)	8.01	7.90
I (1)1, 3, 1, 3, 1, 3	20.47	22.79	S X=FITR(G)	9.29	9.34
I (N)1, 3, 1, 3, 1, 3	18.04	20.38	S X=FRAN()	7.43	7.50
I (-1)Z, Z, Z	15.85	16.70	S X=FEXP(G)	26.50	36.56
I (L)Z, Z, Z	11.93	12.77	S X=FSIN(G)	31.69	48.29
I (1)Z, Z, Z	15.04	15.88	S X=FCOS(G)	33.32	50.06
I (N)Z, Z, Z	12.63	13.46	S X=FATN(G)	25.09	37.10
RETURN	1.42	1.42	S X=FLOG(G)	14.78	48.25
R	.61	.61	S X=FIN()	6.24	6.30
S X=N	4.10	4.11	S X=FIN()+FIN()	11.29	11.38
S X=1	6.47	6.52	S X=[TEN FIN()^S]	65.37	65.60
S X=G	4.15	4.15	S X=FOUT(K)	9.19	9.26
S X=4	6.35	6.37	S X=FOUT(180)	11.91	11.98
S X=1+1	11.67	11.72	S X=FOUT(K)+FOUT(K)	17.20	17.29
S X=H	4.16	4.18	S X=[10 FOUT(K)^S]	82.46	82.88
S X=444444	7.43	7.46	TYPE	1.19	1.19
S X=G+G	6.87	6.91	T	.71	.71
S X=G+1	9.35	9.39	T !	1.32	1.34
S X=1+1	11.67	11.71	T #	1.38	1.38
S X=H+H	6.94	6.97	T "4"	1.46	1.46
S X=444444+H	10.38	10.47	T " 444444"	3.38	3.38
S X=444444+444444	13.69	13.72	T " 444444"!	4.06	4.06
S X=G+G+G	9.75	9.79	T " 444444", !	4.29	4.29
S X=G+(G+G)	11.51	11.59	T H	16.20	26.40
S X=G^J	26.80	34.47	T H, !	16.87	27.06
S X=4^J	29.15	36.82	T 1E25	91.20	163.49
S X=4^10	31.62	39.29	T 0001E5/0001E5	33.27	43.78
S X=4^100	185.37	261.78	T 100000/100000	18.20	20.83
S X=H+H	6.94	6.98	T "[20 SPACES]!"	7.76	7.77
S X=G-G	6.91	6.95	T :20, !	11.56	11.59

## TECHNICAL NOTES

---

### LOADING PS/8 FOCAL, 1971

PS/8 FOCAL, 1971 IS AVAILABLE IN THREE FORMS:

- 1) A PS/8 FORMAT FILE DECTAPE CONTAINING SAVED AND BINARY FILES OF PS/8 FOCAL AND THE 7 FILES NEEDED TO ASSEMBLE PS/8 FOCAL. THIS FORM IS RECOMMENDED FOR MULTIPLE DECTAPE INSTALLATIONS.
- 2) A PS/8 SYSTEM TAPE TO WHICH HAVE BEEN ADDED THE FILES NEEDED TO ASSEMBLE PS/8 FOCAL. THIS FORM IS RECOMMENDED FOR SINGLE DECTAPE INSTALLATIONS. NOTE: USER MUST SEND OMSI A PS/8 SYSTEM TAPE TO RECEIVE PS/8 FOCAL IN THIS FORM.
- 3) EIGHT PAPER TAPES; A BINARY TAPE OF PS/8 FOCAL AND 7 SOURCE TAPES.

FORMS 1) AND 2) ARE AVAILABLE WITH DEMONSTRATIONS AND THE 'HELP' PACKAGE.

