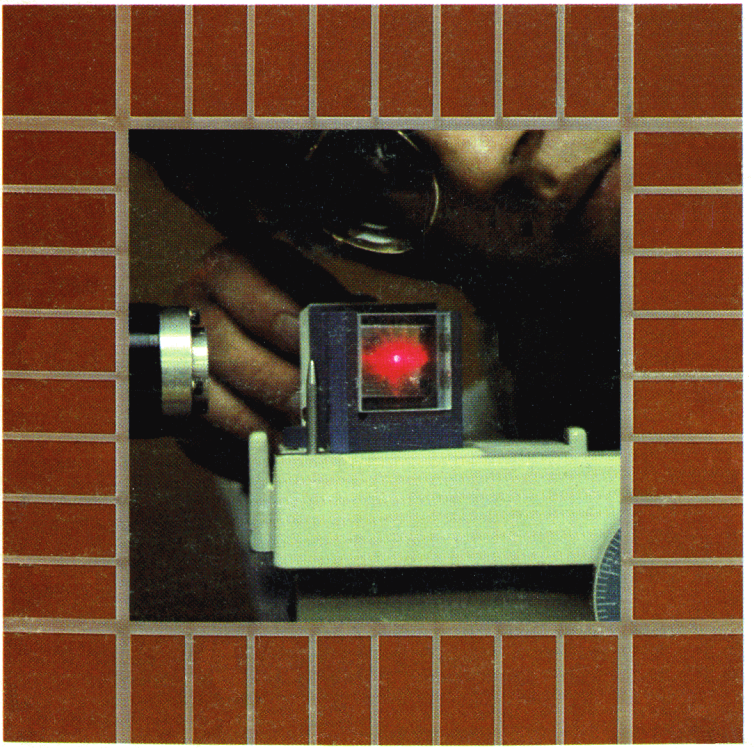
HEWLETT-PACKARD

HP-33E

OWNER'S HANDBOOK AND PROGRAMMING GUIDE



"The success and prosperity of our company will be assured only if we offer our customers superior products that fill real needs and provide lasting value, and that are supported by a wide variety of useful services, both before and after sale."

Statement of Corporate Objectives.
Hewlett-Packard

When Messrs. Hewlett and Packard founded our company in 1939, we offered one superior product, an audio oscillator. Today, we offer over 3500 quality products, designed and built for some of the world's most discerning customers.

Since we introduced our first scientific calculator in 1967, we've sold millions worldwide, both pocket and desktop models. Their owners include Nobel laureates, astronauts, mountain climbers, businesspersons, doctors, students, and homemakers.

Each of our calculators is precision crafted and designed to solve the problems its owner can expect to encounter throughout a working lifetime.

HP calculators fill real needs. And they provide lasting value.

The HP-33E Programmable Scientific Calculator Owner's Handbook and Programming Guide

February 1978

00033-90001

Printed in U.S.A.

Contents

The HP-33E Programmable Scientific Calculator

HP-33E
Programmable Scientific Calculator

Program Memory

00	
01–	13 00
02–	13 00
03–	13 00
04–	13 00
⋮	
48–	13 00
49–	13 00

Automatic Memory Stack

T 0.0000
Z 0.0000
Y 0.0000

Display

X 0.0000

LAST X

0.0000

Storage Registers

R_0	0.0000	
R_1	0.0000	
R_2	0.0000	n
R_3	0.0000	Σx
R_4	0.0000	Σx^2
R_5	0.0000	Σy
R_6	0.0000	Σy^2
R_7	0.0000	Σxy

Function Key Index

Function keys pressed from the keyboard execute individual functions as they are pressed. Input numbers and answers are displayed. All function keys listed below operate either from the keyboard or as recorded instructions in a program.

OFF  ON

[f] Pressed before function key, selects gold function printed above key.

[g] Pressed before function key, selects blue function printed on lower face of key.

Clear [PREFIX] after [f], [g], [STO], [RCL], [FIX], [SCI], [ENG], [GSB], or [GTO] cancels that key.

Digit Entry

[ENTER↑] Enters a copy of number displayed in X-register. Used to separate numbers.

[CHS] Changes sign of mantissa or exponent of 10 in displayed X-register.

[EEX] Enter exponent. After pressing, next numbers keyed in are exponents of 10.

0 through 9
Digit keys.

. Decimal point.

Number Manipulation

Clear [STK]
Clears contents of stack (X, Y, Z, T).

[R↓] Rolls down contents of stack for viewing in displayed X-register.

[X≷Y] Exchanges contents of X- and Y-registers of stack.

[CLx] Clears contents of displayed X-register to zero.

Display Control

[FIX] Fixed point display. Followed by a number key, selects fixed point notation display.

[SCI] Scientific display. Followed by a number key, selects scientific notation display.

[ENG] Engineering display. Followed by a number key, selects engineering notation display.

[MANT] Mantissa. Temporarily displays all 10 digits of the mantissa of the number in the X-register.

Number Alteration

[ABS] Gives absolute value of number in displayed X-register.

[INT] Leaves only integer portion of number in displayed X-register by truncating fractional portion.

[FRAC] Leaves only fractional portion of number in displayed X-register by truncating integer portion.

Manual Storage

[STO] Store. Followed by number key, stores displayed number in storage register specified. Also used to perform storage register arithmetic.

Function Key Index (Continued)

RCL Recall. Followed by number key, recalls value from storage register specified into the displayed X-register.

Clear REG Clears contents of all storage registers.

LST x Recalls number displayed before the previous operation back into the displayed X-register.

Mathematics

√x Computes square root of number in displayed X-register.

x² Computes square of number in displayed X-register.

1/x Computes reciprocal of number in displayed X-register.

π Places value of pi (3.141592654) into displayed X-register.

+ − × ÷ Arithmetic operators.

Trigonometry

DEG Sets decimal degrees mode for trigonometric functions.

RAD Sets radians mode for trigonometric functions.

GRD Sets grads mode for trigonometric functions.

SIN COS TAN Computes sine, cosine, or tangent of value in displayed X-register.

