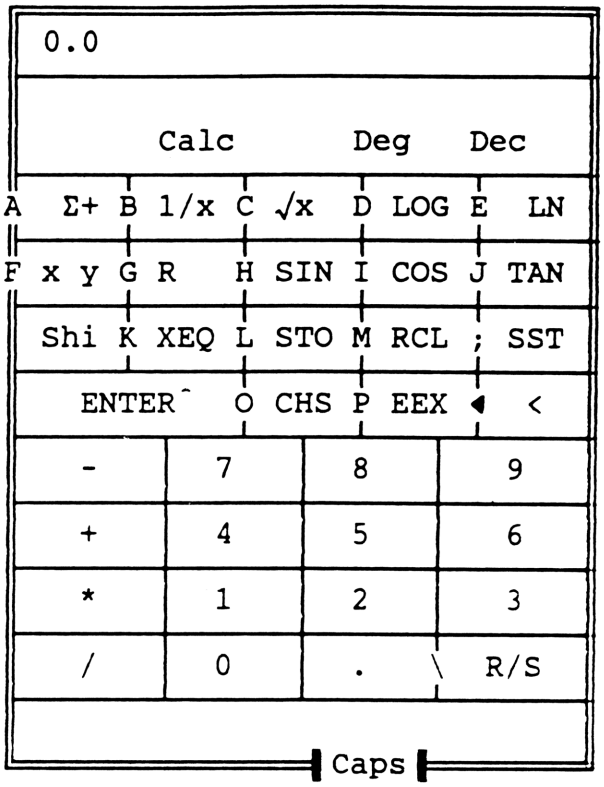
ELI-41

PROFESSIONAL SCIENTIFIC CALCULATOR SOFTWARE

ECLIPSE LOGIC INC.

ECLIPSE LOGIC INC.
P.O. BOX 2003
HUNTINGTON PARK, CA 90255



T A B L E O F C O N T E N T S

L I S T O F E X A M P L E S

L I S T O F T A B L E S

L I S T O F F I G U R E S

TERMS AND CONDITIONS

Eclipse Logic, Inc. makes no representations or warranties as to the merchantability or fitness of ELI-41 for any particular purpose or application. The purchaser is given a lifetime license for the use of ELI-41 and as such is permitted to make a copy for backup purposes only. Reproduction or copying of ELI-41 or it's documentation for any other purpose is a violation of Eclipse Logic's Copyright. In no event shall Eclipse be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential or other damages.

ACKNOWLEDGMENTS

In this manual, references are made to a number of trademarks:

IBM is a trademark of International Business Machines,
HP-41 is a trademark of Hewlett-Packard Company.

ECLIPSE LOGIC INC.
P.O. BOX 2003
HUNTINGTON PARK, CA 90255

OUR THANKS

We, at Eclipse, wish to thank those who have lent their support toward the completion of ELI-41. Many have assisted by phone conversation, beta testing and casual conversation with constructive criticism. The following persons had the patience to put up with our sometimes irritable behavior after working on the ELI-41 project for extended periods. Howard Carter, Joel Farley, Angela George, Gene Stauffer, Rudy & Ruth Wolf. We especially want to thank our families for the understanding. Also appreciation goes out to those who have written articles relating to PC's and high technology for giving us ideas and methods, and those individuals who developed the tools which we used for accomplishing our tasks.

INTRODUCTION

ELI-41 is an advanced scientific calculator program for use with IBM or compatible personal computers having a minimum of 192K RAM. The program simulates the Hewlett-Packard HP-41 calculator and supports all of the HP-41 functions and features. In addition, ELI-41 offers the following features which are not available with the HP-41:

1. Unique view of stack, flags and data registers.

2. Operates in Decimal, Binary, Octal or Hexadecimal mode and automatically converts between these three modes.

3. Unlimited number of program files.

4. Program files can be linked to create programs of virtually unlimited size.

5. Unlimited number of programmed conversion tables.

It is not the purpose of this manual to provide instruction on the basic HP-41 calculator operation, but to provide the necessary information for the proper use of this program. ELI-41 users that are not familiar with the HP-41 calculator should obtain one of the many texts available that cover the use and programming of Hewlett-Packard calculators. The READ.ME file on the distribution disk contains a partial list of texts available.

SYSTEM REQUIREMENTS

ELI-41 works on IBM PC or compatible and requires approximately 111k of memory after initialization routines are completed. The original requirement at start-up is 128K. Initializing routines set aside various work areas and release the balance of the original 128K for system usage. This is done to enable the user to run more than one program in memory, if the operating system permits. If the original required memory is not present at start-up an error message will be displayed and the calculator will refuse to load.

Also required are a Monochrome or Color monitor and card and at least one disk drive using either MS-DOS or PC-DOS 2.0 or higher operating system. A printer and 8087 math coprocessor are optional. At start-up, the calculator checks the machine's configuration, and writes to a color monitor if so configured; otherwise to a monochrome. The printer is assumed to be LPT1. A DOS redirection utility must be run previous to loading if such is not the case.

THE BASICS

The ELI-41 uses memory registers to store number entries and temporary results of calculations, as well as alphanumeric strings. The primary registers used in all calculations will be referred to as the Stack. This Stack uses four (4)

basic registers, which we will refer to as Stack-X, Stack-Y, Stack-Z and Stack-T. Another register that can be used which will contain the last entry in the Stack-X register from some calculations, we will refer to as the Last-X register. Another basic register to the calculator is the Alpha register which can contain only alphanumeric and special characters. It cannot be used for computations, but is used primarily for prompts within programs to prompt the user for a particular response.

The ELI-41 is an RPN type calculator. That is Reverse Polish Notation. This means you enter both of the numbers involved in a calculation before the actual function to be performed with these numbers. You enter these numbers in the same order as you would within an equation. For example,

4 * 3 = 12

In order for the calculator to know that the entry of the first number is ended and the second number is being entered you must terminate the first number's entry with the <ENTER> key. So the above example becomes:

4 <ENTER> 3 *

The result of 12 will appear in the Data Entry area as well as in the Stack-X register.

A correction to a number as it is being entered can be made by using the BACKSPACE key. This key will delete the last digit of the entry and remove it from the display. Continued pressing of this key will remove each digit until all digits are removed. Pressing the BACKSPACE key, with no entry having been made, will clear the display and set the Stack-X register to zero.

1. USING ELI-41

2. LOADING ELI-41

Before using ELI-41, make a copy of the distribution disk and use it as your working copy. Put the original ELI-41 disk in a safe place. You may want to create sub directories for the executable files, program files and conversion table files. On the distribution disk there is a file called INSTALL.BAT that does just that. You may want to modify it to fit your own system configuration. Refer to the DOS manual for information on creating directories and copying files.

There are possibly four other executable files on your distribution disk, depending on your purchase:

ELI41.EXE
ELI4187.EXE
ELI41P.EXE
ELI4187P.EXE

Additionally, the option file (ELI41.OPT) should be copied to the same directory as the executable files if you want to retain the current configuration settings. The calculator can create this file, and with its absence, will use default settings stored in the calculator itself. These settings tell the calculator screen attributes and areas searched for the necessary program and conversion files. But more on that later...

Throughout this manual we will address our comments toward the standard ELI41.EXE file with any variations applying to other versions noted. To run ELI41.EXE:

1. Change current directory to the directory where the executable program and the option file reside. This is necessary so the program will be able to find the option file.

2. Type ELI41 (or the other program file name) in response to the DOS prompt.

To exit ELI41 (from the calculator screen) hit Alt F1.

3. THE FILES USED

ELI-41 accesses several different kinds of files. The type of file is identified by the extension of the filename on a directory listing.

EXTENSION	CONTENTS
.PGM	Files with this extension are program files. These files are created by a word processor or

text editor. The contents of .PGM files are a series of ELI-41 command statements.

.OPT — Files with this extension specify the ELI-41 OPTION FILE. The option file contains information about the environment the calculator is working in, such as directory paths, colors, file names, etc. The complete filename of the loadable file is ELI41.OPT. The default option file supplied on the disk is configured for a monochrome screen with all files in drive A:. This one can be loaded with almost any terminal configuration. The screen attributes, file paths and other environmental functions can be modified on the option screen (Chapter7). Be sure to save the changes when they suit you.

NOTE: If you are using a color card driving a composite monitor rename ELI41.OPT to something else (to save it) and rename COMPOSIT.OPT to ELI41.OPT.

.MEM — Files with this extension are memory files. At anytime while using ELI-41, you may save memory (Stack, All Registers, Flags and User Assigned Keys) to disk. This feature is useful to save intermediate results. Memory files may be restored on subsequent sessions and the exact status of memory variables will be made available.

.CNV — Conversion table data is the information contained within files with this extension. Several conversion tables are provided with the ELI-41 system. You may also create your own conversion table file with a standard word processor or text editor.

4. FILES CONTAINED ON THE DISTRIBUTION DISK

FILE NAME	CONTENTS
----------	EXECUTEABLE PROGRAM FILES ----------
ELI41.EXE	This is the executable calculator program file.
ELI4187.EXE	This is the 8087 version of the calculator program.
ELI41P.EXE	This is the pop-up version of the calculator.
ELI4187P.EXE	This is the pop-up with 8087 support.
ELI41.OPT	A file containing the default options of the ELI-41 system.

---------- PROGRAM FILES ----------

EVENS.PGM	Program to demonstrate the use of ISG.
GTOTEST.PGM	Program to demonstrate the GOTO function.
ROOTS.PGM	Program calculates roots of quadratic formula.
MARY.PGM	Program that plays musical notes.
STORE.PGM	Program that demonstrates the use of STO.
TAX.PGM	Computes TAX based on the salary.
TONE1.PGM	Demonstrates chaining program execution.
TONE2.PGM	
TONE3.PGM	
TONE4.PGM	
TONE5.PGM	
TONE6.PGM	
TONE7.PGM	
TONE8.PGM	
TONE9.PGM	
XEQTEST.PGM	Demonstrates the XEQ function.
CONV.PGM	Example of how to use conversion tables.
RTFIND.PGM	Finds roots of a function entered by user.
WAVE.PGM	Example function used for RTFIND.

---------- CONVERSION TABLES ----------

AREA.CNV	Table for area conversions. (meters, inches, feet, acres, miles).
LENGTH.CNV	Table for length conversions. (centimeters, inches, feet, miles).
POWER.CNV	Table for power value conversions. (J/S, ft. lbs. per second, horsepower, kilowatts, BTU per hour).
VOLUME.CNV	Table for volume conversions. (centimeters, inches, feet, US gallons).

5. SCREEN DISPLAY LAYOUT

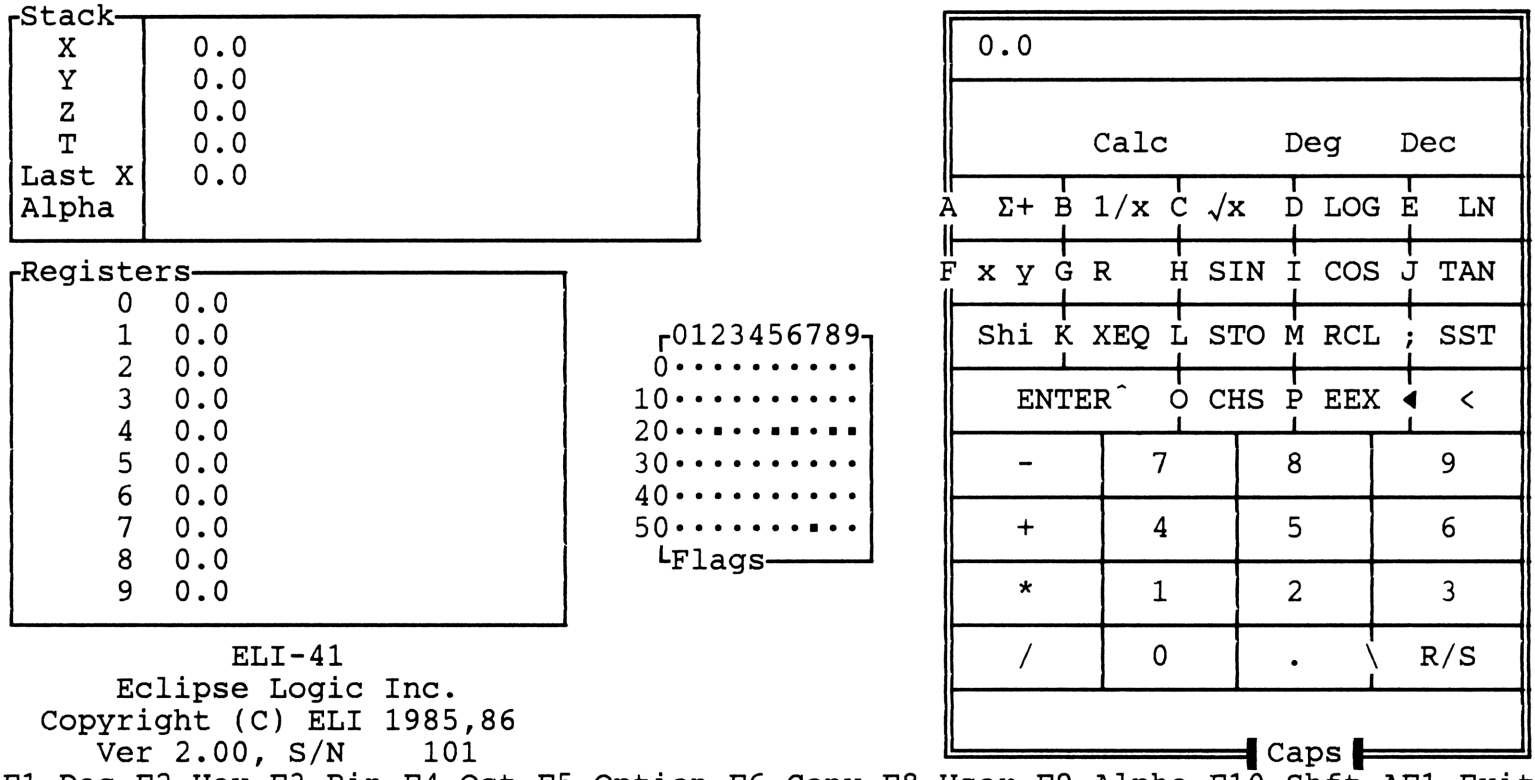


Figure: 2-1 CALCULATOR FACE

Refer to Figure: 2-1 for the following references regarding the calculator screen display. In the upper left-hand corner is the stack register and the alpha register area. A lightbar will highlight the last register used. Located below the stack register area is the data register area. Ten registers are visible at a time. Again, the last used register is highlighted by a lightbar. The data register area my be scrolled by using the cursor direction keys (with the NUM LOCK key off). The action of the cursor keys is as follows:

HOME - Display from register 0.