TO LOAD PS/8 FOCAL FROM DECTAPE, MOUNT THE TAPE ON DRIVE 1, THEN GIVE THE FOLLOWING COMMANDS:

```
. ASSIGN DTA1: DSK:
. R PIP
*SYS:PFOCAL. SV<PFOCAL. SV/I
*CC
```

TO LOAD PS/8 FOCAL FROM PAPER TAPE, PLACE THE TAPE LABELED 'PFOCAL. BN' IN THE TAPE READER AND GIVE THE FOLLOWING COMMANDS:

```
. R ABSLDR
*PTR:=10200$(ALT MODE)
^ (TYPE A SPACE TO READ TAPE)
. SAVE SYS:PFOCAL
```

PS/8 FOCAL, 1971 IS NOW SAVED ON THE SYSTEM DEVICE AND MAY BE RUN BY GIVING THE PS/8 COMMAND:

```
. R PFOCAL
```

PS/8 FOCAL'S INITIAL DIALOGUE WILL THEN ASK YOU IF YOU WISH TO RETAIN STANDARD FEATURES. TYPE 'YES' OR REFER TO PAGE 22 OF THIS MANUAL.

## BUILDING PS/8 FOCAL

PS/8 FOCAL, 1971 CAN BE BUILT FROM FOUR ASCII SOURCE FILES AND THREE BINARY FILES. THE NECESSARY FILES ARE:

FOCDEF. PA DEFINITIONS OF CONSTANTS AND NEW INSTRUCTIONS FROM FOCAL, 1969.

FOCOVR. PA ADDITIONS AND OVERLAYS TO FOCAL PROPER (NOW RESIDENT IN FIELD 1).

FOCINT. PA INITIAL DIALOGUE (MACHINE LANGUAGE PORTION).

FOCLIB. PA LIBRARY AND FILE ROUTINES (RESIDENT IN FIELD 0).

FOCAL. BN BINARY OF FOCAL, 1969.

4WDMOD. BN FOUR-WORD FLOATING POINT PATCHES.

INITAL. BN INITIAL DIALOGUE (FOCAL PROGRAM THAT IS NOT REQUIRED IF INITIAL DIALOGUE IS NOT DESIRED). THIS FILE WAS PRODUCED BY CREATING A FOCAL PROGRAM AND THEN PUNCHING THE PROGRAM AREA IN BINARY.

TO BUILD PS/8 FOCAL, 1971, TRANSFER ALL THE NECESSARY FILES TO THE 'DSK:' AND GIVE THE FOLLOWING COMMANDS:

```
. R PAL8
*PFOCAL<FOCDEF, FOCOVR, FOCINT, FOCLIB
. R ABSLDR
*FOCAL, 4WDMOD, PFOCAL, INITAL/1=10200$(ALT MODE)
. SAVE SYS:PFOCAL
```

<'FOCINT' AND 'INITAL' MAY BE OMITTED>

'PFOCAL. SV' IS NOW SAVED ON THE SYSTEM DEVICE.

TO PREPARE VERSIONS PERMITTING THE USE OF DISPLAY AND JOYSTICK FUNCTIONS OF THE TEKTRONIX T-4002 GRAPHIC COMPUTER TERMINAL, INSERT FILE 'FDIS' (SERIAL INTERFACE 4804) OR '4801' (PARALLEL INTERFACE 4801) AFTER 'FOCDEF' IN THE COMMAND STRING GIVEN TO PAL8. THE T-4002 VERSIONS PROVIDE 10-20% FEWER VARIABLES.

## CHANGING PS/8 FOCAL

FOR ADDITIONAL INFORMATION, YOU MAY CONSULT 'ADVANCED FOCAL TECHNICAL SPECIFICATIONS' (ORDER NUMBER DEC-08-AJBB-DL FROM THE DEC PROGRAM LIBRARY, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS).

NEARLY ALL USER PATCHES TO FOCAL, 1969 CAN BE ADDED TO PS/8 FOCAL, 1971 WITH ONLY MINOR ADAPTATIONS. SOME POINTS WHICH YOU MAY WISH TO CHECK TO INSURE COMPATABILITY INCLUDE:

- 1) SOME FOCAL INTERNAL SUBROUTINES AND POINTERS HAVE BEEN CHANGED (E. G. 'POPA' IS NOW A SUBROUTINE CALL). PS/8 FOCAL ROUTINES WHICH PERFORM IDENTICAL FUNCTIONS ARE INCLUDED FOR ALL COMMONLY USED ROUTINES.
- 2) IN MOST CASES, THE 'HOLES' IN FOCAL, 1969 HAVE BEEN FILLED WITH PATCHES. A USER ROUTINE WHICH USES THESE LOCATIONS MUST BE RELOCATED. THE VARIABLE SYMBOL TABLE AREA IS RECOMMENDED FOR THIS PURPOSE (CHANGE 'STVAR' TO ALLOW FOR PATCHES).
- 3) USER-DEFINED FUNCTIONS REQUIRE POINTERS IN 'FNTABF' AND 'FNTABL'. THE 'FX' AND 'FNEW' SLOTS HAVE BEEN FILLED WITH 'FIN' AND 'FOUT'. RECOMMENDED FUNCTIONS TO DELETE FOR EXTRA SLOTS ARE FDIS, FADC AND FRAN DEPENDING ON THE NEEDS OF YOUR INSTALLATION.
- 4) FOCAL PROPER NOW RESIDES IN FIELD 1 (STARTING ADDRESS 10200). ANY PATCHES TO FOCAL MUST BE LOADED INTO FIELD 1 TO OVERLAY FOCAL. THE LIBRARY ROUTINES, PUSHDOWN STACK, AND TEXT STORAGE USE FIELD 0.

THE FOCOVR. PA SOURCE FILE CONTAINS ALL PATCHES MADE BY PS/8 FOCAL TO FOCAL PROPER AND SHOULD BE EXAMINED TO INSURE THERE ARE NO CONFLICTS WITH A USER PATCH.

SOME TERMINALS HAVE 'ESCAPE' OR 'PREFIX' IN PLACE OF 'ALT MODE'. TO CAUSE PS/8 FOCAL TO RECOGNIZE THE NON-ALT MODE, FIRST SAVE IT ON THE SYSTEM DEVICE. THEN GIVE THE PS/8 MONITOR THE FOLLOWING COMMANDS:

```
. GET SYS:PFOCAL
. ODT
17002/0375 233 (ESCAPE) OR 376 (PREFIX)
^C
. SAVE SYS:XFOCAL
```

IF PS/8 FOCAL IS TO BE USED IN THE PREPARATION OF NUMERICAL CONTROL TAPES, IT WILL BE NECESSARY TO MAKE A CHANGE SO THAT FOUT(141) WILL NOT AUTOMATICALLY OUTPUT BOTH A CARRIAGE RETURN AND LINE FEED. TO DO THIS FIRST SAVE PS/8 FOCAL ON THE SYSTEM DEVICE. THEN GIVE THE PS/8 MONITOR THE FOLLOWING COMMANDS:

```
. GET SYS:PFOCAL
. ODT
17541/4451 4463
^C
. SAVE SYS:NFOCAL
```

CHARACTERS OUTPUT WITH 'FOUT' WILL NOT CHANGE THE 'TAB' POSITION.

# INDEX FOR HELP

---

COMMANDS	SYMBOLS	FUNCTIONS
<b>ELEMENTARY COMMANDS</b> ----- ASK COMMENT CONTINUE DO ERASE FOR GO GOTO IF MODIFY QUIT RETURN SET TYPE WRITE  <b>LIBRARY COMMANDS</b> ----- LIBRARY CALL LIBRARY DELETE LIBRARY GOSUB LIBRARY LIST LIBRARY RUN LIBRARY SAVE LIBRARY EXIT  <b>FILE COMMANDS</b> ----- OPEN INPUT OPEN OUTPUT OPEN RESTORE INPUT OPEN RESTORE OUTPUT OUTPUT CLOSE	<b>ARITHMETIC SYMBOLS</b> ----- ^ EXPONENTIATION * MULTIPLICATION / DIVISION + ADDITION - SUBTRACTION ( ) PARENTHESES [ ] SQUARE BRACKETS < > ANGLE BRACKETS  <b>SPECIAL CHARACTERS</b> ----- ! RETURN/LINE FEED # CARRIAGE RETURN : TAB % OUTPUT FORMATTER \$ SYMBOL TABLE " QUOTATION MARKS ? TRACE E POWER-OF-TEN [&] RUBOUT KEY CTRL/C CTRL/G [BELL] CTRL/L CTRL/P LINE FEED KEY ← KILL LINE RETURN KEY ALT MODE KEY SPACE KEY , COMMA ; SEMICOLON	FABS( ) ABSOLUTE VALUE FADC( ) ANALOG-DIGITAL FATN( ) ARCTANGENT FCOS( ) COSINE FDIS( ) DISPLAY FEXP( ) EXPONENTIAL FITR( ) INTEGER FLOG( ) LOG (BASE E) FRAN( ) RANDOM NUMBER FSGN( ) SIGN PART FSIN( ) SINE FSQT( ) SQUARE ROOT FIN( ) CHARACTER INPUT FOUT( ) CHARACTER OUTPUT  <b>MISCELLANEOUS</b> ----- CHANGES FROM FOCAL, 1969 DEVICE NAMES DIRECT COMMANDS ERROR CODES EXPRESSIONS FILES HIGH SPEED READER INDIRECT COMMANDS INITIAL DIALOGUE LINE NUMBERS MERGING PROGRAMS NAMES OF FILES PRECISION PIP PUSHDOWN LIST OVERFLOW SAMPLE FILES PROGRAM SIZE OF PROGRAMS STRINGS SUBSCRIPTING SUMMARY OF COMMANDS TAB VARIABLES