SIN⁻¹ COS⁻¹ TAN⁻¹ Computes arc sine, arc cosine, or arc tangent of number in displayed X-register.

→H.MS Converts decimal hours or degrees to hours, minutes, seconds or degrees, minutes, seconds.

→H Converts hours, minutes, seconds or degrees, minutes, seconds, to decimal hours or degrees.

→DEG Converts radians to degrees.

→RAD Converts degrees to radians.

Polar/Rectangular Conversion

→P Converts x and y rectangular coordinates placed in X- and Y-registers to polar coordinates with magnitude r and angle θ.

→R Converts polar coordinates with magnitude r and angle θ in X- and Y-registers to rectangular x and y coordinates.

Logarithmic and Exponential

y^x Raises number in Y-register to power of number in displayed X-register.

10^x Common antilogarithm. Raises 10 to power of number in displayed X-register.

e^x Natural antilogarithm. Raises e (2.718281828) to power of number in displayed X-register.

Function Key Index (Continued)

LOG Computes common logarithm (base 10) of number in displayed X-register.

LN Computes natural logarithm (base *e*, 2.718281828) of number in displayed X-register.

Statistics

Σ+ Accumulates number from X- and Y-registers into storage registers R_0 through R_7.

Σ− Subtracts x and y values from storage registers R_0 through R_7 for correcting Σ+ accumulations.

$\bar{x}$ Computes mean (average) of x and y values accumulated by Σ+.

s Computes sample standard deviations of x and y values accumulated by Σ+.

L.R. Linear regression. Computes y-intercept (A) and slope (B) for x and y data points accumulated using Σ+.

$\hat{x}$ Linear estimate. Computes a predicted x value for a given y.

$\hat{y}$ Linear estimate. Computes a predicted y value for a given x.

r Computes correlation coefficient of x and y values accumulated by Σ+.

Percentage

% Computes x% of y.

Programming Key Index

PROGRAM Mode	Automatic RUN Mode	
All function keys except the functions shown below are loaded into program memory when pressed.	PRGM-RUN switch PRGM  RUN to RUN. Function keys may be executed as part of a recorded program or individually by pressing from the keyboard. Input numbers and answers are displayed by the calculator, except where indicated.	
Active keys:	**Pressed from keyboard:**	**Executed as a recorded program instruction:**
In PRGM mode only the following operations are active. These operations are used to help record programs, and cannot themselves be recorded in program memory.		
[GTO] Go to. Followed by .nn positions calculator to line nn of program memory. No instructions are executed **(page 35).**	[GTO] Go to. Followed by .nn sets calculator to line nn of program memory. No instructions are executed **(page 35).**	[GTO] Go to. Followed by nn causes calculator to stop execution, search through program memory to designated line and resume execution there **(page 36).**

Programming Key Index (Continued)

PROGRAM Mode	Automatic RUN Mode	
Active keys:	**Pressed from keyboard:**	**Executed as a recorded program instruction:**
	[RTN] Return. Sets calculator to line 00 of program memory **(page 33).**	[RTN] Return. If executed as a result of pressing [GSB] or execution of a [R/S] instruction, stops execution and returns control to keyboard. If executed as a result of a programmed [GSB] instruction, returns control to next line after the [GSB] instruction **(page 60).**
CLEAR [PRGM] Clear program. Clears program memory to all [GTO] 00 instructions and sets calculator to line 00 **(page 14).**	CLEAR [PRGM] After a prefix key, cancels that key. After other keys, does nothing. Does not disturb program memory or calculator status. Sets calculator to line 00 **(page 14).**	
		[PAUSE] Stops program execution and displays contents of X-register for approximately 1 second, then resumes program execution **(page 38).**

Programming Key Index (Continued)

PROGRAM Mode	Automatic RUN Mode	
Active keys:	**Pressed from keyboard:**	**Executed as a recorded program instruction:**
[BST] Back step. Moves calculator back one line in program memory **(page 35).**	[BST] Back step. Sets calculator to and displays line number and keycode of previous program memory line when pressed; displays original contents of X-register when released. No instructions are executed **(page 35).**	[x≠y] [x=y] [x>y] [x≤y] [x≠0] [x=0] [x>0] [x<0] Conditionals. Each tests value in X-register against 0 or value in Y-register as indicated. If true, calculator executes instruction in next line of program memory. If false, calculator skips one line before resuming execution **(page 49).**

Programming Key Index (Continued)

PROGRAM Mode	Automatic RUN Mode	
Active keys:	**Pressed from keyboard:**	**Executed as a recorded program instruction:**
[SST] Single step. Moves calculator forward one line in program memory **(page 35).**	[SST] Single step. Displays line number and keycode of current program memory line when pressed; executes instruction, displays result, and moves calculator to next line when released **(page 35).**	
	[R/S] Run/stop. Begins execution from current line of program memory. Stops execution if program is running **(page 36).**	[R/S] Run/stop. Stops program execution **(page 36).**
CLEAR [PREFIX] After [f], [g], [STO], [RCL], [FIX], [SCI], [ENG], [GSB], or [GTO], cancels that key.	CLEAR [PREFIX] After [f], [g], [STO], [RCL], [FIX], [SCI], [ENG], [GSB], or [GTO], cancels that key.	

Programming Key Index (Continued)

PROGRAM Mode	Automatic RUN Mode	
Active keys:	**Pressed from keyboard:**	**Executed as a recorded program instruction:**
	[GSB] Go to subroutine. Followed by line number 01-49, causes calculator to start executing instructions beginning with designated line **(page 60).**	[GSB] Go to subroutine. Followed by line number 01-49 causes calculator to go to designated line and execute that section of program memory as a subroutine **(page 60).**
		[NOP] No operation. Calculator executes no operation and continues program execution **(page 73).**
	Any key. Pressing any key on the keyboard stops execution of a running program **(page 40).**	

Section 1

Meet the HP-33E

Congratulations!

Your HP-33E Programmable Scientific Calculator is a truly unique and versatile calculating instrument. Using the Hewlett-Packard RPN logic system, your calculator can slice through the most difficult equations with ease. It is without parallel:

As a scientific calculator. The HP-33E features a multiple-entry keyboard with each of the keys controlling up to three separate operations, ensuring maximum computing power.