END - Display the last ten registers.

DOWN - Move toward the last register.

UP - Move toward the first register.

PgUp - Previous ten registers.

PgDn - Forward ten registers.

The flag status box is in the center of the screen and allows viewing of the sixty (60) internal flags. Those flags in the OFF state are displayed as a small dot. Flags that are in the ON state are displayed as a small square. Refer to Appendix A.1 for definition of flag settings.

The 25th line on the screen displays the function keys and their usage. Refer to Appendix A.2 for a full description of their use.

On the right hand side of the display is the calculator itself. The top line is what we will refer to as the Data entry area. It is used as in input area and as a display area. Error messages also appear in this area. On the second line in this area is where instructional messages are displayed. We will refer to this area as the Instruction Area. Below this area is the Annunciator Area. This area displays the current calculator mode settings, current angular mode and the numeric display type. The six possible calculator mode settings are: CALC, SHIFT CALC, USER, SHIFT USER, ALPHA and SHIFT ALPHA.

There are three types of angle measurements: degrees, radians and gradians. The display in the Annunciator Area will show the present angle measurement type: DEG, RAD or GRAD.

There are four base modes: decimal, hexadecimal, binary and octal. The Annunciator Area will display the current base as DEC, HEX, BIN or OCT.

Below the annunciator area is the main body of the calculator. You will notice that each one of the calculator "keys" contains a function, number or letter. To the left

side of each of these keys is the actual keypress character that you press to execute that function or input the number or letter. If there is no keypress character, simply press the character that is in the "key".

Each time you press a key, that key will be highlighted on the calculator face. This allows you to see what your last command was.

When the NUM LOCK function is active it is indicated by the word, "NUM" located on the bottom border of the calculator. By pressing your NUM LOCK key the NUM indicator disappears. "CAPS" acts likewise, and is located just to the left of the "NUM" display. A second press of either key will change the display and acts a toggle. A third indicator also resides in this area; this is the "PRT" indicator. This will indicate if the printer is currently activated to print actions in the calculator. It can be set on from the Options screen.

When you start ELI-41,the calculator comes up in CALC mode. This is the default mode. The function keys F8, F9, F10 allow you to change the current mode. These are USER mode, ALPHA mode and the SHIFT mode operators, respectively. Like the NUM LOCK key, they act as a toggle. For instance, if the calculator is currently in CALC mode and you want to change to SHIFT CALC mode. Simply press F10. The calculator Annunciator will change from CALC to SHIFT CALC. The contents of "keys" area will display SHIFT CALC type function keys and the keypress character availability will be for SHIFT CALC mode. The SHIFT modes all allow only one key function press and then will revert to the unshifted mode. But if you should want to return to CALC mode, without choosing a function, simply press the SHIFT key (F10) again.

While in CALC mode, you may switch to ALPHA mode by pressing F9. After making your entry in ALPHA mode, press the F9 key again and the mode will be switched back to the previous mode, CALC. While you are in ALPHA mode, you may press F10 which places the calculator in SHIFT ALPHA mode. By pressing F8, ALPHA mode is terminated and the current mode becomes USER mode. In USER mode the face of the calculator changes to display a two-up vertical table display of the available functions. Each function will be identified by an alphanumeric character. This function then may be executed by a single key stroke. Specific information on each of the calculator modes will be discussed in later chapters.

6. THE POPUP

The pop-up version of ELI-41 is loaded in the same manner as the basic calculator. Always load from a DOS prompt or a batch file (".bat"), never from a shell type program. ELI-41 may co-exist with other memory resident programs. Some memory resident programs must be loaded as the last memory resident program. ELI-41 has no restriction in this regard.

The difference from the basic calculator during program loading is a copyright notice, version and serial number is displayed on the screen. If the pop-up version has been loaded previously, a message will be displayed and normal operations will resume. In the event there is insufficient memory for ELI-41 to load, a message will be displayed and ELI-41 will not be installed into memory.

To activate ELI-41, hold the ALT key down and press the back quote key (the unshifted '~' key, located beneath the ']' key on standard IBM PC keyboards). This key combination was designed so as not to conflict with other programs.

Upon pressing the activating keypress, the calculator screen display will appear and its operation is the same as the basic calculator. The end execution keypress (ALT-F1) used in the basic calculator is also used in the pop-up version. Upon pressing ALT-F1, ELI-41 will terminate but remain memory resident and your screen will be restored to what it was prior to using the calculator. The next time the activating keypress is made the calculator will reappear with stack, registers, flags and program buffer in its last used state.

The pop-up version can be removed from memory. From a DOS prompt, press the CTRL key, the SHIFT key and the back quote (the unshifted '~' key) key simultaneously. All memory used by the pop-up will be released back to the system. Any memory resident programs loaded after ELI-41 must be removed prior to removing ELI-41. Failure to do so will cause a 'hole' in low memory which DOS cannot handle and your machine will lock up, requiring a re-boot of the system.

USING THE ELI-41 CALCULATOR

In contrast to hand-held scientific calculators, ELI-41's screen display affords high visibility of all pertinent information, easy keyboard entry and useful additional features.

NUMBER INPUT

There are four different number bases that you may use to input numbers into ELI-41. They are: DECIMAL, BINARY, OCTAL and HEXADECIMAL. The choice of number entry and display mode (within the calculator data entry area) is activated by pressing the appropriate function key. These are displayed on the bottom line of the screen. The display mode you currently have active is displayed on the calculator face as "Dec", "Bin", "Oct", or "Hex". You will notice the data will be presented in the calculator data entry area in the base that you have chosen. In addition, when the input is terminated, the decimal number is displayed in the STACK-X register of the display screen. Any time before an entry is terminated, it may be edited by use of the BACKSPACE key. A brief description of each entry mode follows.

DECIMAL (F1)

The DECIMAL entry mode is the default. When ELI-41 is first run DECIMAL entry mode is activated. The legitimate numeric characters continue entry. Any other terminates an entry. This means that you may only enter the following characters "0123456789.E". The 'E' is used for exponential factoring and is obtained by pressing the "EEX" key. A negative number is entered by keying in the number and pressing the "CHS" key.

NOTE: the "-" is not used for negative number entry but for performing subtraction only.

Valid Entries	
123.45	in Stack-X register
123.45	"CHS" (number is -123.45)
123.45E21	in Stack-X register
123.45E21	"CHS" (number is -123.45E21)

HEXADECIMAL (F2)

HEXADECIMAL entry mode allows input of numbers in base 16 and consists of the digits "0123456789abcdef". The largest number that can be input in this mode is 2,147,483,646 decimal (7FFFFFFE in Hexadecimal). If a number is larger, an "OUT OF RANGE" message is displayed, the gray BACKSPACE key must be pressed and the calculator automatically switches to the DECIMAL entry and display mode.

BINARY (F3)

BINARY entry mode allows input of numbers in base 2 and consists only of the digits zero and one. As soon as a key other than zero or one is pressed, the entry is terminated and performs the identified function for that key (if defined). The largest number that can be input in the BINARY entry mode is 67,108,863 decimal (11111111111111111111111111 in Binary). The smallest number is -67,108,863 decimal (10000000000000000000000000 001 in Binary). If a number is larger, an "OUT OF RANGE" message is displayed, the "gray BACKSPACE" key must be pressed and the calculator automatically switches to the DECIMAL entry and display mode.

OCTAL (F4)

OCTAL entry mode allows input of numbers in base 8 and consists of the digits zero through seven. Any other key pressed terminates the entry and performs the identified function for that key (if defined). The largest number that can be input in this mode is 2,147,483,646 decimal (17777777776 in Octal). If a number is larger, an "OUT OF RANGE" message is displayed, the "gray BACKSPACE" key must be pressed and the calculator automatically switches to the DECIMAL entry and display mode.

NUMBER DISPLAY

The contents of the Stack and Register areas are always displayed in decimal notation (with non-significant zero suppression) at full precision unless the number is too large to be displayed in the allocated area or exceeds precision. In this case the number is displayed in SCIENTIFIC notation. The calculator input area follows the above rule also except when you have changed to a different base.

DECIMAL NOTATION

This is the standard number display format. In the calculator input area, comma editing and the suppression of nonsignificant zeroes is performed. It does not display any exponents unless the number is either too large or too small for the allocated area. Upon initialization of the ELI-41 system, the default of 9 decimal positions is set. To change the number of displayed decimal places, change to "SHIFT CALC" mode and press the "FIX" key. You are then prompted for the number of decimal places. This number must be between zero and 12. This changes only the display format of the calculator input area. If Flag 29 is on, editing with commas is performed. If comma editing is not

desired set Flag 29 off.

SCIENTIFIC NOTATION

The scientific notation displays one digit to the left of the decimal point and up to a three digit exponent. The number is rounded to the specified number of decimal positions. To change the number of displayed decimal places, change to "SHIFT CALC" mode and press the "SCI" key. You are then prompted for the number of decimal positions. This number must be between zero and 12. This changes only the display format of the calculator data entry area.

ENGINEERING NOTATION

The engineering notation displays between one and three digits to the left of the decimal point and up to a three digit exponent. The number is rounded to the specified number of decimal positions. To change the number of displayed decimal places, change to "SHIFT CALC" mode and press the "ENG" key. You are then prompted for the number of decimal positions. This number must be between zero and 12. This changes only the display format of the calculator input area.

The following example assumes 4 decimal places was specified.

DECIMAL	SCIENTIFIC	ENGINEERING
123.4569	1.2345E+02	123.4500E+00
1234.5679	1.2345E+03	1.2345E+03
0.1234567	1.2346E-01	123.4567E-03

THE ELI-41'S KEYS

In this chapter we will describe the various functions implemented in the calculator. Some of the functions can be executed by more than one method. We will also attempt to show those functions that are intimately related to the particular function being discussed.

Some functions will enable stack lift, which means that the next number entered will push the stack registers and the number being entered will overwrite the present contents of the Stack-X register. Those functions enabling the stack lift will be noted. Some functions will store the contents of the Stack-X register in the Last-X register, thus saving it, before the function is performed. This action will also be noted for any function that performs it.

USER mode is not outlined here as it is a duplicate of CALC mode until the user assigns other functions to USER mode keys. The same is true of SHIFT USER mode. Any available function in the calculator and any available program in the program directory can be assigned to any key in the USER or SHIFT USER mode. See the **ASN** key in SHIFT CALC mode section for further details.

This chapter is organized by calculator mode then by keypress character. The index in the last pages of this manual will be indispensable when using this chapter as a reference manual. A quick reference table of all the available functions and the calculator modes to which they are assigned is presented at the end of this chapter. The table attempts to condense the following information into a quick reference format.

THE CALC MODE

KEY	A (Σ+)
NAME	SUM PLUS
STACK	Disables Stack-lift
LAST-X	Precomputation Stack-X
FUNCTION	

The SUM PLUS function is a statistical function that uses six (6) memory registers called the "Statistical Block". The default beginning register of this block (stat_reg) is the memory register 11, unless set prior to the use of this function by the dREG function.

This function assumes two entries are being made. The "x" entry is in the Stack-X register and the "y" entry is in the Stack-Y register. Performing this function will accumulate the registers mentioned into the statistical block leaving the number in the Stack-Y register (the "y" component), and displaying

the number of the current entry sets in the data entry area. This number will also be in the Stack-X register.

The following formulas reflection the actions taken upon the statistical block:

Reg(stat_reg) = ΣX
Reg(stat_reg+1) = ΣX^2
Reg(stat_reg+2) = ΣY
Reg(stat_reg+3) = ΣY^2
Reg(stat_reg+4) = ΣXY
Reg(stat_reg+5) = number of accumulated entries

EXAMPLE

See Example 4-1.

RELATED FUNCTIONS Σ-, ΣREG, CLΣ, MEAN, SDEV

KEY B (**1/x**)
NAME RECIPROCAL
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The RECIPROCAL function takes the contents of the Stack-X register divides it into 1 and returns the result in the Stack-X register replacing its former contents.

EXAMPLE

KEYSTROKES	DATA ENTRY DISPLAY	COMMENTS
5	5_	NUMBER
B [1/X]	0.2	RECIPROCAL OF 5

PROGRAMMING EXAMPLE

```
LEXMPL2                 ; ARBITRARY LABEL
N5                      ; PLACES 5 IN STACK-X
S1/X                    ; COMPUTE RECIPROCAL OF 5
SEND
```

RELATED FUNCTIONS None

KEY C (**√x**)
NAME SQUARE ROOT

STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The SQUARE ROOT function takes the contents of the data entry area (the Stack-X register) and returns the square root of that number in the Stack-X register. Only positive numbers are allowed.

EXAMPLE

See Example 4-2.

RELATED FUNCTIONS X^2

KEY D (**LOG**)
NAME LOGARITHM
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The LOGARITHM function takes the contents of the Stack-X register and returns the Base 10 log of that number in the Stack-X register.

EXAMPLE

See Example 4-3.

RELATED FUNCTIONS 10^x, E^x, LN, E^{X-1}, LN1+X

KEY E (**LN**)
NAME NATURAL LOGARITHM
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The NATURAL LOG function takes the contents of the Stack-X register and returns the Base e (2.718...) log of that number in the Stack-X register.

EXAMPLE

See Example 4-3.