THE HELP PACKAGE FOR PS/8 FOCAL, 1971 PERMITS THE ON-LINE REVIEW OF ANY OF THE ABOVE WORDS, PHRASES OR SYMBOLS. HELP IS AVAILABLE ON A DECTAPE WHICH SHOULD BE MOUNTED ON DRIVE #1 HAVING BEEN ASSIGNED THE NAME 'DSK:'. 'HELP' IS REQUESTED WITH THE 'L G HELP' COMMAND ("LET'S GET HELP") WHILE THE USER IS WRITING A PS/8 FOCAL PROGRAM. HE WILL BE RETURNED TO HIS PROGRAM (SAVED AS 'FOCAL.TM') WHEN HE HAS HAD ENOUGH 'HELP'.

DECIMAL ASCII CODES

FOR FINK > AND FOUT <

<u>CODE</u>	<u>CHARACTER</u>	<u>CODE</u>	<u>CHARACTER</u>	<u>CODE</u>	<u>CHARACTER</u>
128	CTRL/SHFT/P (LEADER)	160	SPACE	193	A
129	CTRL/A	161	!	194	B
130	CTRL/B	162	"	195	C
131	CTRL/C	163	#	196	D
132	CTRL/D	164	\$	197	E
133	CTRL/E	165	%	198	F
134	CTRL/F	166	&	199	G
135	CTRL/G (BELL)	167	'	200	H
136	CTRL/H	168	<	201	I
137	CTRL/I	169	>	202	J
138	LINE FEED	170	*	203	K
139	CTRL/K	171	+	204	L
140	CTRL/L	172	,	205	M
141	CARRIAGE RETURN	173	-	206	N
142	CTRL/N	174	.	207	O
143	CTRL/O	175	/	208	P
144	CTRL/P	176	0	209	Q
145	CTRL/Q	177	1	210	R
146	CTRL/R	178	2	211	S
147	CTRL/S	179	3	212	T
148	CTRL/T	180	4	213	U
149	CTRL/U	181	5	214	V
150	CTRL/V	182	6	215	W
151	CTRL/W	183	7	216	X
152	CTRL/X	184	8	217	Y
153	CTRL/Y	185	9	218	Z
154	CTRL/Z	186	:	219	[
155	CTRL/SHFT/K	187	;	220	\
156	CTRL/SHFT/L	188	<	221	]
157	CTRL/SHFT/M	189	=	222	^
158	CTRL/SHFT/N	190	>	223	←
159	CTRL/SHFT/O	191	?	253	ALT MODE
		192	@	255	RUBOUT

FOUT(141) WILL OUTPUT A RETURN/LINEFEED; FOUT(13) WILL OUTPUT A CARRIAGE RETURN ONLY. CODE 144 (CTRL/P) IS PS/8 FOCAL'S BREAK CHARACTER. 224 THROUGH 250 ARE LOWER-CASE LETTERS ON SOME TERMINALS. MANY TERMINALS USE SHIFT/K, /L, AND /M FOR '[', '\', AND ']'.