As a problem-solving machine. Following simple, step-by-step instructions in the *HP-33E Applications* books, you can key in any of dozens of programs from the areas of mathematics, statistics, games, finance, surveying, and other fields and begin using your calculator. Immediately.

As a personal programmable calculator. The HP-33E is so easy to program and use that it requires no prior programming experience or knowledge of mysterious programming languages. Yet even computer experts can appreciate the sophisticated programming features of the calculator:

- 8 data storage registers.
- 49 lines of program memory.
- Fully merged prefix and function keys that mean more programming power per line.
- Easy-to-use editing features for correcting and modifying programs.
- Powerful unconditional and conditional branching.
- 3 levels of subroutines.

And in addition, the HP-33E can be operated from its rechargeable battery pack for *complete portability,* anywhere.

If you are new to HP calculators and their RPN logic system, you may want to carefully work through *Solving Problems With Your Hewlett-Packard Calculator* before consulting this handbook. Even if you already own another HP calculator, you will find some new features in the problem solving book.

Now let's take a closer look at your calculator to see how easy it is to use, whether we solve a problem manually or use its programming power to solve the problem automatically.

Manual Problem Solving

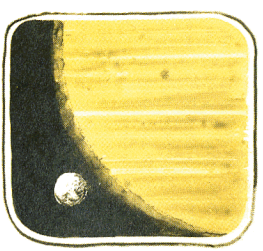
Before proceeding you should be comfortable solving problems manually. If not, refer to the Getting Started section of *Solving Problems With Your Hewlett-Packard Calculator*.

To see the close relationship between the manual solution to a problem and a programmed solution, let's solve a problem manually, and then use a program to solve the same problem and others like it.

If you were to calculate the surface area of a sphere, you would use the formula $A = \pi d^2$ where:

A is the surface area of the sphere.
d is the diameter of the sphere.
π is the value of pi, 3.14592654.

Example: Ganymede, one of Jupiter's 12 moons, has a diameter of 3,200 miles. You can use the calculator to manually compute the surface area of Ganymede. Merely press the following keys in order.



Keystrokes	Display	
3200	*3,200.*	Diameter of Ganymede.
[g] [x²]	*10,240,000.00*	Square of the diameter.
[g] [π]	*3.1416*	The quantity π.
[×]	*32,169,908.78*	Surface area of Ganymede in square miles.

Programmed Problem Solving

After calculating the surface area of Ganymede, suppose you decided you wanted to calculate the surface area of *each* moon. You could repeat the procedure you used for Ganymede 12 times, using a different diameter d each time. However, an easier and faster method is to create a *program* that will calculate the surface area of any sphere from its diameter rather than pressing all the keys for each moon.

To calculate the area of a sphere using a program, you should first *write* the program, then you must *load* the program into the calculator, and finally you *run* the program to calculate each answer.

Writing the Program. You have already written it! A program is nothing more than the series of keystrokes you would execute to solve the problem manually.

Loading the Program. To load the keystrokes of the program into the calculator:

1. Slide the PRGM-RUN switch PRGM RUN to PRGM (program).
2. Press [f] CLEAR [PRGM] to clear program memory.
3. Press the following keys in order. (When you are loading a program, the display gives you information that you will find useful later, but which you can ignore for now.)

Keystrokes	
[g] [x²] [g] [π] [×]	These are the same keys you pressed to solve the problem manually.
[GTO] 00	Returns the calculator to top of program memory and halts execution. The program can then be run again with new data.

The calculator will now remember this keystroke sequence.

Running the Program. To run the program to find the area of any sphere from its diameter:

1. Slide the PRGM-RUN switch PRGM RUN to RUN.
2. Press [g] [RTN] to set calculator to top of program memory.
3. Key in the value of the diameter.
4. Press [R/S] *(run/stop)* to run the program.

When you press [R/S], the sequence of keystrokes you loaded is automatically executed by the calculator, giving you the same answer you would have obtained manually.

For example, to calculate the surface area of Ganymede with a diameter of 3,200 miles:

Keystrokes	Display	
3200	*3,200.*	
[R/S]	*32,169,908.78*	Square miles.

With the program you have loaded, you can now calculate the surface area of any of Jupiter's moons—in fact, of *any* sphere—using its diameter. Leave the calculator in RUN mode and key in the diameter of each sphere for which you want the surface area, then press [R/S].

For example, compute the surface area of Jupiter's moon Io with a diameter of 2,310 miles.

Keystrokes	Display	
2310 [R/S]	*16,763,852.56*	Square miles.

Now compute the surface areas for the moons Europa, diameter 1,950 miles, and Callisto, diameter 3,220 miles.

Keystrokes	Display	
1950 [R/S]	*11,945,906.07*	Area of Europa in square miles.
3220 [R/S]	*32,573,289.27*	Area of Callisto in square miles.

Programming is *that* easy! The calculator remembers a series of keystrokes and then executes them whenever you wish. In fact, your HP-33E can remember up to 49 separate operations (and many more keystrokes since many operations require two or three keystrokes).

Section 2

Specific Features of the HP-33E

Most of the features found on the HP-33E are discussed in *Solving Problems With Your Hewlett-Packard Calculator*. However, several features unique to the HP-33E (or new to HP calculators) are discussed in the following pages.

Self-Check Routine

Your new Hewlett-Packard calculator is loaded with features that make it easy to use and give you confidence that the answers you calculate are right, every time. The self-check routine, a feature found on many sophisticated electronic instruments and computers, was designed for just those reasons. We don't expect you to ever have a problem with your calculator, but if you think that it isn't operating properly, try this:

Keystrokes	Display
STO ENTER↑	-8,8,8,8,8,8,8,8,8,8

The display shown above will appear if your calculator is operating properly. Press any key to clear the display back to zero. If your calculator is not operating properly, your display will show *Error 9* or an erroneous display. This tells you that a problem exists in the calculator's circuitry and you should send it in for service (refer to Shipping Instructions in this owner's handbook). Pressing any key will replace the *Error 9* in the display with a number that tells a Hewlett-Packard Service Engineer which circuit in the calculator is at fault. That's right, the calculator not only tells you it's having problems, it tells us where the problem is, so we can fix it as quickly and inexpensively as possible and return it to you without delay.