RELATED FUNCTIONS 10^x, E^x, LOG, E^{X-1}, LN1+X

KEY F **(X<>Y)**
NAME EXCHANGE X WITH Y
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The EXCHANGE X WITH Y function swaps the contents of the Stack-X and the Stack-Y registers.

RELATED FUNCTIONS X<> nn, RDN, R↑

KEY G **(R↓)**
NAME ROLL DOWN STACK
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The ROLL DOWN STACK function moves the contents of the various stack registers downward through the registers, that is, the contents of the Stack-X register is moved to the Stack-T register, the Stack-T register is moved to the Stack-Z register, etc.

RELATED FUNCTIONS R↑, X<> nn, X<>Y

KEY H **(SIN)**
NAME SINE
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The SINE function returns the sine of the number stored in the Stack-X register. The number is assumed to be measured in degrees, radians or gradians according to the present angle mode.

The number is also assumed to be in decimal format. A conversion can be taken of degree measurement that is in the form dd.mmss (dd: degrees, mm: minutes, ss: seconds) prior to using the sine function. In this case the function to use is DMS, which is performed by the XEQ function.

EXAMPLE

See Example 4-4.

RELATED FUNCTIONS COS, TAN, ASINE, ACOS, ATAN

KEY I (**COS**)
NAME COSINE
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The COSINE function parallels the SINE function in its usage except that it returns the cosine of the number entered. See the SINE function above.

EXAMPLE

See Example 4-4.

RELATED FUNCTIONS SIN, TAN, ASINE, ACOS, ATAN

KEY J (**TAN**)
NAME TANGENT
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The TANGENT function performs in the same manner as the SINE function (above) except that it returns the tangent of the Stack-X register.

EXAMPLE

See Example 4-4.

RELATED FUNCTIONS SIN, COS, ASINE, ACOS, ATAN

KEY K (**XEQ**)
NAME EXECUTE
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The EXECUTE function is a key that executes other functions and programs. This function will need additional parameters. The data entry area will display **XEQ** __. The calculator face will automatically switch to ALPHA mode to take a program or function name, composed of up to seven alphanumeric characters. Program file names must be limited to six characters or less, and must begin with an alpha character.

The two underscores signify that a two digit label can be entered. The entry of a digit will switch the

display to CALC mode to take the final digit. If there is no available label matching the numeric label entered, an error message will display and the instruction area will give details on proceeding.

A search is made of all the available functions to the calculator in the following order:

1. Available keystroke functions that could be executed by a keypress function in CALC, SHIFT CALC, and SHIFT ALPHA modes.

2. Internal functions to the calculator. Refer to internal functions section of this chapter for the available functions.

3. Any program file residing in the PROGRAM directory by the same name as that name entered (without the file extension).

In the event that the function or program is not found according to the above rules, an error message will be displayed. The error message display can only be cleared by pressing the BACKSPACE key, this action is indicated in the instruction area.

All the available internal functions can be displayed from the OPTIONS screen or CATALOG 3. All the available program files in the program directory can be viewed by the CATALOG 2 function discussed in the SHIFT CALC mode section.

EXAMPLE

KEYSTROKES	DATA ENTRY DISPLAY	COMMENTS
K [XEQ]	XEQ __	NOTE THE AUTOMATIC SHIFT
MARY	XEQ MARY_	INTO ALPHA MODE!
F9 [ALPHA]		
OR		
[ENTER KEY]	Executing Program!	MARY PROGRAM IS EXECUTED

PROGRAMMING EXAMPLE

```
LEXMPL11          ; ARBITRARY LABEL
XMARY             ; DENOTES XEQ MARY
AMARY             ; PLACES MARY IN ALPHA REGISTER
TASTO 00          ; STORES MARY IN REGISTER 00
XIND 00           ; DENOTES XEQ IND 00
AMARY             ; PLACES MARY IN ALPHA REGISTER
TASTO STK T       ; STORES MARY IN STACK-T
XIND STK T        ; INDIRECTLY EXECUTES MARY THROUGH STACK-T
SEND
```

RELATED FUNCTIONS GTO, BST, SST, CATALOG

KEY	L (**STO**)
NAME	STORE
STACK	Enables Stack-lift
LAST-X	Not applicable
FUNCTION	

The STORE function is used to store intermediate results in memory registers. It takes additional parameters. The data entry area will display **STO __**; the two underscores signify the need for a two digit entry. A list of additional acceptable characters and options follows:

1. A modifying character (+-*/) which signifies an addition, subtraction, multiplication or division to the register number to be entered. The data entry area will display **STO + __** in the case of the '+' key entry. The underscores signifying further entry is needed.

2. The F10 (shift key) signifies an indirect move. The display will be updated to display **STO IND __**, still signifying more entry is needed. If a two digit number is entered, the integer contents of the named register will point to the memory register which will receive a copy of the contents of the Stack-X register.

3. A dot (period) signifies a stack move. The Stack-X register can be copied to any of the other stack registers directly. The display is updated to **STO STK _**. The single underscore specifies one of the stack register characters (X,Y,Z,T or L) is needed.

Any or all of these special modifications to the basic STORE function can be used but only in the order that they are presented here. Examples follow:

STO + IND STK Y would add the Stack-X register to the register pointed to by the Stack-Y register.
STO + IND 02 would add the Stack-X register to the register number pointed to by the contents of the memory register 02.

Note: in a program use: **ST+ IND STK Y**
and: **ST+ IND 02**
for the above examples.

Any entry made out of place will not trigger an error response with the exception of the stack character

entry. The illegitimate key will simply be ignored. Pressing the BACKSPACE key will terminate the function with no action being taken but to redisplay the contents of the Stack-X register.

EXAMPLE

See Example 4-5.

RELATED FUNCTIONS RCL, ST+, ST-, ST*, ST/

KEY M **(RCL)**
NAME RECALL nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The RECALL function is the reverse of the STORE function discussed above. The register nn is copied into the Stack-X register and all of the options noted with STORE are available with RECALL, except for (+-/*). If Stack-lift was enabled previously, it will push the Stack and then continue with its normal operation.

EXAMPLE

See Example 4-5.

RELATED FUNCTIONS STO, ST+, ST-, ST*, ST/

KEY ; **(SST)**
NAME SINGLE STEP
STACK Not applicable
LAST-X Not applicable
FUNCTION

The primary purpose of the SINGLE STEP function is in the debugging of programs. A program must be loaded into the internal program buffer using either the COPY or the XEQ function. Note however, that the execution of an END statement in a program will clear the internal buffer; therefore, the **SST** function must be used before the END statement is reached.

The instruction area will display the executed program line. Continued pressing of the **SST** key will progressively step through the program until the END statement is reached. The current line number is displayed, but any comments within the file will not

be displayed.

Any errors found during execution will display a pertinent error message and the line number of the statement in error.

RELATED FUNCTIONS BST, XEQ, GTO, COPY

KEY <ENTER> or N (**ENTER↑**)
NAME ENTER
STACK Disables Stack-lift
LAST-X Not applicable
FUNCTION

The function of the ENTER key is to push the contents of the stack, each register into the next register with the Stack-X register being duplicated in the Stack-Y register. This key terminates the entry of a number into the Stack-X register. Thus, any number entry made after this function will overwrite the contents of the Stack-X register.

RELATED FUNCTIONS RDN, R↑, X<>Y, X<> nn

KEY O (**CHS**)
NAME CHANGE SIGN
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The CHANGE SIGN function can be used for three different purposes:

1. At any point after the entry of a digit into the data entry area, the number currently there can have a sign change; negative to positive, or positive to negative.

2. After any operation that terminated a digit entry, the present contents of the Stack-X register can be switched from negative to positive, or from positive to negative.

3. This is the only facility for entering a negative exponential value. This must be used anytime before terminating the entry of the exponent and anytime after calling the ADD EXPONENT function (below).

The function acts as a toggle so a number can be switched to the opposite sign.

EXAMPLE

See Example 4-6.

RELATED FUNCTIONS EEX

KEY P (**EEX**)
NAME ADD EXPONENT
STACK Not applicable
LAST-X Not applicable
FUNCTION

The ADD EXPONENT function enables the adding of an exponent to an Engineering notation type number. The current number in the data entry area will have an **E** appended to the number and an underscore (cursor) will appear. The calculator then awaits further input. The exponent portion of a number can be a three digit number but is limited to a maximum of 300.

To add a negative number, first enter the exponent and then press the **CHS** function key.

EXAMPLE

See Example 4-6.

RELATED FUNCTIONS CHS

KEY BACKSPACE (◀)
NAME BACKSPACE
STACK Disables Stack-lift with no previous entries
LAST-X Not applicable
FUNCTION

The BACKSPACE key is used for editing the current number or alpha entry in the data entry area. Repeated pressing of this key will remove the last entry character from the display area until the display is cleared; at which time, the Stack-X register will be cleared to zero.

RELATED FUNCTIONS None

KEY ✎ (**R/S**)
NAME RUN/STOP
STACK Enables Stack-lift

LAST-X FUNCTION Not applicable

The RUN/STOP function is used to perform one of the two functions as the name implies. The continuation of a program that was halted because of a PROMPT instruction or a halt by the previous use of the RUN/STOP key. If a program was being run in SINGLE STEP mode, this key will change the mode to execution mode.

RELATED FUNCTIONS GTO, SST, BST

THE SHIFT CALC MODE

KEY NAME: A (Σ-)
STACK: SUM MINUS
LAST-X: Disables Stack-lift
FUNCTION: Precomputation Stack-X

The SUM MINUS function acts to reverse the action of the SUM PLUS function from the CALC mode. It removes the Stack-X and Stack-Y elements from the accumulated statistical register block.

EXAMPLE

See Example 4-1.

RELATED FUNCTIONS Σ+, ΣREG, CLΣ, MEAN, SDEV

KEY NAME: B (y^x)
STACK: Y TO POWER OF X
LAST-X: Enables Stack-lift, Stack popped
FUNCTION: Precomputation Stack-X

The Y TO POWER OF X function raises the contents of the Stack-Y register to the power of the Stack-X register. The result is placed in the Stack-X register.

RELATED FUNCTIONS X^2

KEY NAME: C (x^2)
STACK: X SQUARED
LAST-X: Enables Stack-lift
FUNCTION: Precomputation Stack-X

The X SQUARED function multiplies the contents of the Stack-X register by itself. The Stack-X register is replaced by the result.

RELATED FUNCTIONS $\sqrt{x}$, y^x

KEY NAME: D (10^x)
STACK: COMMON EXPONENTIAL
Enables Stack-lift

LAST-X FUNCTION — Precomputation Stack-X

The COMMON EXPONENTIAL returns in the Stack-X register the value of 10 raised to the power of the Stack-X register. This is the reverse action of the COMMON LOGARITHM.

RELATED FUNCTIONS — LOG, LN, e^x, E©X-1, LN1+X

KEY — E (**ex**)
NAME — NATURAL EXPONENTIAL
STACK — Enables Stack-lift
LAST-X FUNCTION — Precomputation Stack-X

The NATURAL EXPONENTIAL function returns in the Stack-X register the exponential of the Stack-X register. This is the reverse of the NATURAL LOGARITHM.

RELATED FUNCTIONS — LOG, LN, 10^x, LN1+X, E©X-1

KEY — F (**CLΣ**)
NAME — CLEAR SUM
STACK — Enables Stack-lift
LAST-X FUNCTION — Not applicable

The CLEAR SUM function clears the registers presently being used as the statistical block of registers. The default block starts at memory register 11. It may be set to any other set of 6 registers by the SUMREG function (internal functions).

RELATED FUNCTIONS — **Σ-**, **Σ+**, **Σ**REG, CL**Σ**, MEAN, SDEV

KEY — G (**%**)
NAME — PERCENT
STACK — Enables Stack-lift
LAST-X FUNCTION — Precomputation Stack-X

The PERCENT function works with the Stack-X and the Stack-Y register. This function uses the Stack-X register as a percent and returns in the Stack-X register the percent of the Stack-Y register. The

contents of the Stack-Y register is unchanged.

RELATED FUNCTIONS %CH

KEY H **(ASIN)**
NAME ARCSINE
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The ARCSINE is the inverse of the SINE function. It uses the contents of the Stack-X register as the sine of an angle and returns in Stack-X the corresponding angle in the current angle measurement. That is, if the DEGrees annunciator is showing the Stack-X will represent decimal degrees. If the RADians annunciator was present then the measurement will be in radians.

NOTE: The decimal degrees measurement can be converted to degrees minutes seconds with the HMS function.

RELATED FUNCTIONS SIN, COS, TAN, ACOS, ATAN

KEY I **(ACOS)**
NAME ARCCOSINE
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The ARCCOSINE is the inverse of the COSINE function. It uses the contents of the Stack-X register as the cosine of an angle and returns in Stack-X the corresponding angle in the current angle measurement. That is, if the DEGrees annunciator is showing the Stack-X will represent the angle in decimal degrees. The same would apply with the RADian or GRADian annunciators.

RELATED FUNCTIONS SIN, COS, TAN, ASIN, ATAN

KEY J **(ATAN)**
NAME ARCTANGENT
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

The ARCTANGENT function is the inverse of TANGENT function. It uses the contents of the Stack-X register as the tangent of an angle and returns in Stack-X the corresponding angle in the current angle measurement. That is, if the DEGrees annunciator is showing the Stack-X will represent decimal degrees. The same would apply to the RADian or GRADian annunciator.

RELATED FUNCTIONS SIN, COS, TAN, ASIN, ACOS

KEY	K **(ASN)**
NAME	ASSIGN
STACK	Enables Stack-lift
LAST-X	Not applicable
FUNCTION	

The ASSIGN function is used to assign a particular function or program name to a specific key to be used in USER mode. Any function named in the calculator can be assigned, even internal functions otherwise only available through the use of the XEQ or the CATALOG function. Program file names in the program directory can also be assigned to keys.

When the ASSIGN key is pressed the calculator will shift to ALPHA mode to receive the function name (six characters maximum). The data entry area will display **ASN** __ requesting alphanumeric characters. After entering the last character, the entry is terminated by the <ENTER> key or the F9 (ALPHA mode) key.