CODES 0 THROUGH 127 ARE SIMILAR TO 128-255 EXCEPT THAT THE HIGH ORDER BIT OF ITS BINARY REPRESENTATION IS EQUAL TO ZERO.



**PS/8 FOCAL, 1971 ERROR CODES**  
 -----  
 ERROR CODES VALID ONLY FOR 6/1/71 VERSION OF PS/8 FOCAL

**CODE EXPLANATION**

?00. 00	MANUAL START FROM CONSOLE
?01. 00	BREAK FROM KEYBOARD VIA 'CTRL/P'
?01. 40	BAD LINE NUMBER
?01. 91	GROUP NUMBER LARGER THAN 31
?01. ;3	ZERO IS ILLEGAL LINE NUMBER
?02. 32	'DO' REFERENCED NON-EXISTENT GROUP
?02. 52	'DO' REFERENCED NON-EXISTENT LINE
?03. 05	NON-EXISTENT LINE AFTER 'GOTO' OR 'IF'
?03. 28	ILLEGAL COMMAND
?04. 39	BAD VARIABLE IN 'FOR' OR 'SET' COMMAND
?04. 52	EXCESS RIGHT TERMINATORS
?04. 60	ILLEGAL TERMINATOR IN 'FOR' OR 'SET' COMMAND
?04. :3	MISSING ARGUMENT IN DISPLAY FUNCTION
?05. 48	BAD ARGUMENT IN 'MODIFY' COMMAND
?06. 06	ILLEGAL USE OF A FUNCTION OR NUMBER
?06. 54	TOO MANY VARIABLES
?07. 38	NO OPERATOR BEFORE PARENTHESIS
?07. :9	FUNCTION ARGUMENT MISSING
?07. ;6	ILLEGAL FUNCTION OR DOUBLE OPERATORS
?08. 47	ENCLOSURES DO NOT MATCH
?09. 11	BAD ARGUMENT IN 'ERASE' COMMAND
?10. :5	STORAGE FILLED BY TEXT
?11. 35	INPUT BUFFER OVERFLOW
?15. 28	BAD RESTORE COMMAND
?15. ;1	ATTEMPT TO WRITE PAST END-OF-FILE
?20. 34	LOGARITHM OF ZERO REQUESTED
?22. 23	NO OUTPUT FILE TO RESTORE
?22. 51	BAD FILES 'OPEN' COMMAND
?23. 05	DATA FILE NOT FOUND
?23. 15	NO INPUT FILE TO RESTORE
?23. 36	TOO MANY DIGITS IN NUMBER
?23. 39	PUSHDOWN OVERFLOW; PROGRAM TOO LONG
?24. 60	TWO PERIODS IN FILE NAME OR NO NAME
?25. 37	BAD DEVICE OR 2 PAGE HANDLER
?26. 18	BAD LIBRARY COMMAND
?26. 58	NO PROGRAM NAMED BY 'SAVE' COMMAND
?26. 99	EXPONENT TOO LARGE OR NEGATIVE
?27. 10	PROGRAM NOT FOUND
?27. 14	'LIBRARY CALL' FROM NON-DIRECTORY DEVICE
?28. 06	'LIBRARY LIST' OF NON-DIRECTORY DEVICE
?28. 73	DIVISION BY ZERO
?29. 53	FILE ALREADY DELETED
?29. 68	DEVICE ERROR: WRITE LOCK OR PARITY
?29. 91	CANNOT OPEN OUTPUT FILE
?30. 05	IMAGINARY SQUARE ROOTS REQUIRED
?31. <7	UNAVAILABLE COMMAND OR FUNCTION
?	ATTEMPT TO READ PAST END-OF-FILE (INPUT IS SWITCHED TO TERMINAL AND PROGRAM CONTINUES)

NOTE: TYPE =, TYPE >, TYPE ], OR TYPE > OUTPUTS A STRING OF ZEROES. THIS CAN BE STOPPED BY HITTING 'CTRL/P'.

SHOULD YOU EXPERIENCE A PS/8 MONITOR ERROR SUCH AS ATTEMPTING TO WRITE ON AN UNSELECTED DEVICE, IT IS BEST TO RE-ENTER FOCAL THROUGH THE INITIAL DIALOGUE.