> **Note:** Using the self-check routine causes all memory to be cleared, including the stack, data registers, and program memory.

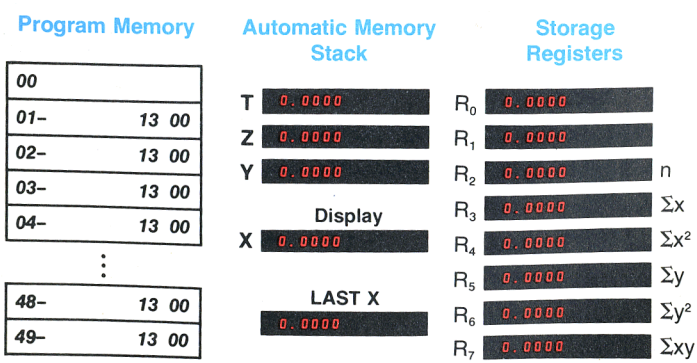
Mantissa

When in any display format you wish to view the contents of the true mantissa (all decimal places), press [9] [MANT] and hold the key down. This operation displays all 10 digits of the mantissa held internally. Release the key and the display will revert back to its original contents. For more information on display formatting refer to Display Control in *Solving Problems With Your Hewlett-Packard Calculator*.

Storage Registers

In addition to the power that exists in the four-register automatic memory stack and LAST X register, your HP-33E contains eight addressable data storage registers and 49 lines of program memory. Storage registers R_2 through R_7 also serve as statistical storage registers.

For information concerning data storage registers refer to Storing and Recalling Numbers in *Solving Problems With Your Hewlett-Packard Calculator*. Statistical registers and programming are discussed later in this handbook.



Number Alteration Keys

Besides [CHS], your HP-33E has three other keys provided for altering numbers—[ABS], [INT], and [FRAC]. These keys are often used in programs for manipulating numbers.

Absolute Value

Some calculations require the absolute value or magnitude of a number. To obtain the absolute value of the number in the displayed X-register, press the [g] prefix key followed by the [ABS] *(absolute value)* key. For example, to calculate the absolute value of −3.

Keystrokes	Display	
3 [CHS]	*-3.*	
[g] [ABS]	*3.0000*	$\lvert -3 \rvert$

Integer Portion of a Number

To extract and display the integer portion of a number, press the [g] prefix key followed by the [INT] *(integer)* key. For example, to display only the integer portion of the number 123.456:

Keystrokes	Display	
123.456	*123.456*	
[g] [INT]	*123.0000*	Only the integer portion remains.

When [g] [INT] is pressed, the fractional portion of the number is lost. The entire number, of course, is preserved in the LAST X register.

Fractional Portion of a Number

To extract and display only the fractional portion of a number, press the [g] prefix key followed by the [FRAC] *(fraction)* key. For example, to see the fractional portion of the 123.456 used above:

Keystrokes	Display	
f LST x	123.4560	Summons the original number back to the X-register.
g FRAC	0.4560	Only the fractional portion of the number is displayed, rounded here to FIX 4 display.

When g FRAC is pressed, the integer portion of the number is lost. The entire number, of course, is preserved in the LAST X register.

Statistical Functions

Accumulations

Executing the Σ+ *(summation)* function automatically gives you several different sums and products of the values in the X- and Y-registers at once. In order to make these values accessible for sophisticated statistics problems, they are automatically placed by the calculator into storage registers.

Before you begin any calculations using the Σ+ key, you should first clear the storage registers used in accumulations by executing the f CLEAR REG *(clear registers)* function.

When you key a number into the calculator and press the Σ+ key, each of the following operations is performed:

1. The number 1 is added to the contents of register R_2. After all of the following steps are performed, the total number in R_2 is placed in the displayed X-register. The number that you keyed into the X-register is preserved in the LST x register.
2. The number in the X-register is added to the contents of register R_3.
3. The square of the number in the X-register is added to the contents of register R_4.
4. The number in the Y-register of the stack is added to the contents of register R_5.

5. The square of the number in the Y-register is added to the contents of register R_6.
6. The number in the X-register is multiplied by the number in the Y-register, and the product is added to the contents of register R_7.

Thus, each press of the [Σ+] key updates these sums and products. The contents of the displayed X-register and the applicable storage registers are as follows:

Register	Data	
Displayed X	n	Number of entries.
R_2	n	Number of entries.
R_3	Σx	Summation of x values.
R_4	Σx^2	Summation of x^2 values.
R_5	Σy	Summation of y values.
R_6	Σy^2	Summation of y^2 values.
R_7	Σxy	Summation of products of x and y values.

In addition, the y-value present before the last press of the [Σ+] key is retained in the Y-register, while the x-value present before [Σ+] was pressed is retained in the LAST X register. To see any of the summations at any time, you have only to recall the contents of the desired storage register. In the case of the [Σ+] key, recalling storage register contents or keying in a number simply writes over the number of entries *(n)* that is displayed. The stack does not lift.

Example: Find Σx, Σx^2, Σy, Σy^2 and Σxy for paired values of x and y listed below.

y	7	5	9
x	5	3	8

Keystrokes	Display	
[f] CLEAR [REG]	*0.0000*	Clears storage registers. (Assumes stack cleared to zeros.)
7 [ENTER↑]	*7.0000*	
5 [Σ+]	*1.0000*	First pair is accumulated; n = 1.

Keystrokes	Display	
5 [ENTER↑]	5.0000	
3 [Σ+]	2.0000	Second pair is accumulated; n = 2.
9 [ENTER↑]	9.0000	
8 [Σ+]	3.0000	Third pair is accumulated; n = 3.
[RCL] 2	3.0000	Number of entries from R_2, (n = 3).
[RCL] 3	16.0000	Sum of x values in R_3.
[RCL] 4	98.0000	Sum of squares of x values in R_4.
[RCL] 5	21.0000	Sum of y values in R_5.
[RCL] 6	155.0000	Sum of squares of y values in R_6.
[RCL] 7	122.0000	Sum of products of x and y values in R_7.