As an example let us assume you were going to assign the internal function **CLST** to the **A** key of USER mode. When you entered CLST and pressed the <ENTER> key, the calculator would then switch to the user mode and the display will be updated to **ASN CLST _**. The underscore signifying the need for the key to be reassigned. The entering of the character **A** would terminate the assignment, assigning the key strokes entered to the supplied key, and revert to the CALC mode. As with any other function, the execution of the function returns the calculator to the unshifted mode.

Similar to the preceding example, suppose you want to assign the **CLST** function to the **A** key, but in SHIFT USER mode. When the display contained **ASN CLST** _ rather then pressing the **A** immediately you would press the SHIFT actuator key F10. The display is updated **ASN CLST _**, moving the underscore over one position signifying acceptance of your

intention. The display would also switch to SHIFT USER mode to accept the character to receive the assignment. As before the calculator will return to CALC mode.

RELATED FUNCTIONS None

KEY L **(LBL)**
NAME LABEL
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

For the sake of completeness the LABEL key is shown on the face of the calculator. At this time it has no function and will inform you of that fact if you try to use the key.

RELATED FUNCTIONS None

KEY M **(GTO nn)**
NAME GOTO
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The GOTO function key is used in program debugging. A program must be currently residing in the program buffer for this key to function. If that is not the fact an error message will inform you. If you have loaded a program prior to its use, **GTO __** will be displayed. The two underscores ask for a program numeric label. If one exists in the program, that entry will set an internal pointer to that label and the SST key could be used for single stepping the program from that point. The entry of the RUN/STOP key would begin execution at that point.

Alphanumeric labels can also be used in programs; therefore, the entry of an alphanumeric label in the program will set the internal pointer to that label if found.

Another option is entering a period in response to the two underscore prompt. Upon entering this period the display will be updated to **GTO ___**. The three underscores signify that a program line number (300 maximum) must be entered. Upon entering this line number, the internal pointer will be set to it. NOTE: line number 1 should be entered as 001.

RELATED FUNCTIONS XEQ, BST, SST

KEY ; **(BST)**
NAME BACK STEP
STACK Not applicable
LAST-X Not applicable
FUNCTION

FUNCTION

This function has not been implemented at this time. It is present for the sake of completeness in displaying the standard HP-41 calculator face.

RELATED FUNCTIONS SST, R/S, XEQ, GTO

KEY N **(CATALOG n)**
NAME CATALOG n
STACK Not applicable
LAST-X Not applicable
FUNCTION

The CATALOG function is used to view the various functions of the calculator, program files or conversion files and optionally to run any of these.

Pressing the CATALOG key will result in the data entry area displaying **CATALOG** _. The single underscore represents a number from 1 to 5. An invalid entry will bring the standard error response (escapable with the BACKSPACE key). The entry of a digit will result in changing the calculator face to a two column display of the various functions related to the entered catalog number. The valid number entries are as follows:

1. Program files.
2. Program files.
3. Functions internal to the calculator.
4. Memory dump files.
5. Conversion files.

An entry of one of the above numbers will display in the column mode the available functions or files.

The lightbar will overlay the first entry and this bar can be moved by the used of the cursor positioning keys: switch columns with the right and left arrow key, up and down with the corresponding arrow key, PgUp and PgDn perform their assigned function, and the Home and End keys give the beginning or ending page.

Pressing the <ENTER> key with the lightbar on a function or program file executes that function or program. The ESCAPE key simply clears the display and returns to the previous display.

RELATED FUNCTIONS None

KEY O **(ISG nn)**
NAME INCREMENT AND SKIP IF GREATER
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

This function is a programming function that has no use in command mode. Its use is similar to the DSE function (an internal function). This is a looping type function. It uses the register, supplied by the program line, which contains a mask. This mask enables the use of the register as a incrementing counter. The mask is of the format: aaaaa.bbbcc. The cc portion serves as a incrementing step function, in that, when this instruction is executed the present value of aaaaa is incremented by the step cc and the present value of aaaaa is compared to bbb. If aaaaa is greater than or equal to bbb the next sequential program line is skipped; otherwise, the next instruction is executed.

If the cc part of the mask is blank, the calculator assumes you want to increment by 1. If it is supplied, it must contain two digits.

If the bbb part of the mask is blank, the calculator assumes that you want to stop incrementing at zero. This part of the mask must contain three digits, if it is supplied.

To put it in a formula:

if (aaaaa + cc) >= bbb skip next instruction.

In a program, the register containing the mask must be supplied in ISG nn; where, nn is the register containing the mask.

RELATED FUNCTIONS DSE

KEY P (**RTN**)
NAME RETURN
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The RETURN function is a programming function. Pressing this key will display the end of program message. If a program is resident from a prior loading, with SST or XEQ, and the program has not reached its end point because of a halt with the RUN/STOP key, this key will reset the program to its beginning.

RELATED FUNCTIONS XEQ, GTO, R/S

KEY BACK SPACE (**CLX**)
NAME CLEAR STACK-X
STACK Disables Stack-lift
LAST-X Not applicable
FUNCTION

The CLEAR STACK-X function will clear the contents of the Stack-X register.

RELATED FUNCTIONS CL , CLP, CLRG

KEY Q (**X=Y?**)
NAME IS X EQUAL TO Y?
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The IS X EQUAL TO Y? function is a programming function. Within a program, the next sequential instruction is skipped unless the Stack-X register equals the Stack-Y.

RELATED FUNCTIONS X<=Y?, X>Y?, X=0?, etc.

KEY R (**SF nn**)
NAME SET FLAG
STACK Enables Stack-lift
LAST-X Not applicable

FUNCTION

The SET FLAG function will set the flag which is supplied to the prompt by the user to ON state. This is primarily a program function but may be useful in other instances, such as turning the printer on. The data entry area will display **SF** __ and the calculator will wait for a two digit flag number.

Certain flags are reserved for the user and are refered to in Appendix A.1. These flags can be tested within programs and specific actions can be taken accordingly.

RELATED FUNCTIONS CF, FC?, FC?C, FS?, FS?C

KEY S (**CF nn**)
NAME CLEAR FLAG
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

FUNCTION

The CLEAR FLAG function performs the opposite action to that of the SET FLAG function. That is, the specified flag will be turned OFF. The display is **CF** __ and waits for a two digit flag number. This is also a programming function. It could also be used for turning the printer off without going to the OPTION screen.

RELATED FUNCTIONS SF, FC?, FC?C, FS?, FS?C

KEY T (**FS? nn**)
NAME FLAG SET? nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

This function tests the condition of a specific flag. It is primarily a program function. It skips the next sequential program step unless the tested flag is on.

RELATED FUNCTIONS CF, FC?, FC?C, SF, FS?C

KEY U **(X Y?)**
NAME X LESS THAN OR EQUAL TO Y?
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

This function tests whether or not the Stack-X register is less than or equal to the Stack-Y register. Its primary use is within a program.

RELATED FUNCTIONS X>=Y?, X>Y?, X=0?, etc.

KEY V **(BEEP)**
NAME BEEP
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The BEEP key will sound a descending scale of notes. This is primarily useful in a program, but you could use it to test if your speaker is working.

RELATED FUNCTIONS TONE

KEY W **(P-R)**
NAME POLAR TO RECTANGULAR
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

This function assumes the Stack-X register contains a length value and the Stack-Y register contains an angle in the currently displayed angle annunciator type. The return from this function is into both the Stack-X and Stack-Y registers. The Stack-X register will be the horizontal or 'X' direction on a Cartesian plane and the Stack-Y will be in the vertical or 'Y' direction.

RELATED FUNCTIONS R-P

KEY X **(R-P)**
NAME RECTANGULAR TO POLAR
STACK Enables Stack-lift
LAST-X Precomputation Stack-X
FUNCTION

FUNCTION

This function assumes the Stack-X register contains the length measurement in the 'X' direction on a Cartesian plane and the Stack-Y in the 'Y' direction. After the execution of the function the Stack-X register will contain the resultant length in previous length measurement, and the Stack-Y register will contain the polar angle.

RELATED FUNCTIONS P-R

KEY Y (x<y?)
NAME X < Y?
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

This function is primarily used in programs. This function would cause program execution to skip the next sequential instruction unless the number in the Stack-X register was less than the Stack-Y. In command mode, a message will appear saying yes or no.

RELATED FUNCTIONS X<=Y?, X>Y?, X=0?, etc.

KEY Z **(FIX nn)**
NAME FIX DIGITS nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The function of the FIX DIGITS is to fix the number of significant decimal digits that are displayable in the data entry area. The Stack and memory register displays are unaffected by this command. This function can use indirect addressing to set the digits count. When used **FIX** __ appears in the display and requests two decimal digits. Acceptable digits are 0 through 13. If you should choose to use the indirect register method, use the shift key (F10) and **FIX IND** __ will be updated to the data entry area. The two digits requested refer to a memory register which contains the fixing length. Only the acceptable digits will set properly any other will have no affect.

RELATED FUNCTIONS ENG, SCI

KEY = **(SCI nn)**
NAME SCIENTIFIC NOTATION nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

This function will set the display in the data entry area to use the scientific notation to display the Stack-X register. As with the FIX function, acceptable digits are 0 through 13; any others will have no effect. The indirect addressing method may be used and the memory register must contain an acceptable digit entry.

RELATED FUNCTIONS ENG, FIX

KEY ? **(ENG nn)**
NAME ENGINEERING NOTATION nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

As with the two previous display modifiers, the significant decimal digit display is set. This notation, in addition, will set the number of the exponent display and the decimal point to reflect exponents divisible by 3. Acceptable digits are 0 through 13.

RELATED FUNCTIONS FIX, SCI

KEY : **(x=0?)**
NAME X = 0 ?
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

This is a programming function. Within a program, the next sequential instruction is skipped unless the Stack-X register is equal to zero. In command mode, a message will display to inform you of the fact.

RELATED FUNCTIONS X<=Y?, X>Y?, X=0?, etc.

KEY SPACEBAR **(c)**
NAME PI
STACK Enables Stack-lift, Stack is pushed
LAST-X Not applicable
FUNCTION

The use of this key will enter the value of PI into the Stack-X register. The Stack-X register will display 3.14159265359 but the number is stored internally in sixteen (16) significant digits.

RELATED FUNCTIONS None

KEY COMMA **(LASTx)**
NAME LAST-X
STACK Enables Stack-lift, the Stack is pushed
LAST-X Not applicable
FUNCTION

The purpose of this function is to restore the previous value of the Stack-X register to that register. Stack lift is enabled before as well as after this function so the value of the Stack-X register is saved into the Stack-Y register before the Stack-X register is updated.

RELATED FUNCTIONS None

KEY **\ (VIEW nn)**
NAME VIEW REGISTER nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

FUNCTION

The VIEW REGISTER function is used to redisplay the current memory registers to show the selected memory register. The displayed prompt **VIEW __** requests a two digit memory register number. Upon entry of this register number, the registers surrounding the selected register will be displayed. The lightbar will highlight the chosen register. If the requested register is visible in the memory register display at the time, the lightbar will simply be moved to highlight that register.

NOTE: Since this uses a two digit memory register prompt, the highest direct register viewing is limited to register 99. To view a higher register number (up to 500), indirect register usage is necessary. The particular register number to be viewed must be installed in a directly addressable register (up to 99). Then the entry would be used as follows: **VIEW IND nn.** The **IND**, of course, is gotten by the entry of the F10 key.

RELATED FUNCTIONS ARROW KEYS (with NUM LOCK off)

THE SHIFT ALPHA KEYS

KEY K **(APP)**
NAME APPEND TO ALPHA
STACK Not applicable
LAST-X Not applicable
FUNCTION

The APPEND TO ALPHA function allows addition to the ALPHA register after the last character. As with all SHIFT modes, after this keypress the calculator will revert to ALPHA mode, and be ready for character entry. Shifting back and forth between ALPHA and SHIFT ALPHA mode can be done without terminating the ALPHA entry. Pressing <ENTER> or F10 will terminate character entry.

The length of the ALPHA register is twenty-six (26) characters, any additional characters will shift the register left one character, accept the last character and beep to notify you that string truncation has occurred.

RELATED FUNCTIONS None

KEY L **(ASTO nn)**
NAME ALPHA STORE nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The ALPHA STORE function will copy the first six (6) characters of the ALPHA register in the memory register supplied by the user. The prompt displayed in the data entry area is **ASTO __**, and the calculator waits for the memory register number. Indirection can also be used; in fact, it is necessary to access registers numbered above 99.

Use of the **ASHF** function (available as an internal function) will allow the user to store the complete alpha string in sequential registers. The ALPHA SHIFT function will shift the first six characters out of the alpha register so the next six characters can be used.

RELATED FUNCTIONS ASHF, ARCL

KEY M **(ARCL nn)**
NAME ALPHA RECALL nn

STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The ALPHA RECALL function copies the contents of the register supplied by the user into the Alpha register. The display changes to **ARCL __**, and waits for the entry of a two digit memory register number. Upon entry of the second digit, the contents of the memory register is copied into the ALPHA register. If the contents of the memory register is not alphanumeric they will be converted to alphanumeric characters using the current FIX, ENG, SCI settings.

RELATED FUNCTIONS ASTO

KEY BACKSPACE **(CLA)**
NAME CLEAR ALPHA
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The CLEAR ALPHA function clears the present contents of the ALPHA register replacing it with a null string.

RELATED FUNCTIONS CLST, CLX, CLP, CLRG, CL

KEY **\ (AVIEW nn)**
NAME AVIEW nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The AVIEW function has no effect in the SHIFT ALPHA mode because the ALPHA register is being viewed continuously. It is primarily a programming function.

RELATED FUNCTIONS VIEW

INTERNAL FUNCTIONS

KEY	Use CATALOG 3 or **XEQ ABS**
NAME	ABSOLUTE
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
FUNCTION	

The ABSOLUTE function removes the negative sign of the Stack-X register. Necessary to SQUARE ROOT function and others.