**PS/8 FOCAL, 1971**  
**SUMMARY OF COMMANDS**

**\*\*\* ELEMENTARY COMMANDS :**

\*ASK X ..... ACCEPT VALUE OF X FROM INPUT DEVICE  
 \*COMMENT ..... IGNORE REST OF LINE  
 \*CONTINUE ..... DUMMY COMMAND  
 \*DO [LINE OR GROUP NUMBER] ..... SUBROUTINE CALL  
 \*ERASE ..... ZERO VARIABLES  
 \*ERASE <LINE OR GROUP NUMBER> ..... DELETE LINE OR GROUP  
 \*ERASE ALL ..... DELETE ENTIRE INDIRECT PROGRAM  
 \*FOR X=E1[, E2], E3; [COMMAND] ..... COMMAND REPEATED (E3-E1)/E2+1 TIMES  
 \*GO ..... START PROGRAM EXECUTION AT FIRST LINE  
 \*GOTO <L1> ..... SEND PROGRAM EXECUTION TO LINE L1  
 \*IF (E1)L1[, L2, L3]; ..... -, 0, OR + VALUE OF E1 GIVES NEXT LINE EXECUTED  
 \*MODIFY L1 ..... EDIT LINE L1  
 \*QUIT ..... STOP EXECUTION  
 \*RETURN ..... TERMINATE A 'DO' OR 'LIBRARY GOSUB' SUBROUTINE  
 \*SET X=<NUMBER, VARIABLE OR EXPRESSION> ..... ASSIGN VALUE TO VARIABLE  
 \*TYPE [NUMBERS, VARIABLES, EXPRESSIONS, TEXT IN QUOTES, !, #, :, \$ OR %]  
 \*WRITE [GROUP OR LINE NUMBER] ..... LIST PROGRAM

**\*\*\* LIBRARY COMMANDS :**

\*LIBRARY CALL [DEVICE:]<PROGRAM NAME> ..... LOAD SAVED PROGRAM  
 \*LIBRARY DELETE [DEVICE:]<PROGRAM NAME> ..... UNSAVE PROGRAM  
 \*LIBRARY GOSUB [DEVICE:]<PROGRAM NAME> [GROUP OR LINE NUMBER] ..... 'DO'  
 \*LIBRARY LIST [DEVICE:]<FIRST FILE TO BE LISTED> ..... CATALOG  
 \*LIBRARY RUN [DEVICE:]<PROGRAM NAME> [LINE NUMBER] ..... CHAIN  
 \*LIBRARY SAVE [DEVICE:]<PROGRAM NAME> ..... SAVE PROGRAM  
 \*LIBRARY EXIT ..... RETURN TO PS/8 MONITOR

**\*\*\* FILE COMMANDS :**

\*OPEN INPUT [DEVICE:]<FILE NAME>[, ECHO] ..... PREPARE TO READ FILE  
 \*OPEN OUTPUT [DEVICE:]<FILE NAME>[, ECHO] ..... PREPARE TO WRITE FILE  
 \*OPEN RESTORE INPUT[, ECHO] ..... RESUME INPUT FROM FILE  
 \*OPEN RESTORE OUTPUT[, ECHO] ..... RESUME OUTPUT TO FILE  
 \*OUTPUT CLOSE ..... OUTPUTS BUFFER; CLOSSES OUTPUT FILE  
 ---INPUT---, ECHO ..... INPUT IS ECHOED ON OUTPUT DEVICE  
 ---OUTPUT---, ECHO ..... OUTPUT IS ECHOED ON TERMINAL

**\*\*\* GENERAL FORM :**

⊙ ENCLOSE REQUIRED TERMS AND [ ] ENCLOSE OPTIONAL TERMS.  
 X IS A VARIABLE. E1, E2 AND E3 ARE ARITHMETIC EXPRESSIONS.  
 L1, L2, AND L3 ARE LINE NUMBERS. LINE AND GROUP NUMBERS MAY  
 BE REPLACED BY ANY EXPRESSION NOT BEGINNING WITH THE LETTER 'A'.  
 .FC AND .FD ARE ASSUMED PROGRAM AND DATA FILE NAME EXTENSIONS.