Mean

The [x̄] function is used to calculate the mean (arithmetic average) of x and y values accumulated in the statistical registers.

When you press [g] [x̄]:

1. The mean of x is calculated using the data accumulated in the registers that contain n and Σx (R_2 and R_3). The resultant value for the mean of x is placed in the display and the X-register.
2. The mean of y is calculated using the data accumulated in the registers that contain n and Σy (R_2 and R_5). The resultant value for mean of y is placed in the Y-register. Simply press [x≷y] to bring that value into the X-register for use.

The easiest way to accumulate the data required for the [x̄] function is by using the [Σ+] function as described above.

Standard Deviation

The [s] function is used to calculate the sample standard deviation (a measure of dispersion around the mean) of data accumulated in the statistical registers. When you press [g] [s]:

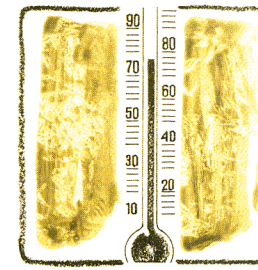
1. The sample standard deviation of x is calculated using data

accumulated in the statistical registers containing n, Σx, and Σx^2 (R_2, R_3, and R_4). The resultant x value is placed in the X-register and display.

2. The sample standard deviation of y is calculated using data accumulated in the statistical registers containing n, Σy, Σy^2 (R_2, R_5, and R_6). The resultant y value is placed in the Y-register. Simply press [x≷y] to place the y value in the X-register for use.

Again, as in the use of [x̄], the easiest way to accumulate the required data in the statistical registers is by using the [Σ+] function.

Example: Below is a chart of daily high and low temperatures for a winter week in Fairbanks, Alaska. What are the average high and low temperatures and the standard deviation of the high and low temperatures for the week selected?



	Sun	Mon	Tue	Wed	Thur	Fri	Sat
High	6	11	14	12	5	−2	−9
Low	−22	−17	−15	−9	−24	−29	−35

Keystrokes	Display	
[f] CLEAR [REG]	*0.0000*	Clears storage registers. (Assumes stack cleared to zeros.)
6 [ENTER↑] 22 [CHS]	*−22.*	
[Σ+]	*1.0000*	First entry. Number of data pairs is now 1.
11 [ENTER↑] 17 [CHS]	*−17.*	
[Σ+]	*2.0000*	Second entry. Number of pairs is now 2.
14 [ENTER↑] 15 [CHS]	*−15.*	
[Σ+]	*3.0000*	

Keystrokes	Display	
12 [ENTER↑] 9 [CHS]	-9.	
[Σ+]	4.0000	
5 [ENTER↑] 24 [CHS]	-24.	
[Σ+]	5.0000	
2 [CHS] [ENTER↑]	-2.0000	
29 [CHS] [Σ+]	6.0000	
9 [CHS] [ENTER↑]	-9.0000	
35 [CHS] [Σ+]	7.0000	Number of pairs is now 7 (n = 7).
[g] [x̄]	-21.5714	Average low temperature (mean of x) is in displayed X-register.
[x≷y]	5.2857	Average high temperature (mean of y) is placed in the displayed X-register.
[g] [s]	8.7912	Standard deviation of low temperatures (x values) is in the displayed X-register.
[x≷y]	8.2808	Standard deviation of high temperatures (y values) is placed in the displayed X-register.

Deleting and Correcting Data

If you key in an incorrect value and have not executed [Σ+], press [CLx] to delete the incorrect number or digits, then key in the correct number.

If one of the values is changed, or if you discover that one of the values is in error after you have executed the [Σ+] function, you can correct the summations by using the [Σ−] *(summation minus)* function as follows:

1. Key the incorrect data pair into the X- and Y-registers. (You can use [LST x] to return a single incorrect data value to the displayed X-register.)
2. Press [f] [Σ−] to delete the incorrect data.
3. Key in the correct values for x and y. (If one value of an x, y data pair is incorrect, both values must be deleted and reentered.)
4. Press [Σ+].

The corrected values for mean and standard deviation are now obtainable by executing the $\bar{x}$ and s functions.

For example, suppose that you discover a recording error in the data you have gathered on the temperatures in Fairbanks, and you find that the Monday high and low values are actually 5 and −19. To account for the change in mean and standard deviation values:

Keystrokes	Display	
11 ENTER↑	11.0000	The incorrect y value.
17 CHS	−17.	The incorrect x value.
f Σ−	6.0000	The incorrect values have been deleted and the number of entries is now 6 (n = 6).
5 ENTER↑	5.0000	The correct y value.
19 CHS	−19.	The correct x value.
Σ+	7.0000	The correct values have been summed. The number of entries is now 7.
g $\bar{x}$	−21.8571	The correct mean of the low temperatures (mean of x).
x≷y	4.4286	The correct mean of high temperatures (mean of y).
g s	8.6492	The correct standard deviation of the low temperatures (x values).
x≷y	7.8921	The correct standard deviation of the high temperatures (y values).

Linear Regression

Linear regression is a statistical method for finding a straight line that best fits a set of data points, thus providing a relationship between two variables. If there is equal time or space between data points, then this is called a trend line. Because of the calculation technique used, linear regression is often referred to as a *least squares fit*.

Naturally, at least two data points must be placed in the machine before a line can be drawn or fitted to them. After you have totaled the data points using the [Σ+] key, you can calculate the coefficients of the linear equation

$$y = Ax + B$$

by pressing [f] [L.R.]. B is the y-intercept and appears in the display; A represents the slope of the line and is stored in the Y-register.

Example: Big Lyle Turncoat, owner-operator of the Turncoat Oil Company, wishes to know the slope and y-intercept of a least squares line for the consumption of motor fuel in the United States against time since 1945. This will help him determine motor fuel demand. He knows the data given in the table below:

Motor Fuel Demand (Millions of Barrels)	969	994	1330	1512	1750	2162	2385 (est)
Year	1945	1950	1955	1960	1965	1970	1975

Solution: Turncoat *could* draw a plot of motor fuel demand against time. However, with his HP-33E, he needs only to key in the data using the [Σ+] operation, then press [f] [L.R.].