RELATED FUNCTIONS None

KEY	Use CATALOG 3 or **XEQ ADV**
NAME	ADVANCE PAPER
STACK	Enables Stack-lift
LAST-X	Not applicable
FUNCTION	

The ADVANCE PAPER function will advance the paper one line if the printer flag (21) is set on. It has no effect otherwise.

RELATED FUNCTIONS None

KEY	Use CATALOG 3 or **XEQ AND**
NAME	AND
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
FUNCTION	

The AND function ANDs the value of the Stack-Y register against the value in the Stack-X register. The operation is a boolean operation. This is mainly useful to programmers using a preset mask in the Stack-Y register.

RELATED FUNCTIONS XOR, OR, NOT, <<, >>

KEY	Use CATALOG 3 or **XEQ AOFF**
NAME	ALPHA OFF
STACK	Enables Stack-lift
LAST-X	Not applicable
FUNCTION	

The AOFF function turns ALPHA mode OFF. This is mainly used in programs to switch modes after a ALPHA prompt function. Flag 48 (ALPHA mode) is turned OFF.

RELATED FUNCTIONS AON

KEY Use CATALOG 3 or **XEQ AON**
NAME ALPHA ON
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The AON function turns ON the ALPHA mode (flag 48). Useful in programs to take ALPHA input into ALPHA register during program execution.

RELATED FUNCTIONS AOFF

KEY Use CATALOG 3 or **XEQ ASHF**
NAME ALPHA SHIFT
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The ALPHA SHIFT function shifts the Alpha register left six (6) characters. The first six (6) characters are thus shifted out of the Alpha register. This is useful when storing the contents of the Alpha register in other registers. All the other registers can only contain six characters.

RELATED FUNCTIONS ASTO

KEY Use CATALOG 3 or **XEQ BIN**
NAME BINARY
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The BINARY function will change the present mode to allow for binary input and display. The acceptable entry digits in this mode are 0 and 1. Refer to Chapter 3 for further information.

RELATED FUNCTIONS HEX, DEC, OCT

KEY — Use CATALOG 3 or **XEQ CLD**
NAME — CLEAR DISPLAY
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

CLEAR DISPLAY has no useful function at this time since the display is immediately rewritten after the execution of any function.

RELATED FUNCTIONS None

KEY — Use CATALOG 3 or **XEQ CLP**
NAME — CLEAR PROGRAM
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

CLEAR PROGRAM clears the program presently in program buffer. Since only one program can reside in the program buffer at a time, this function has limited use.

RELATED FUNCTIONS CLA, CLST, CLRG, CL

KEY — Use CATALOG 3 or **XEQ CLRG**
NAME — CLEAR REGISTERS
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

CLEAR REGISTERS will clear all of the memory registers to zero. It does not affect the flags or the Stack.

RELATED FUNCTIONS CLA, CLST, CLP, CL

KEY — Use CATALOG 3 or **XEQ CLST**
NAME — CLEAR STACK
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The CLEAR STACK function clears all of the Stack registers except for the Alpha register.

RELATED FUNCTIONS CLA, CLP, CLRG, CL

KEY Use CATALOG 3 or **XEQ DATE**
NAME DATE
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The DATE function will display the current system date and time in the data entry area. Press the BACKSPACE key to clear the display and continue.

RELATED FUNCTIONS None

KEY Use CATALOG 3 or **XEQ DEC** or **F1**
NAME DECIMAL
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The DECIMAL function switches the data entry and display mode to the default of decimal format. This is useful for ending display modes of binary, octal or hexadecimal and take the display back to the default mode.

RELATED FUNCTIONS BIN, HEX, OCT

KEY Use CATALOG 3 or **XEQ DEG**
NAME DEGREES
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The DEGREE function will reset the angular measurement back to the default of degree mode. This is useful for restoring the default mode after using radian and gradian angular modes.

RELATED FUNCTIONS RAD, GRAD

KEY Use CATALOG 3 or **XEQ DEL**
NAME DELETE FROM LINE
STACK Not applicable
LAST-X Not applicable

FUNCTION

The DELETE FROM LINE function would delete program lines beginning at line entered. This function is not implemented at this time.

RELATED FUNCTIONS None

KEY Use CATALOG 3 or **XEQ DSE nn**
NAME DECREMENT AND SKIP IF EQUAL nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

DECREMENT AND SKIP IF EQUAL function is a program function that is dependent on a particular register containing a mask. This mask enables the use of the register as a decrementing counter. The mask is of the format aaaaa.bbbcc. The cc portion serves as a step function which when executed, the present value of aaaaa is decremented by the step cc and the present value of aaaaa is compared to bbb. If aaaaa is less than or equal to bbb the next sequential program line is skipped; otherwise, the next instruction is executed.

To put it in a formula:
if (aaaaa - cc) <= bbb skip next instruction.

In a program, the register containing the mask must be supplied in DSE nn; where, nn is the register containing the mask.

RELATED FUNCTIONS ISG nn

KEY Use CATALOG 3 or **XEQ D-R**
NAME DEGREES to RADIANS
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The DEGREES to RADIANS function converts the Stack-X register from degrees to radians.

RELATED FUNCTIONS R-D

KEY Program function
NAME END

STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The END statement is a necessary to end any program or subroutine. Its function is to take the calculator out of program mode.

RELATED FUNCTIONS RTN

KEY Use CATALOG 3 or **XEQ E©X-1**
NAME E©X-1
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

This function is used for natural logarithms of numbers whose value is close to zero.

RELATED FUNCTIONS LN1+X, LN, LOG, e^x, 10^x

KEY Use CATALOG 3 or **XEQ FACT**
NAME FACTORIAL
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

This function returns in Stack-X the factorial of the number in the Stack-X register.

RELATED FUNCTIONS None

KEY Use CATALOG 3 or **XEQ FC? nn**
NAME FLAG CLEAR? nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

FLAG CLEAR? is a program function that tests the particular flag specified and skips the next sequential program line unless the flag is clear.

RELATED FUNCTIONS FC?C, FS?, FS?C

KEY Use CATALOG 3 or **XEQ FC?C nn**

NAME FLAG CLEAR? CLEAR nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

FLAG CLEAR? CLEAR is a program function that tests the condition of a specified flag, clears it, and skips the next sequential instruction unless the prior condition of the flag was clear.

RELATED FUNCTIONS FC?, FS?, FS?C

KEY Use CATALOG 3 or **XEQ FRC**
NAME FRACTION
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The FRACTION function returns in the Stack-X register the fractional (decimal) portion of the Stack-X register.

RELATED FUNCTIONS INT

KEY Use CATALOG 3 or **XEQ FS?C nn**
NAME FLAG SET? CLEAR nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

FLAG SET? CLEAR is a programming function which tests the condition of a particular flag, clears it, and then skips the next sequential program line unless the previous condition of the flag was ON.

RELATED FUNCTIONS FS?, FC?, FC?C

KEY Use CATALOG 3 or **XEQ GRAD**
NAME GRADIANS
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The GRADIANS function changes the present angular measurement mode to gradians.

RELATED FUNCTIONS RAD, DEG

KEY	Use CATALOG 3 or **XEQ HEX** or **F2**
NAME	HEXADECIMAL
STACK	Enables Stack-lift
LAST-X	Not applicable
FUNCTION	

The HEXADECIMAL function switches the data entry and display mode to hexadecimal format. The acceptable input characters are 0 - 9 and A - F. Functions whose key is within these limits are disabled until the calculator is switched to another input mode. Programs in execution are not affected by this limitation.

RELATED FUNCTIONS DEC, BIN, OCT

KEY	Use CATALOG 3 or **XEQ HMS**
NAME	HOURS MINUTES SECONDS
STACK	Enables Stack-lift
LAST-X	Precomputational Stack-X
FUNCTION	

The HOURS MINUTES SECONDS function converts a decimal equivalent of degrees or hours to a display using hh.mmss where:

hh	indicates hours or degrees
mm	indicates minutes
ss	indicates seconds

If the seconds are not an even two digit number, the seconds portion display indicates fractional seconds.

RELATED FUNCTIONS HMS+, HMS-, HR

KEY	Use CATALOG 3 or **XEQ HMS+ nn**
NAME	HOURS MINUTES SECONDS + nn
STACK	Enables Stack-lift, Stack popped
LAST-X	Precomputational Stack-X
FUNCTION	

This function adds the contents of the Stack-X register to the contents of the register indicated by the nn. The addition assumes the contents of both registers are in the format: hours . minutes seconds.

RELATED FUNCTIONS HMS, HMS-, HR

KEY — Use CATALOG 3 or **XEQ HMS-**
NAME — HOURS MINUTES SECONDS - nn
STACK — Enables Stack-lift, Stack popped
LAST-X — Precomputational Stack-X
FUNCTION

This function subtracts the contents of the Stack-X register from the contents of the register indicated by the nn. The subtraction assumes the contents of both registers are in the format: hours . minutes seconds.

RELATED FUNCTIONS HMS, HMS+, HR

KEY — Use CATALOG 3 or **XEQ HR**
NAME — HOURS TO DECIMAL
STACK — Enables Stack-lift
LAST-X — Precomputational Stack-X
FUNCTION

This function converts the contents of the Stack-X register from the format: hours . minutes seconds to the decimal equivalent.

RELATED FUNCTIONS HMS, HMS-, HMS+

KEY — Use CATALOG 3 or **XEQ INT**
NAME — INTEGER
STACK — Enables Stack-lift
LAST-X — Precomputational Stack-X
FUNCTION

The INTEGER function returns in the Stack-X register the integer portion of the Stack-X register.

RELATED FUNCTIONS FRC

KEY — Use CATALOG 3 or **XEQ LLIST**
NAME — LIST PROGRAM TO PRINTER
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The LLIST function will list to the printer a

program that is resident within the program file directory. Executing the function will display **LLIST __** and the program waits for a maximum seven alphanumeric character program name to be entered. The printer flag (21) must have previously been set on.

RELATED FUNCTIONS None

KEY Use CATALOG 3 or **XEQ LN1+X**
NAME LN1+X
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

This function returns in the Stack-X register the natural logarithm of numbers close to one.

RELATED FUNCTIONS E↑X-1

KEY Use CATALOG 3 or **XEQ MEAN**
NAME MEAN
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The MEAN returns in the Stack-X and Stack-Y registers the mean values of the contents of the stat block. The function assumes values have been entered using the + function.

RELATED FUNCTIONS SDEV, Σ+, Σ-, ΣREG

KEY Use CATALOG 3 or **XEQ MOD**
NAME MODULO
STACK Enables Stack-lift, Stack popped
LAST-X Precomputational Stack-X
FUNCTION

The MODULO function divides the Stack-Y register by the Stack-X register and returns in the Stack-X register the remainder of that division.

RELATED FUNCTIONS None

KEY — Use CATALOG 3 or **XEQ NOT**
NAME — NOT
STACK — Enables Stack-lift
LAST-X — Precomputational Stack-X
FUNCTION

The NOT function returns in the Stack-X register the boolean negation of the previous contents of the Stack-X register.

RELATED FUNCTIONS — XOR, AND, OR

KEY — Use CATALOG 3 or **XEQ OCT** or **F3**
NAME — OCTAL
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The OCTAL function switches the data entry and display mode of values to an octal format. The acceptable digits are 0 through 8.

RELATED FUNCTIONS — DEC, BIN, HEX

KEY — Use CATALOG 3 or **XEQ OFF**
NAME — OFF
STACK — Not applicable
LAST-X — Not applicable
FUNCTION

This function will terminate the calculator. If used with the pop-up version, it will terminate the action of the calculator but leave it resident.

RELATED FUNCTIONS — None

KEY — Use CATALOG 3 or **XEQ OR**
NAME — OR
STACK — Enables Stack-lift
LAST-X — Precomputational Stack-X
FUNCTION

The OR function ORs the contents of the Stack-Y register against the contents of the Stack-X register. The result is placed in the Stack-X register. This is a boolean operation.

RELATED FUNCTIONS — XOR, AND, NOT

KEY — Use CATALOG 3 or **XEQ COPY**
NAME — COPY FROM DISK
STACK — Not applicable
LAST-X — Not applicable
FUNCTION

This function loads a program into the program buffer but does not execute it. This is useful with the SINGLE STEP function for debugging purposes.

RELATED FUNCTIONS — SST, XEQ, GTO

KEY — Use CATALOG 3 or **XEQ PROMPT**
NAME — PROMPT
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The PROMPT function is used in a program to display the Alpha register, usually containing a prompt for the user to make an entry of some kind. A message will display in the instruction area to press the RUN/STOP key to continue execution of the program.

RELATED FUNCTIONS — PSE

KEY — Use CATALOG 3 or **XEQ PSE**
NAME — PAUSE
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The PAUSE function within a program halts execution for approximately 1 second and displays the contents of the Stack-X register.

RELATED FUNCTIONS — PROMPT

KEY — Use CATALOG 3 or **XEQ R-D**
NAME — RADIANS TO DEGREES
STACK — Enables Stack-lift
LAST-X — Precomputational Stack-X
FUNCTION

The RADIANS TO DEGREES function is a conversion process which replaces the contents of the Stack-X register with the degree equivalent.

RELATED FUNCTIONS D-R

KEY Use CATALOG 3 or **XEQ R↑**
NAME ROLL UP
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The ROLL UP function shifts the contents of the Stack registers up one register. The previous contents of the Stack-X register is rotated down into the Stack-T register.

RELATED FUNCTIONS RDN, ENTER↑

KEY Use CATALOG 3 or **XEQ RAD**
NAME RADIANS
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The RADIANS function changes the angular measurement mode to radians.

RELATED FUNCTIONS DEG, GRAD

KEY Use CATALOG 3 or **XEQ RAN#**
NAME **RAN#**
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The RAN# function generates a random number between 0 and 32768.

RELATED FUNCTIONS None

KEY Use CATALOG 3 or **XEQ SDEV**
NAME STANDARD DEVIATION
STACK Enables Stack-lift
LAST-X Precomputational Stack-X

FUNCTION

The STANDARD DEVIATION function computes the standard deviation sample of the stat block. It places the sample of the X values in the Stack-X and the y values in the Stack-Y.