FOLLOW COMMANDS WITH A SPACE; ONE LETTER ABBREVIATIONS PERMITTED.

**CTRL/P IS PS/8 FOCAL'S BREAK CHARACTER**

## COMMENTED DIRECTORY FOR 9/11/71 DECTAPE

9/11/71

HELP . FC	6	CENTRAL PROGRAM IN THE PS/8 FOCAL 'HELP' PACKAGE
HELPER. RE	2	'HELP' PACKAGE
DIRECA. RD	6	'HELP' PACKAGE
DIRECI. RD	4	'HELP' PACKAGE
DIRECR. RD	4	'HELP' PACKAGE
SDIREC. RD	2	'HELP' PACKAGE
ERASE . FD	1	DATA FILE CALLED BY 'BTLSHP'
HANGND. FD	2	DATA FILE WITH WORDS FOR 'HANGNM'
LIBRAR. RE	6	'HELP' PACKAGE
HFILE1. RE	6	'HELP' PACKAGE
HFILE2. RE	6	'HELP' PACKAGE
COMMA1. RE	6	'HELP' PACKAGE
COMMA2. RE	4	'HELP' PACKAGE
COMMA3. RE	6	'HELP' PACKAGE
CATALOG. FC	3	CATALOG FOR PUBLIC LIBRARY OF PS/8 FOCAL PROGRAMS
INDEX . RE	6	'HELP' PACKAGE
HERROR. RE	6	'HELP' PACKAGE
FUNCT1. RE	6	'HELP' PACKAGE
FUNCT2. RE	5	'HELP' PACKAGE
SYMB01. RE	6	'HELP' PACKAGE
SYMB02. RE	6	'HELP' PACKAGE
SUMCOM. RE	6	'HELP' PACKAGE
MISCL1. RE	6	'HELP' PACKAGE
MISCL2. RE	6	'HELP' PACKAGE
MISCL3. RE	6	'HELP' PACKAGE
PFOCAL. SV	26	PS/8 FOCAL, 1971
TECO . SV	14	TECO-8 FOR PS/8
FOCAL . TM	1	TEMPORARY FOCAL PROGRAM SAVED BY 'LIBRARY GOSUB' COMMAND
<EMPTY>	115	'HOLE' FOR USER PROGRAMS
LISTAL. FD	4	A DATA FILE CONTAINING A SERIES OF DIRECT COMMANDS WHICH LIST PUBLIC LIBRARY PROGRAMS WHEN THE 'OPEN INPUT LISTAL' COMMAND IS GIVEN IN COMMAND MODE
LISTAL. RD	4	SAME AS LISTAL. FD BUT FOR THE 'HELP' PACKAGE
INTRST. FC	3	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
CHISQR. FC	5	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
STAT1 . FC	6	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
STAT2 . FC	5	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
ADDEX . FC	2	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
BARON . FC	6	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
TRIANG. FC	4	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
CALNDR. FC	3	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
HANGNM. FC	5	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
HANDI . FC	5	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
PLOTER. FC	5	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
FACTOR. FC	2	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
POLAR . FC	4	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
GRAPH . FC	5	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
HARMON. FC	4	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
WEB . FC	3	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY
BTLSHP. FC	7	PS/8 FOCAL PROGRAM ---- PUBLIC LIBRARY

```

HELP .RE 6 BACKUP FILE FOR 'HELP.FC'
TECO .PA 151 TECO-8 SOURCE FILE
TECO .BN 16 TECO-8 BINARY FILE (LOAD WITH ABSLDR) STARTS AT 200)
FOCDEF.PA 4 PS/8 FOCAL, 1971 SOURCE FILE --- DEFINITIONS/INSTRUCTIONS
FOCOVR.PA 28 PS/8 FOCAL, 1971 SOURCE FILE --- OVERLAYS TO FOCAL, 1969
FOCINT.PA 9 PS/8 FOCAL, 1971 SOURCE FILE --- INITIAL DIALOGUE (PAL8)
FOCLIB.PA 69 PS/8 FOCAL, 1971 SOURCE FILE --- LIBRARY AND FILE ROUTINES
FOCAL .BN 17 FOCAL, 1969 BINARY FILE
4WDMOD.BN 1 FOCAL FLOATING POINT PACKAGE FOUR-WORD PATCHES
INITAL.BN 7 PS/8 FOCAL, 1971 INITIAL DIALOGUE (FOCAL PROGRAM)
FDIS .PA 8 PS/8 FOCAL, 1971 SOURCE FILE --- T-4002 SERIAL INTERFACE
4801 .PA 8 PS/8 FOCAL, 1971 SOURCE FILE --- T-4002 PARALLEL INTERFACE
PFOCAL.BN 34 PS/8 FOCAL, 1971 BINARY FILE --- STARTS AT 10200
MAP .SV 5 CORE MAPPER --- SPECIFY BINARY FILE FOR COMMAND DECODER
SPY .SV 3 DECTAPE ODT --- VERY 'EXPERIMENTAL'
SV2BN .SV 3 'SAVE-TO-BINARY' CONVERTER
115 FREE BLOCKS