Keystrokes	**Display**	
[f] CLEAR [REG]	*0.0000*	Clears registers. (Assumes stack is cleared to zeros.)
696 [ENTER↑]	*696.0000*	
1945 [Σ+]	*1.0000*	
994 [ENTER↑]	*994.0000*	
1950 [Σ+]	*2.0000*	
1330 [ENTER↑]	*1,330.0000*	
1955 [Σ+]	*3.0000*	
1512 [ENTER↑]	*1,512.0000*	

Keystrokes	Display	
1960 Σ+	*4.0000*	
1750 ENTER↑	*1,750.0000*	
1965 Σ+	*5.0000*	
2162 ENTER↑	*2,162.0000*	
1970 Σ+	*6.0000*	
2385 ENTER↑	*2,385.0000*	
1975 Σ+	*7.0000*	All data pairs have been keyed in.
f L.R.	*-107,975.0000*	The y-intercept of line.
x≷y	*55.8786*	Slope of line.

Linear Estimate

With the data totaled in registers R_2 through R_7, a predicted y (that is, $\hat{y}$) can be calculated by keying in a new x-value and pressing $\hat{y}$. A predicted x (that is, $\hat{x}$) can be calculated by keying in a new y-value and pressing $\hat{x}$.

For example, with the data intact from the previous example, if Turncoat wished to predict the demand for oil for the years 1980 and 2000, he has only to key in the new x-values and press $\hat{y}$.

Keystrokes	Display	
1980 f $\hat{y}$	*2,664.5714*	Predicted demand in millions of barrels for the year 1980.
2000 f $\hat{y}$	*3,782.1429*	Predicted demand in millions of barrels for the year 2000.

With the data still intact from before, if Turncoat now wishes to predict in what years will the predicted demand in millions of barrels of oil be 6,000 and 10,000 millions of barrels of oil, all he needs to do is key in new y-values and press $\hat{x}$.

Keystrokes	Display	
6000 f $\hat{x}$	*2,039.6907*	Predicted year — 2039.
10000 f $\hat{x}$	*2,111.2744*	Predicted year — 2111.

Correlation Coefficient

To establish how well the data used fits the linear regression, you may want to calculate the correlation coefficient, r, by using [f] [r] *(correlation coefficient)*.

The correlation coefficient is a value between -1 and 1. At $r = 0$ you have no fit, while at $r = \pm 1$ you have a perfect fit.

Example: Calculate the correlation coefficient for the previously calculated linear regression.

Keystrokes	Display	
[f] [r]	*0.9967*	A very good fit.

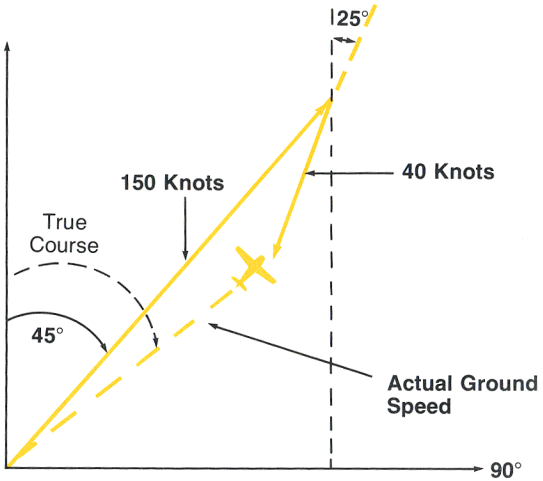
Vector Arithmetic

You can use your HP-33E to add or subtract vectors by combining the polar/rectangular conversion functions (the [g] [→P] and the [f] [→R] keys) with the summation functions (the [Σ+] and [f] [Σ−] keys).

To use *only* the Σx and Σy that you have accumulated in the storage registers, you can press [RCL] followed by [Σ+]. This brings Σx into the displayed X-register and Σy into the Y-register, overwriting the contents of those two stack registers. The stack does not lift. This feature is particularly useful when performing vector arithmetic.

Example: On his way to search for an albino caribou, grizzled bush pilot Apeneck Sweeney's converted Swordfish aircraft has a true air speed of 150 knots and an estimated heading of 45°. The Swordfish is also being buffeted by a head wind of 40 knots from a bearing of 25°. What is the actual ground speed and true course of the Swordfish?

Method: The true course and actual ground speed are equal to the difference of the vectors. (North becomes the x-coordinate so that the problem corresponds with navigational convention.)



Keystrokes	Display	
f CLEAR REG	*0.0000*	Clears storage registers. (Display assumes no results remain from previous examples.)
g DEG	*0.0000*	Sets degrees mode.
45 ENTER↑	*45.0000*	θ for 1st vector is entered to Y-register.
150	*150.*	r for 1st vector is keyed in.
f →R	*106.0660*	Converted to rectangular coordinates.
Σ+	*1.0000*	1st vector coordinates accumulated in storage registers R_3 and R_5.

Keystrokes	Display	
25 [ENTER↑]	*25.0000*	θ for 2nd vector is entered to Y-register.
40	*40.*	r for 2nd vector is keyed in.
[f] [→R]	*36.2523*	Converted to rectangular coordinates.
[f] [Σ−]	*0.0000*	2nd vector rectangular coordinates subtracted from those of 1st vector.
[RCL] [Σ+]	*69.8137*	Recalls both R_3 and R_5.
[g] [→P]	*113.2417*	Actual ground speed of the Swordfish in knots.
[x≷y]	*51.9389*	True course of the Swordfish in decimal degrees.

Section 3

Simple Programming

What Is a Program?

A program is nothing more than a series of calculator keystrokes that you would press to solve a problem manually. The HP-33E remembers these keystrokes when you key them in, then executes them in order whenever you wish. No prior programming experience is necessary for programming your HP-33E.

Why Write Programs?

Programs can save you time on repetitive calculations. Once you have written the keystrokes procedure for solving a particular problem and recorded it in the calculator, you no longer need to devote attention to individual keystrokes that make up the procedure. You can let the calculator solve each problem for you. You saw this earlier when we computed the surface area of the moons of Jupiter.