RELATED FUNCTIONS MEAN, Σ+, Σ-, REG

KEY Use CATALOG 3 or **XEQ SIGN**
NAME SIGN
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The SIGN function replaces the Stack-X register with a one (1) using the sign of the previous Stack-X register.

RELATED FUNCTIONS None

KEY Use CATALOG 3 or **XEQ ST+ nn**
NAME STORE PLUS nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The STORE PLUS function adds the contents of the Stack-X register to the register number supplied. This function can use indirection.

RELATED FUNCTIONS ST-, ST*, ST/

KEY Use CATALOG 3 or **XEQ ST- nn**
NAME STORE MINUS nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The STORE MINUS function subtracts the contents of the Stack-X register from the register number supplied. This function can use indirection.

RELATED FUNCTIONS ST+, ST*, ST/

KEY — Use CATALOG 3 or **XEQ ST* nn**
NAME — STORE TIMES nn
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The STORE TIMES function multiplies the contents of the Stack-X register with the register number supplied. This function can use indirection.

RELATED FUNCTIONS — ST+, ST-, ST/

KEY — Use CATALOG 3 or **XEQ ST/ nn**
NAME — STORE DIVIDE nn
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The STORE DIVIDE function divides the contents of the register number supplied by the Stack-X register. This function can use indirection.

RELATED FUNCTIONS — ST+, ST-, ST*

KEY — Is used within a program
NAME — STOP
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The STOP function halts the execution of a program.

RELATED FUNCTIONS — XEQ, R/S, END

KEY — Use CATALOG 3 or **XEQ TONE n**
NAME — TONE n
STACK — Enables Stack-lift
LAST-X — Not applicable
FUNCTION

The TONE function sounds a tone corresponding to a number supplied. This supplied number must be within the range 0 through 9, the lower the number, the lower the tone generated.

NOTE: In program usage two (2) digits must be supplied within the program text (ie. STONE 02).

RELATED FUNCTIONS BEEP

KEY Use CATALOG 3 or **XEQ XOR**
NAME EXCLUSIVE OR
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The EXCLUSIVE OR function XORs the Stack-X register contents against the Stack-Y register contents and places the result in the Stack-X register. This is a boolean operation.

RELATED FUNCTIONS AND, OR, NOT, <<, >>

KEY Use CATALOG 3 or **XEQ X#0?**
NAME Stack-X NOT EQUAL 0?
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The Stack-X NOT EQUAL 0 function is used within programs to skip the next sequential instruction unless the contents of the Stack-X register does NOT equal zero.

RELATED FUNCTIONS X<0?, X>0?, X#Y?, etc.

KEY Use CATALOG 3 or **XEQ X<0?**
NAME Stack-X LESS THAN 0?
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The Stack-X LESS THAN 0 function is used in programs to skip the next sequential instruction unless the Stack-X register is less than zero.

RELATED FUNCTIONS X#0?, X>0?, X#Y?, etc.

KEY Use CATALOG 3 or **XEQ X>0?**
NAME Stack-X GREATER THAN 0?
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The Stack-X GREATER THAN 0 function is used in programs to skip the next sequential instruction unless the contents of the Stack-X register is greater than zero.

RELATED FUNCTIONS X#0?, X<0?, X#Y?, etc.

KEY Use CATALOG 3 or **XEQ X#Y?**
NAME Stack-X NOT EQUAL TO Stack-Y?
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The Stack-X NOT EQUAL TO Stack-Y function is used within programs to skip the next sequential instruction if the contents of the Stack-X register is equal to the contents of the Stack-Y register.

RELATED FUNCTIONS X#0?, X>0?, X<0?, etc.

KEY Use CATALOG 3 or **XEQ X<> nn**
NAME Stack-X EXCHANGE WITH nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

The Stack-X EXCHANGE WITH function exchanges the contents of the Stack-X register with register number supplied.

RELATED FUNCTIONS X<>Y

KEY Use CATALOG 3 or **XEQ %CH**
NAME PERCENT CHANGE
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The PERCENT CHANGE function calculates the percent of change the Stack-X register has to the Stack-Y register. This is computed by the formula:

((Stack-X - Stack-Y) * 100) / Stack-Y.

The the resultant percent is stored in the Stack-X register.

RELATED FUNCTIONS %

KEY Use CATALOG 3 or **XEQ ΣREG nn**
NAME REG nn
STACK Enables Stack-lift
LAST-X Not applicable
FUNCTION

This function sets the current six (6) register block to be used as the stat block beginning with the register supplied. Any statistical functions used from that point on will use the new stat block rather than the default stat block beginning at register 11.

RELATED FUNCTIONS Σ+, Σ-, MEAN, SDEV

KEY Use CATALOG 3 or **XEQ << n**
NAME SHIFT LEFT n
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The SHIFT LEFT function shifts the contents of the Stack-X register left the number of bits supplied. This is a boolean function useful to programmers.

RELATED FUNCTIONS >> n

KEY Use CATALOG 3 or **XEQ >> n**
NAME SHIFT RIGHT n
STACK Enables Stack-lift
LAST-X Precomputational Stack-X
FUNCTION

The SHIFT RIGHT function shifts the contents of the Stack-X register right the number of bits supplied. This is a boolean function useful to programmers.

RELATED FUNCTIONS << n

Function Tables

The following abbreviations are used in the tables:

D = Stack-lift is disabled.
E = Stack-lift is enabled.
I = The function can use indirection.
L = The precomputational Stack-X is store in the Last-X register.
P = The Stack registers are popped.
U = The Stack registers are pushed.
N/A= Not applicable.

CALC MODE

Key	Name	Execute Name	Prg Ctl	Description	Stack
A	Σ+	Σ+	S	Accumulates X & Y regs into stat block	L,D
B	1/X	1/X	S	Reciprocal of Stack-X	L,E
C	√x	SQRT	S	Square root of Stack-X	L,E
D	LOG	LOG	S	Natural Logarithm of the Stack-X	L,E
E	LN	LN	S	Common Logarithm of the Stack-X	L,E
F	X<>Y	X<>Y	S	Exchange contents of Stack-X w/ Stack-Y	E
G	R	RDN	S	Rolls the contents of Stack regs down	E
H	SIN	SIN	S	Sine of Stack-X	L,E
I	COS	COS	S	Cosine of Stack-X	L,E
K	XEQ	XEQ	X	Execute a function or program	E,I
L	STO	STO nn	T	Stores content Stack-X in register nn	E,I
M	RCL	RCL nn	T	Copy content of register nn into Stack-X	U,E,I
;	SST	SST	N/A	Single step a program	N/A
N	ENTER	ENTER©	S	Copy content of Stack-X into Stack-Y	D,U
O	CHS	CHS	S	Change the sign of Stack-X	E
P	EEX	EEX	N/A	Add the exponent to number in data entry	N/A
<	N/A	N/A	N/A	Edit the number input, clear Stack-X	N/A
↖	R/S	STOP	S	Halt or continue program executing	E
-	-	-	S	Stack-Y minus Stack-X	L,E,P
+	+	+	S	Stack-Y plus Stack-X	L,E,P
*	*	*	S	Stack-Y times Stack-X	L,E,P
/	/	/	S	Stack-Y divided by Stack-X	L,E,P

FUNCTION TABLES
SHIFT CALC FUNCTIONS

Key	Name	Execute Name	Prg Ctl	Description	Stack
A	Σ-	Σ-	S	Removes last entries from stat block	L,D
B	Y↑X	Y↑X	S	Raises the Stack-Y to power of Stack-X	L,E,P
C	X↑2	X^2	S	Squares the Stack-X register	L,E
D	10x	10^X	S	Common Exponential	L,E
E	E^x	E^X	S	Natural Exponential	L,E
F	CLd	CLd	S	Clears the stat block	E
G	%	%	S	Takes percent (Stack-X) of Stack-Y	L,E
H	ASIN	ASIN	S	Arc sine function (Stack-X = sine)	L,E
I	ACOS	ACOS	S	Arc cosine function (Stack-X = cosine)	L,E
J	ATAN	ATAN	S	Arc tangent function (Stack-X = tangent)	L,E
K	ASN	ASN	N/A	Assign function to a key	N/A
L	LBL	LBL	L	Not implemented in command mode	E
M	GTO	GTO	G	Set internal pointer to line or label	E,I
;	BST	BST	N/A	Not implemented	N/A
P	RTN	RTN	S	End subroutine or program	E
BS	CLX	CLX	S	Clear Stack-X register	D
Q	x=y?	X=Y?	S	Skip unless Stack-X equal to Stack-Y	E
R	SF	SF nn	T	Set flag nn ON 00 <= nn <= 29	E,I
S	CF	CF nn	T	Clear flag nn (OFF) 00 <= nn <= 29	E,I
T	FS?	FS? nn	T	Skip unless flag nn is ON 00 <= nn <= 55	E,I
U	X<=Y	X<=Y?	S	Skip unless Stack-X equal to Stack-Y	E
V	BEEP	BEEP	S	Sound four tones	E
W	P-R	P-R	S	Polar to Rectangular conversion	L,E
X	R-P	R-P	S	Rectangular to Polar conversion	L,E
Y	X<Y?	X<Y?	S	Skip unless Stack-X less than Stack-Y	E
Z	FIX	FIX nn	T	Display nn significant digits	E,I
=	SCI	SCI nn	T	Display nn significant digits Scientific	E,I
?	ENG	ENG nn	T	Display nn significant digits Engineerin	E,I
:	X=0?	X=0?	S	Skip unless Stack-X equal to zero	E
Space	PI	PI	S	PI to Stack-X	E,U
,	LASTx	LASTX	S	Last-X register to Stack-X	E,U
✎	VIEW	VIEW nn	T	View the nn register	E,I

SHIFT ALPHA FUNCTIONS

Key	Name	Execute Name	Prg Ctl	Description	Stack
K	APP	N/A	@	Set status to append to current ALPHA	N/A
L	ASTO	ASTO nn	T	Stores first 6 char of ALPHA in reg nn	E,I
M	ARCL	ARCL nn	T	Copies contents of reg nn into ALPHA	E,I
<	CLA	CLA	S	Clears contents of ALPHA register	E
✎	AVIEW	AVIEW	S	Views contents of ALPHA reg	E

Name	Execute Name	Prg Ctl	Description	Stack
ABS	ABS	S	Absolute value Stack-X	L,E
ADV	ADV	S	Advances paper if print ON	E
AND	AND	S	Stack-X = Stack-X anded with Stack-Y	L,E
AOFF	AOFF	S	Alpha mode OFF	E
AON	AON	S	Alpha mode ON	E
ASHF	ASHF	S	Shift the ALPHA reg right 6 characters	E
BIN	BIN	S	Switches data entry display to binary	E
CLD	CLD	S	Clears the display (use BACK SPACE)	E
CLP	CLP	S	Clears program from memory	E
CLRG	CLRG	S	Clears the memory registers	E
CLST	CLST	S	Clears the Stack registers	E
DATE	DATE	S	Display present date and time	L,E
DEC	DEC	S	Switches data entry display to decimal	E
DEG	DEG	S	Switches angle measurement to degrees	E
DEL	DEL nnn	N/A	Has no function	
DSE	DSE nn	T	Program reverse counter	E,I
D-R	D-R	S	Degrees to radians conversion	L,E
END	END	S	End of program	E
E^X-1	E^X-1	S	Natural logarithm of numbers close to 1	L,E
FACT	FACT	S	Stack-X! factorial	L,E
FC?	FC? nn	T	Skips prog line if flag nn is clear	E,I
FC?C	FC?C nn	T	Skips line if flag nn is set, & clears	E,I
FRC	FRC	S	Stack-X equals decimal part Stack-X	L,E
FS?C	FS?C nn	T	Skips line if flag nn is clear, & clears	E,I
GRAD	GRAD	S	Switches angle measurement to gradians	E
HEX	HEX	S	Switches data entry to hexadecimal	E
HMS	HMS	S	Decimal hours to hours.minutes seconds	L,E
HMS+	HMS+ nn	T	Hours min sec add to register nn	L,E,P
HMS-	HMS- nn	T	Hours min sec sub from register nn	L,E,P
HR	HR	S	Hours min sec to decimal hours	L,E
INT	INT	S	Integer portion of Stack-X	L,E
LLIST	LLIST	N/A	Lists a program to printer w/line #s	E
LN1+X	LN1+X	S	Natural logarithm of numbers close to 1	L,E
MEAN	MEAN	S	Means of stat block Stack-X and Stack-Y	L,E
MOD	MOD	S	Stack-Y mod Stack-X (Remainder)	L,E,P
NOT	NOT	S	Negate Stack-X register	L,E
OCT	OCT	S	Switches data entry to octal	E
OFF	OFF	S	Exits calculator (pop-up remains res)	
ON	ON	N/A	Has no function	
OR	OR	S	Stack-X = Stack-X ORed with Stack-Y	L,E
COPY	COPY	N/A	Loads program (without execution)	
PROMPT	PROMPT	S	Displays ALPHA reg with hold (R/S mesg)	E
PSE	PSE	S	Holds display (1 sec) during prog exec	E
R-D	R-D	S	Convert Stack-X radians to degrees	L,E
R↓	R↓	S	Rolls stack up 1 register	E
RAD	RAD	S	Switches angle measurement to radians	E
RAN#	RAN#	S	Generate random number (0 to (2©15 - 1))	L,E
SDEV	SDEV	S	Standard deviation of stat block	L,E
SIGN	SIGN	S	Stack-X register's sign to 1 (1 or -1)	L,E
ST+	ST+ nn	T	Add Stack-X to register nn	E,I
ST-	ST- nn	T	Subtract Stack-X from register nn	E,I
ST*	ST* nn	T	Multiply register nn with Stack-X	E,I

Name	Execute Name	Prg Ctl	Description	Stack
ST/	ST/ nn	T	Divide register nn by Stack-X	E,I
STOP	STOP	S	Stop execution of program	E
TONE	TONE n	T	Sound tone (0 to 9) /prgm needs (nn)	E,I
XOR	XOR	S	Stack-x = Stack-X XORed with Stack-Y	L,E
X#0?	X#0?	S	Skip program line if Stack-X not equal 0	E
X<0?	X<0?	S	Skip line unless Stack-X less than 0	E
X>0?	X>0?	S	Skip line unless Stack-X greater than 0	E
X#Y?	X#Y?	S	Skip line unless Stack-X not equal 0	E
X<>	X<> nn	T	Exchange Stack-X reg with Stack-Y reg	E
%CH	%CH	S	Percent change from Stack-Y to Stack-X	L,E
ΣREG	ΣREG nn	T	Set stat block beginning register	E
<<	<< n	T	Shift Stack-X left n bits (boolean)	L,E
>>	>> n	T	Shift Stack-X right n bits (boolean)	L,E

Example 4-1:

A series of measurements of the length of a concrete runway is made using a steel tape. The results (in meters) are tabulated below:

1363.7 1364.5 1364.0 1363.8 1364.0 1364.1

Compute the mean and the standard deviation for the measurements.