```

## NEW VERSION - - -

The July 10, 1972 version of PS/8 FOCAL adds the FIND(character) function and Variable Reallocation. Variable reallocation permits the user to add new variable names to his symbol table when it is full by first setting variables that are no longer needed, to be equal to zero. Trig function precision is now nearly uniform at 8 digits across the range of the functions as a result of suggestions by Gregory Ruth, Draper Lab, Cambridge, Mass. The number of non-zero variables permitted as a result of initial dialogue responses is slightly different from the chart on page 22 of the manual and is now:

	No Extended Functions	FSIN & FCOS Only (1)	FSIN, FCOS, FLOG, FEXP, & FATN (2)
	-----	-----	-----
6 Digit Precision:	217	196	118
10 Digit Precision:	181	163	98

## PS/8 FOCAL CHARACTER SEARCH FUNCTION

FIND(a) will input characters through the one whose decimal ASCII code is the function argument. If input is being echoed, all characters up to, but not including the search character will be echoed. FIND can search many times faster than is possible using FIN and IF. See the example below in which FIND is used with ECHO to copy a file eighty times faster than is possible using FIN, FOUT, and IF:

C-PS/8 FOCAL, 1971

```

01. 10 COMMENT: EXECUTION TIMES OF FILE COPYING ROUTINES COMPARING
01. 20 COMMENT: FIN() AND FOUT() FUNCTIONS WITH FIND() FUNCTION.
01. 30 COMMENT: IN EACH CASE, THE SEARCH CHARACTER IS CTRL/Z (CODE 154).
01. 40
01. 50 COMMENT: WHEN INPUT FILE IS ECHOING, FIND() ECHOES ALL CHARACTERS
01. 60 COMMENT: UP TO BUT NOT INCLUDING THE SEARCH CHARACTER.
01. 70
01. 80 COMMENT: TIMINGS WERE MADE BY COPYING A 10 BLOCK FILE FROM AND TO
01. 90 COMMENT: AN RF08 DISK USING A FDP-8 (1.5 MICROSECOND CYCLE TIME).

02. 10 COMMENT: COPY FILE1 INTO FILE2 --- FIN()/FOUT(): 156.5 SECONDS
02. 20 OPEN INPUT FILE1; OPEN OUTPUT FILE2
02. 30 ERASE
02. 40 SET Z=154; SET L=138; S N=2.6; S D=0; S P(138)=2.7; S P(154)=2.8
02. 50 GOTO 2.7
02. 60 S D=FOUT(C)
02. 70 S C=FIN(); I ((Z-C)*(C-L))/N; P(C), N
02. 75 COMMENT: LINE 2.7 SUPPRESSES EXTRA LINEFEED FROM FOUT(141)
02. 80 OUTPUT CLOSE; OPEN INPUT TTY.; ECHO
02. 90 TYPE "DONE"!

03. 10 COMMENT: COPY FILE1 INTO FILE2 --- FIND(): 1.95 SECONDS!!
03. 20 OPEN INPUT FILE1; ECHO; OPEN OUTPUT FILE2
03. 30 SET DUMMY=FIND(154)
03. 40 OUTPUT CLOSE; OPEN INPUT TTY.; ECHO
03. 50 TYPE "DONE"!

```