And because you can easily check the procedure in your program, you have more confidence in your final answer since you don't have to worry each time about whether or not you have pressed an incorrect key.

Keycodes

Set the PRGM-RUN switch PRGM  RUN to PRGM. Press the first keys f CLEAR PRGM 9 x^2 of the surface area of a sphere program (refer to page 14) and the display will change to:

01- 15 0

The two-number code *01-* that has appeared on the left side of the display designates the line number of program memory that is being displayed.

The first digit denotes the row of the key. The second digit denotes the number of the key in that row. Thus, the keycode *15* represents the key in the first row on the calculator and the fifth key in that row: the 9 key.

The *0* represents x^2 which is located on the zero digit key.

The numbers at the right of the display (*15 0*) designate the key stored in that line at program memory (g x^2). Each key on the keyboard except digit keys, has a two-digit keycode. Digit keys are coded 0 through 9. All other keys are coded by their position on the keyboard.

This handy matrix system allows you to easily determine the code for each instruction without using a reference table.

In every case, a single operation (e.g. f SIN , STO + 1, $x \rightleftarrows y$) uses only one line of program memory.

> **Note:** Each operation, prefixed or not, requires only one line of program memory.

The keys for finding the area of a sphere and their corresponding displays are shown below. Press each key in turn and verify the keycode shown in the display.

Keystrokes	Display	
f CLEAR PRGM	*00*	Clears program memory.
g x^2	*01- 15 0*	
g π	*02- 15 73*	
×	*03- 61*	
GTO 00	*04- 13 00*	

In this case, a program consisting of eight keystrokes takes only four lines of program memory.

Problems:

1. What would be the keycodes for the following operations: g $\frac{1}{x}$, g GRD, f →H.MS, STO + 1? (Answers: *15 3, 15 23, 14 6, 23 51 1*).

2. How many lines of program memory would be required to load the following sections of programs?
   a. 2 ENTER↑ 3 + .
   b. 10 STO 6 RCL 6 × .
   c. 100 STO 1 50 STO × 1 RCL 2 g π × .
   (Answers: a. 4, b. 5, c. 10).

Introductory Program

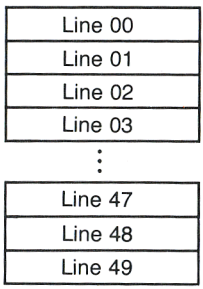
The area of a sphere program you wrote, recorded, and executed earlier showed you that the sequence of keystrokes used to solve a problem manually is the same sequence used in a program. Now let's return our attention to that program to explain the information displayed in PRGM mode.

First, set the PRGM-RUN switch PRGM RUN to PRGM so that the sequence of keystrokes will be recorded for later execution. Second, press f CLEAR PRGM to clear the calculator of previous programs. The display will show:

00

This tells you that you are at the beginning of program memory. Line 00 contains an automatic stop instruction and cannot be used to record your program keystrokes. Program keystrokes are recorded in lines 01 through 49. (See figure below.)

Program Memory



As you can see, the program memory for the HP-33E is separate from the four stack registers, the LAST X register, and the eight storage registers (see page 4a).

With *00* displayed in PRGM mode, you are ready to key in your program. Surface area of a sphere is calculated using the formula $A = \pi d^2$. The short list of keystrokes for the surface area of a sphere program is shown on the following page:

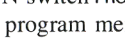
Keystrokes	Comments
[g] [x²]	These keys square the diameter.
[g] [π]	These keys place π in the X-register.
[×]	This key multiplies d^2 by π.
[GTO] 00	

Running a Program

To run a program, you have only to:

1. Set the calculator to RUN mode.
2. Press [g] [RTN] to set calculator to top of program memory.
3. Key in any needed data.
4. Press [R/S] to execute the program.

For example, to use the program now in the calculator to solve for the surface area of spheres with diameters of 3 inches, 6 meters, and 9 miles:

First set the calculator's PRGM-RUN switch PRGM  RUN to RUN. Then press [g] [RTN] to go to top of program memory.

Keystrokes	Display	
3 [R/S]	*28.2743*	Square inches.
6 [R/S]	*113.0973*	Square meters.
9 [R/S]	*254.4690*	Square miles.

Writing a Second Program

To further explore the programming capabilities of your HP-33E, let's write a second program. Suppose you want to write a program that will calculate the increase in volume of a spherical balloon as its diameter increases using the formula:

$$\text{Increase in volume} = {}^1/_6\,\pi\,(d_1{}^3 - d_0{}^3),$$

where d_0 is the original diameter of the balloon and d_1 is the new diameter. If d_0 were entered in the Y-register and d_1 were keyed into

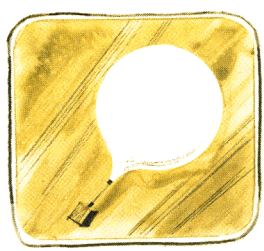
the X-register, the problem could be solved manually by pressing the keys shown in the left-hand column below. The program keystrokes for this problem are the same as the manual keystrokes. Switch to PRGM mode and press the keys shown below.

Keystrokes	Display	
f CLEAR PRGM	00	
ENTER↑	01– 31	
3	02– 3	
f y^x	03– 14 3	Cube the new diameter.
$x \gtrless y$	04– 21	
3	05– 3	
f y^x	06– 14 3	Cube the original diameter.
−	07– 41	Subtract the cubes.
g π	08– 15 73	
×	09– 61	Multiply by π.
6	10– 6	
÷	11– 71	Divide by 6.
GTO 00	12– 13 00	

Notice that you had to load the ENTER↑ key as an instruction in this program. The ENTER↑ instruction here separates the number 3 in the second line of the program from the digits for the new diameter you will key in later.

To run the program, switch to RUN mode and press g RTN (or GTO 00) so that the calculator will begin execution from line 00. Then try the following example.

Example: Find the increase in volume of a spherical balloon if the diameter changes from 30 feet to 35 feet.