KEY PRESS	DISPLAY	DESCRIPTION
F10	0.0	SHIFT CALC
F	0.0	CLd
1363.7	1363.7_	
A	1.0	Σ+
1364.5	1364.5_	
A	2.0	Σ+
1364.0	1364.0_	
A	3.0	Σ+
1363.8	1363.8_	
A	4.0	Σ+
1364.0	1364.0_	
A	5.0	Σ+
1464.1	1464.1_	
A	6.0	OH! TYPING ERROR
1464.1	1464.1_	
F10	1464.1	SHIFT CALC
A	5.0	Σ-
1364.1	1364.1_	
A	6.0	Σ+
K	XEQ __	
MEAN	XEQ MEAN_	
ENTER KEY	1,364.01666667	MEAN OF THE DATA
K	XEQ __	
SDEV	XEQ SDEV_	
ENTER KEY	0.278687398	STANDARD DEVIATION OF DATA

PROGRAMMING EXAMPLE:

```
LSTATTST
TΣREG 20
SCLΣ
N1363.7
SΣ+
N1364.5
SΣ+
N1364.0
SΣ+
N1363.8
SΣ+
N1364.0
SΣ+
N1364.1
SΣ+
SMEAN
AMEAN=
```

```
SARCL STK X
SPROMPT
SSDEV
ASDEV=
SARCL STK X
SPROMPT
SEND
```

EXAMPLE 4-2:
Find the square root of 16 and the square of the result. NOTE: Square root only accepts positive numbers.

KEY PRESS	DISPLAY	DESCRIPTION
16	16_	
C	4.0	SQUAREROOT
F10	4.0	SHIFT CALC
C	16.0	SQUARE
O	-16.0_	CHANGE SIGN
C	DATA ERROR!	NUMBER MUST BE >= 0.0

EXAMPLE 4-3:
Find the logarithm and the natural logarithm of 13 also 10 and (e) to the power of Stack-X.

KEY PRESS	DISPLAY	DESCRIPTION
13	13_	NON NEGATIVE NUMBER
D	1.113943352	LOG BASE 10 OF 13
F10	1.113943352	SHIFT CALC
D	13.0	10 TO THE POWER OF X
E	2.564949357	LOG BASE e(2.718...) OF 13
F10	2.564949357	SHIFT CALC
E	13.0	e TO THE POWER OF X

EXAMPLE 4-4:
Find the sine of c/4 in radians. What is the angle whose sine is 0.7854.

KEY PRESS	DISPLAY	DESCRIPTION
K	XEQ __	
RAD	0.0	CHANGE MODE TO RADIANS
F10	0.0	SHIFT CALC
Space bar	3.141592654	PI
4	4_	
/	0.785398163	
H	0.707106781	SINE OF c/4
F10	0.707106781	SHIFT CALC
H	0.785398163	ARC SINE OF 0.707106781

EXAMPLE 4-5:
Following is an example of different forms of executing store and recall.

```
PROGRAMMING EXAMPLE
LSTOTEST
N10
TSTO 00                 ; STORE 10 IN REG 00
TSTO IND 00             ; STORE 10 IN REG 10
TSTO STK T              ; STORE 10 IN STACK-T
TSTO IND STK T          ; STORE 10 IN REG 10
TST+ 00                 ; ADD STACK-X TO REG 00
TST+ IND 00             ; ADD STACK-X TO REG POINTED TO BY REG 00
TST+ STK T              ; ADD STACK-X TO STACK-T
TST+ IND STK T          ; ADD STACK-X TO REG POINTED TO BY STACK-T
TRCL 00
TRCL IND 00
TRCL STK T
TRCL IND STK T
SEND
```

EXAMPLE 4-6:
The following example is supplied to show the use of EEX and CHS in data entry.

KEY PRESS	DISPLAY	DESCRIPTION
234.567	234.567_	
O	-234.567_	CHANGE SIGN
P	-234.567E_	EEX
32	-234.567E32_	EXPONENT ENTRY
O	-234.567E-32_	NEGATIVE EXPONENT
ENTER KEY	-2.345670000E-30_	

PROGRAMMING THE ELI-41

ELI-41 programs are structured the same as Hewlett-Packard calculator programs. The program instructions follow the same sequence as if the key strokes were being directly entered at the time of the calculation. It is beyond the scope of this manual to give detailed instruction on programming technique or structure; however, some simple examples will be given to illustrate the use of ELI-41's special functions. Users that are not familiar with programming in Hewlett-Packard's RPN format are advised to refer to the READ.ME file on the distribution disk for sources of detailed programming instruction.

CREATING AN ELI-41 PROGRAM WITH AN EDITOR

ELI-41 programs can be created with any text editor that outputs an ASCII file to disk. Program structure is identical to the HP-41 calculator with the addition of a program control character at the beginning of each line. When writing a program, the following syntax must be adhered to:

1. The label statement.
   A program must start with a label statement that contains the program file name, less the file extension. Comment lines may come before the label statement.

2. The first character.
   The first character in a statement must be either a semicolon (;) or a program control character. The program control characters and the ELI-41 command types that they are to be used with are listed in Table 5.1. Chapter4 lists all of the ELI-41 commands and the assigned program control characters.

3. Maximum characters.
   The maximum number of characters per line is 79.

4. Comments.
   A comment must begin with a semicolon (;) in/or before the 26th column. A space must separate any executable statement or command and the semicolon.

5. Maximum program length.
   The maximum program length is 300 executable statements. Programs longer than this may be broken into segments or subroutines and linked together with the XEQ or GTO command. This feature is illustrated in Example 5.3. (Program lines that do not contain an executable statement or command are not counted when determining the program length.)

6. The END statement.
   A program must end with an END statement.

Cntr Char	Command Format	Description
L	Label	Marks the beginning of a program or subroutine. The variable 'label' may be up to 7 alpha or numeric characters.
N	Nnn	Places the numeric value 'nnn' in the Stack-X register. The variable 'nnn' may be up to 16 digits.
T	Tfunc	Designates a function, 'func' that requires additional parameters (ie. STO).
A	Aaaa	Places the string 'aaa'in the ALPHA register. The variable 'aaa' may be up to 24 alpha characters.
S	Sfunc	Designates a single key function that operates directly with Stack-X register with no additional parameters (ie. COS).
X	Xlabel	Designates the XEQ command. Branches program execution to the disk resident program or subroutine specified by 'label'. Program executing returns to the calling program on the line following the XEQ command when a RTN command is encountered. XEQ subroutines may be nested to 30 levels deep.
G	Glabel	Designates the GTO command. Transfers program execution to the specified disk resident program, local alpha or numeric label.
C	Clabel	Loads the conversion file specified by 'label'. The file may be loaded at any time during program execution prior to a Fnn command.
F	Fnn	Specifies the Convert From matrix element, 'nn', of the conversion file previously loaded with the Clabel command. Must be immediately followed by a Dnn command.
D	Dnn	Specifies the Convert To matrix element, 'nn', of the previously loaded conversion file and performs the conversion. Must immediately follow a Fnn command.
@	@aaa	Appends the string 'aaa' to the Alpha register. The @ symbol is a substitute for the normal HP-41 symbol (├).

Table:5-1 PROGRAM PREFIXES

The following examples will illustrate some programming basics and the proper program format.

A common problem encountered in many disciplines is the solution to quadratic equations having the form:

$$x_1 = \frac{-v + \sqrt{(b2 - 4ac)}}{2a}$$

and:

$$x_2 = \frac{-b - \sqrt{(b2 - 4ac)}}{2a}$$

A possible ELI-41 keystroke sequence to calculate the solution is:

KEYSTROKE	DESCRIPTION
nn	Enter coefficient 'a'
STO 01	Store in register 01
nn	Enter coefficient 'b'
STO 02	Store in register 02
nn	Enter coefficient 'c'
STO 03	Store in register 03
RCL 02	Recall 'b' coefficient
CHS	Loads -'b' on the stack
RCL 02	Recalls 'b' coefficient
X^2	Calculates & loads 'b' squared into stack
RCL 01	Recall 'a' coefficient
RCL 03	Recall 'c' coefficient
*	Calculates a * c
4	
*	
-	
SQRT	Calculates (b 2 - 4ac)
STO 04	Stores intermediate result in 04
+	
2	
/	
RCL 01	
/	
X<>Y	
RCL 04	
-	
2	
/	
RCL 01	
/	Calculates X_2 root

Example:5-1 SOLUTION TO QUADRATIC EQUATIONS

An ELI-41 program can be written directly from this keystroke sequence. By adding a Program Control Character

to each function and saving the file, this program can be recalled and executed whenever desired. User prompts have been added to enhance the program operation.

```
;This program solves for the real roots of a quadratic equation.
;                                        ** CAUTION **
;             This program will work only if both roots are real.
;
LROOTS          ;Program file must be ROOTS.PGM
AINPUT a
SPROMPT
TSTO 01         ; Inputs 'a' to register 01
AINPUT b
SPROMPT
TSTO 02         ; Inputs 'b' to register 02
AINPUT c
SPROMPT
TSTO 03         ; Inputs 'c' to register 03
TRCL 02
SCHS            ; Loads Stack-X with -b
TRCL 02         ; Pushes Stack-X register and stores register 02 in X
SX©2            ; Loads stack with b2
TRCL 01
TRCL 03
S*
N4
S*
S-
SSQRT           ; Calculates (b - 4ac) (1/2)
TSTO 04         ; Stores the above intermediate result in register 04
S+              ; Calculates -b + √(b - 4ac)
N2              ; (Recall that -b was loaded in the stack
S/
TRCL 01
S/              ; Calculates X1
TSTO 05         ; Stores X1
SPSE            ; Pauses to display X1
SX<>Y           ; Moves -b from Stack-Y to Stack-X register
TRCL 04
S-              ; Calculates -b - √(b - 4ac)
N2
S/
TRCL 01
S/              ; Calculates X2
TSTO 06         ; Stores X2
; SRTN          ; Could be included to allow use as a subroutine
SEND            ; All programs must conclude with an END statement
```

Example:5-2 ASCII PROGRAM FILE

SUBROUTINES

When a program encounters an 'Xlabel' statement, the subroutine 'label' will be loaded from disk and executed. The stack and registers will not be effected during the load and may be used to pass values to and from the subroutine. When a RTN statement is encountered in the subroutine, execution

is returned to the calling program on the line immediately following the Xlabel statement.

To understand the way subroutines work, let's look at the operation of the programs that will be used in this example. This one is included on your ELI-41 disk.

The program XEQTEST reads:

```
LXEQTEST        ; Program name label
XTONE1          ; Calls the subroutine 'TONE1'
AMARY           ; Places the string 'MARY' in the
TASTO 00        ;   alpha register 00
XIND 00         ; This is an indirect subroutine call.
; When this instruction is executed the
; program will branch to the label contained
; in the referenced register (00).
; In this case register 00 contains 'MARY'
; and the program loads and executes the
; MARY program.
SEND            ; Program ends
```

Example:5-3 SUBROUTINE CALLS

CONVERSION TABLES

A conversion table is a standard ascii text file which contains information to change a value from one unit of measure to another. ELI supplies several standard tables with the ELI-41 System diskette.

USING ELI-41'S CONVERSION TABLES

The value to be converted is assumed to reside in the Stack-X register. That being true, the conversion table is loaded into memory by one of two methods.

1. The conversion key is pressed (F6).

2. The catalog function is called, (the 'N' key, located in the SHIFT CALC mode) and the CATALOG 5 function is chosen.

The calculator face will reveal the .CNV files resident in the CONVERSION directory. Using the arrow keys to place the inverse cursor over the desired conversion file, the enter key is pressed to signify the choice.

In the following example **Boldface** signifies user input. Assume the Stack-X register contains a value of 3.345 cm. And the result looked for being in feet, the calculator face would contain from the LENGTH.CNV file:

1. cm
2. inches
3. feet
4. miles

Convert From **01** Convert To **03**

The result rendered as 0.109743875 feet would be in the Stack-X register.

The calculator displays **CONVERT FROM** __ on the command line, seeking a two digit number. The number entered represents the type value of the Stack-X register contents as represented by the types showing on the calculator face. Place holders are necessary, so type one (1) would be chosen as 01.

The character string **CONVERT TO** __ is then displayed in the calculator input area, requesting a two digit number corresponding to the type of the resulting conversion. That number will replace the contents of the Stack-X register.

NOTE: At any time before the final key press the conversion can be aborted or the conversion 'from' or 'to' can be edited by use of the gray BACKSPACE key. Repeated entry of this key backs out of this function.

MAKING YOUR OWN CONVERSION TABLES

In addition to the supplied tables, others may generated with a word processor or text editor by the user. Up to 90 different tables can be accessed (within a single directory path). Each table can contain up to a 10 by 10 matrix. An example of a linear measure conversion table follows.

CONVERT FROM	CONVERT TO			
	Inches	Feet	Yards	Miles
Inches	1.0	0.833333	0.277777778	0.0000157828
Feet	12.0	1.0	0.333333333	0.0001893939
Yards	36.0	3.0	1.0	0.0005681818
Miles	63360.0	5280.0	1760.0	1.0

CONVERSION WORK TABLE

The following represents a disk file derived from this conversion table. It would be created with the use of a text editor and saved to disk. Naming the file must be done according to standard DOS conventions. Consult your DOS manual for further information.

```
                    1    1    2    2
Line       1...5....0....5....0....5.....

  1        ;Linear Conversion Table
  2        ; Note that comments always start with a semicolon
  3        ;      and may be included on statement lines.
  4        4                    ;Matrix Size
  5        Inches               ;Description of Units
  6        Feet                 ;
  7        Yards                ;
  8        Miles                ;
  9        1.0                  ;Inches to Inches
 10        0.833333             ;Inches to Feet
 11        0.277777778          ;Inches to Yards
 12        0.0000157828         ;Inches to Miles
 13        12.0                 ;Feet to Inches
 14        1.0                  ;Feet to Feet
 15        0.3333333333         ;Feet to Yards
 16        0.0001893939         ;Feet to Miles
 17        36.0                 ;Yards to Inches
 18        3.0                  ;Yards to Feet
 19        1.0                  ;Yards to Yards
 20        0.0005681818         ;Yards to Miles
 21        63360.0              ;Miles to Inches
 22        5280.0               ;Miles to Feet
 23        1760.0               ;Miles to Yards
 24        1.0                  ;Miles to Miles
 ********** End Of File **********
```

Table:6-1 CONVERSION TABLE FILE LAYOUT

RULES FOR CODING CONVERSION TABLES

1. Conversion table file names.
   Conversion Table files must be named with a ".CNV" file extension.

2. Matrix sizes.
   Maximum of a 10 by 10 matrix is allowed.

3. Required Statements.

   Matrix Size This statement defines the size of the matrix.

   Descriptions There must be the exact number of description statements as the number specified in the Matrix Size statement. Description names greater than seven characters long will be truncated for display purposes.

   Conversion Factors The number of conversion factor statements must be the value of the Matrix Size statement squared. Values may be recorded in either decimal format or scientific notation.

4. Line lengths. The length of either a Comment Line or a Statement Line cannot exceed 79 characters.

5. Comment lines.
   Comments within the file always begin with a semicolon. There are two types of comments. Comment lines and comment statements. A Comment Line is an individual record within the file containing a semicolon in position 1 of the record followed by any text. A Comment Statement is any text preceded by a semicolon and a space on any of the required statement records. A space must always separate the required statement's value and the semicolon.

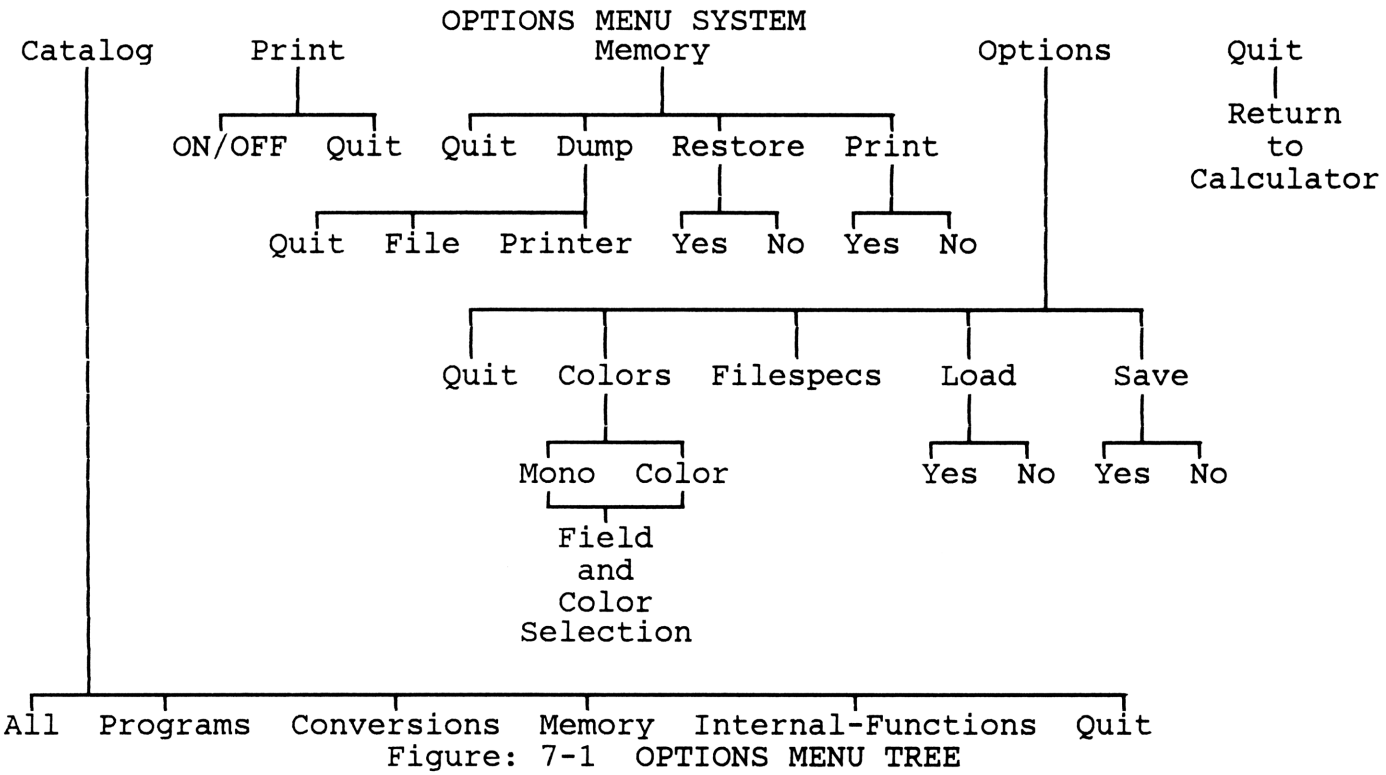


Figure: 7-1 OPTIONS MENU TREE

OPTIONS MENU

You invoke the Options Menu from the calculator display screen by pressing F5. This system allows you to make changes to ELI-41 to fit your machine environment and preferences. The highlighting, intensity and characteristics of various elements of the display can be changed from the defaults set at startup for the monocrome screen, as an instance. The colors, of course, can be changed for a color card and monitor.

The display is changed to a rectangular window containing a menu selection line and a description line which describes each function of the menu. This description is of the presently highlighted item.

You can make a selection in either of two ways. The first, overlay the menu item with the highlight bar using the cursor direction keys and press <ENTER>. And the second, press the first letter of any menu item. The program treats this just as if you had moved the highlight bar over the menu item and had pressed <ENTER>.

This menu system is multilayered, meaning, a selection may take you to a nested level of menus. At any time you may press the ESC key to return to the preceding menu. To return to the calculator display screen simply press the ESC key or select Quit from the top menu level.

Catalog

This main menu option obtains directory listings of files or internal functions according the following levels.

All

provides a catalog of all files from the "Utility Path" regardless of file extension. Sub directories will not be listed.

Programs

provides a catalog of **ELI-41** programs contained in the "Program Path" with file extension of ".PGM".

Conversion

provides a catalog of Conversion table files in the "Program Path" with file extension ".CNV".

Memory

provides a catalog of **ELI-41** "Dumped Memory" files in the "Memory Path" with file extension ".MEM".

Internal Functions

displays a list of all functions internal to ELI-41.

Quit

returns to the previous menu level.

Print

Activates and deactivates printer. This turns on or off the printer flag 21.

ON/OFF

Toggles the printer on or off. On the lower edge of the calculator face, an annuciator will inform of the condition of printer

Quit

returns to the previous menu level.

Memory

The memory management features the ability to save the present state of all registers and flags for reinstatement at a later time.

Dump

saves or prints ELI-41 memory contents. The Stack registers and flags are saved as they are but only memory registers whose contents are not zero are saved. Any program resident in memory is not saved.

File

saves the contents of memory to a disk file whose name must be specified in "Options/Filespecs". If none has been supplied the default of ELI.MEM is used.

Printer

prints the memory contents, Stack, memory registers, and flags. Does not depend on the state of the print flag (21), but a printer must have been on-line when the calculator was loaded. A status check on the printer was done at that time.

Quit

returns to the previous menu level.

Restore

retrieves a previously dumped memory file ("?.MEM") from disk and places it in memory. If no other file name has been supplied on the "Options/Filespecs" level, the default file name of ELI.MEM is used, if existent.

No

states no, do not retrieve file, do not disturb the contents of memory.

Yes

confirms, continue with operation, replace memory with data contained in the file.

Initialize

No

states no, do not initialize memory registers to zero, flags to defaults or Stack registers to zero.

Yes

confirms, continue with operation, initialize memory registers to zero, flags to defaults and Stack registers to zero.

Quit

returns to the previous menu level.

Options

miscellaneous environment settings are supplied on this level. They take effect immediately.

Colors

An equipment check was made at startup. If it showed no color equipment, changing colors will have no effect. The monochrome attributes , of course, would.

Monochrome

changes attributes for use of a monochrome monitor.

Select the screen display area you want to change (selections "A" through "N" on the right side of the screen) and you will be prompted to select a color attribute. The monochrome color selections are:

A - Normal Intensity
B - Bright Intensity
C - Reverse Highlight
D - Black

Color

Change attributes for use with a color monitor.

Select the screen display area you want to change (selections "A" through "N") and you will be prompted to select a color attribute. The available colors are:

A - Black
B - Blue
C - Green
D - Cyan
E - Red
F - Magenta
G - Brown
H - Light Grey
I - Dark Grey
J - Light Blue
K - Light Green
L - Light Cyan
M - Light Red
N - Light Magenta
O - Yellow
P - White

Note: Background color cannot be changed from Black. Any color attribute identified by "???" is a special combination of a foreground and background color. A combination of foreground and background colors (ie: Blue Foreground on a Red Background) cannot be accomplished.

Filespecs

allows changing directory paths for program, utility, conversion and memory files. The file name for the memory file is specified here as well.

Utility Directory

A general purpose directory path can be supplied to display a catalog of any drive and path.

Program Directory

A directory path must be supplied for accessing ELI-41 Programs (".PGM" files) and Conversion Tables (".CNV" files).

Memory Directory

The Directory path must be supplied to access saved memory files (".MEM"). This is used for saving or restoring ELI-41 memory.

Program File Name

The name of the most current program file executed is displayed.

Memory File Name

The name of the memory file to either save to or restore from must be supplied. A default of ELI.MEM is provided.

Load

Requests the loading of a disk option file ("ELI41.OPT") from here.

No

Do not load options file, use option settings currently in memory.

Yes

If ELI41.OPT is existent in the memory file directory, that file will be loaded into memory. The replacement takes the place of the options that are currently in memory.

Save

requests the saving of the current options to an options file ("ELI41.OPT") on disk. The current directory for memory files is used.

No

Do not save options to disk.

Yes

saves the current options to disk, overwriting any existing "ELI41.OPT"

file. The next time ELI-41 is started these newly saved options will become the defaults.

Quit

returns to the calculator display screen.

Flags

Sixty flags are continuously displayed on the calculator screen. An "ON" status is indicated by a small box while an "OFF" status is displayed as a centered dot.

Flags are set on by the "SF" command, off by the "CF" command. Both of these commands are in **Shift-Calc** mode. If you attempt to manipulate a flag that is either reserved for internal use, or is not available for use by the programmer (such as Flag 27 - User mode) a "NONEXISTANT" error message is given.

Flags 0 through 20 and flag 28 are not predefined and available for the programmer to define and set from within a program.

Other flags you can set that are predefined are:

Flag	Usage
21	Printer Activation
24	Range Error Ignore
25	Error Ignore
26	Audible Tone Enable/Disable
28	Separator (Used for Comma Editing)

The remaining flags are used by **ELI-41** for a number of purposes they can be tested for "ON/OFF" status but cannot be set by any programmer commands.

The following is a complete list of the flags used by the calculator. Those that are user changeable are asterisked.

*00	User	31	..Internal Use..
*01	User	32	..Internal Use..
*02	User	33	..Internal Use..
*03	User	34	..Internal Use..
*04	User	35	..Internal Use..
*05	User	36	..Internal Use..
*06	User	37	..Internal Use..
*07	User	38	..Internal Use..
*08	User	39	..Internal Use..
*09	User	40	Fixed Display Format
*10	User	41	Engineering Display Format
*11	User	42	Gradians Mode
*12	User	43	Radians Mode
*13	User	44	..Internal Use..
*14	User	45	..Internal Use..
*15	User	46	..Internal Use..
*16	User	47	Shift Mode
*17	User	48	Alpha Mode
*18	User	49	..Internal Use..
*19	User	50	Message Display
*20	User	51	SST
*21	Printer Activation	52	Program Mode
22	Numeric Data Input	53	..Internal Use..
23	Alpha Data Input	54	Pause

*24	Range Error Ignore	55	..Internal Use..
*25	Error Ignore	56	Numeric Lock
*26	Audible Tone Enable	57	Caps Lock
27	User Mode	58	..Internal Use..
*28	Comma editing	59	Execute
*29	Decimal (period/comma)	60	Run Mode
30	Catalog		

FUNCTION KEYS

F1 - Display contents of Stack-X in BASE 10 (Decimal)
F2 - Display contents of Stack-X in BASE 16 (Hex)
F3 - Display contents of Stack-X in BASE 2 (Binary)
F4 - Display contents of Stack-X in BASE 8 (Octal)
F5 - Go to System Options Screen
F6 - Perform Data Value Conversions (using ".CNV" disk file)
F7 - Not Used
F8 - Get into User mode
F9 - Get into ALPHA mode
F10 - Mode Shift Key (By pressing the Mode Shift Key in any mode the new mode will shift-mode. A second press of the Mode Shift key will return to the original mode).
EG:

KEYPRESS	MODE
	CALC
F10	SHIFT-CALC
F10	CALC

ALT/F1 - Exit from ELI-41 System

ECLIPSE LOGIC INC.
P.O. BOX 2003
HUNTINGTON PARK, CA 90255