Keystrokes	Display	
30 [ENTER↑]	*30.0000*	Enter original diameter into Y.
35 [R/S]	*8,312.1306*	Key new diameter into X and run the program. The answer is displayed in cubic feet.

Displaying Each Step

In order to look at this program, you need to be able to display each line. Two operations allow you to step through program memory: [SST] *(single step)* and [BST] *(back step)*.

With the increase in sphere volume program still recorded in the calculator, set the PRGM-RUN switch PRGM  RUN to RUN and press [g] [RTN] to reset the calculator to line 00. Then switch to PRGM mode and press [SST] once. The display will change to:

01– 31

Press [SST] again and the display will change to:

02– 3

Now press [g] [BST]. You can see what has happened. You are back at program memory line 01. Press [g] [BST] again and line 00 is displayed. Pressing [g] [BST] again does nothing.

[SST] displays the contents of the *next* line of program memory. [g] [BST] displays the contents of the *previous* line of program memory.

Of course because these two keys work in PRGM mode, neither can be stored in program memory.

Going to a Line Number

You can see that if you wanted to single step forward from line 00 to some remote line number in program memory, it would take a great deal of time and a number of presses of the [SST] key. To make it easy for you, the HP-33E gives you another nonrecordable operation, [GTO] • nn, that permits you to go to any line number of program memory.

Whether the PRGM-RUN switch is set to PRGM or to RUN, when you press [GTO] . nn, the calculator immediately jumps to the program memory line number specified by the two-digit number nn. No instructions are executed. If in RUN mode, you can momentarily slide the PRGM-RUN switch PRGM RUN to PRGM to view this line of program memory. If the calculator is already in PRGM mode, the line number and keycode for the instruction contained in that line are displayed. If you now press [R/S] in RUN mode, execution will then begin with that line of program memory. Loading of new keystrokes will begin with the next line of program memory.

Programmed Stops

When programming, there may be occasions when you want a program to halt during execution so that you can key in data. Or you may want the program to pause so that you can quickly view results before the program automatically resumes running. Two keys, [R/S] *(run/stop)* and [PAUSE] *(pause),* are used for program interruptions.

Stopping During Program Execution

The [R/S] *(run/stop)* function can be used either as an instruction in a program or as an operation pressed from the keyboard.

When pressed from the keyboard:

1. If a program is running, [R/S] stops the program.
2. If a program is stopped or not running, and the calculator is in RUN mode, [R/S] starts the program running beginning with the current line in program memory.

When executed as an instruction during a running program, [R/S] stops program execution after the current line of program memory. If [R/S] is then pressed from the keyboard, execution begins with the current line of program memory. (When [R/S] is pressed and held in RUN mode, it displays the line number and keycode of that current line—when released, execution begins with that line.)

You can use these features of the [R/S] instruction to stop a running program at points where you want to key in data. After the data has been keyed in, restart the program using the [R/S] key from the keyboard.

Example: Universal Tins, a canning company, needs to calculate the volumes of various cylindrically-shaped cans. Universal would also like to be able to record the area of the base of each can before the volume is calculated.

The program below calculates the area of the base of each can and then stops. After you have written down the result, the program can be re-started to calculate the final volume. The formula used is:

$$\text{Volume} = \text{base area} \times \text{height} = \pi r^2 \times h$$

The radius *(r)* and the height *(h)* of the can are keyed into the X- and Y-registers, respectively, before the program is run.

To record this program, set the PRGM-RUN switch PRGM RUN to PRGM then key in the following list of keys.

Keystrokes	Display	
f CLEAR PRGM	*00*	Clears program memory and displays line 00.
g x^2	*01– 15 0*	Square the radius.
g π	*02– 15 73*	Place π in X.
×	*03– 61*	Calculate the area of the base.
R/S	*04– 74*	Stop to record the area.
×	*05– 61*	Calculate the final volume.
GTO 00	*06– 13 00*	

In order to run this program, set the PRGM-RUN switch PRGM RUN to RUN and press g RTN so the program will begin execution from line 00.

Then use the program to complete the table below:

Height	Radius	Area of Base	Volume
25	10	?	?
8	4.5	?	?

Keystrokes	Display	
25 [ENTER↑]	*25.0000*	Enter the height into the Y-register.
10 [R/S]	*314.1593*	Program stops to display the area.
[R/S]	*7,853.9816*	Volume of first can is calculated.
8 [ENTER↑]	*8.0000*	Enter the height into the Y-register.
4.5 [R/S]	*63.6173*	Program stops to display the area.
[R/S]	*508.9380*	Second volume is calculated.

With the height in the Y-register and the radius in the X-register, pressing [R/S] in automatic RUN mode calculates the area of the can's base; the program stops at the first [R/S] instruction encountered. Pressing [R/S] again calculates the volume of the can and program execution stops at line 00.

In general, you should record [R/S] into a program when you need to display *more* than one answer. To display *only one* answer of the final answer of a series, the [GTO] 00 instruction in a program is more convenient since the calculator ends execution at line 00, ready to begin again.

Pausing During Program Execution

An [f] [PAUSE] instruction executed in a program interrupts program execution to display results momentarily before execution is resumed. The length of the pause is about one second, but you can use more than one consecutive [f] [PAUSE] instruction to lengthen the time.

To see how [f] [PAUSE] can be used in a program, we'll modify the cylinder volume program in the previous example. In the new program the area of the base will be briefly displayed before the volume is calculated. This example will also show how different programming approaches can be taken to solve the same problem.

To key in the program, set the PRGM-RUN switch PRGM RUN to PRGM and press f CLEAR PRGM to clear program memory and display line 00. Then key in the following list of keys.

Keystrokes	Display	
f CLEAR PRGM	00	
g x^2	01– 15 0	Squares the radius in X.
g π	02– 15 73	Places π in X.
×	03– 61	Calculates the area of the base.
f PAUSE	04– 14 74	Pauses to show the base area for one second.
×	05– 61	Calculates final volume of can.

This program also assumes the height has been entered into the Y-register and the radius has been keyed into the X-register. If you have stored the instructions, set the PRGM-RUN switch PRGM RUN to RUN and press g RTN so that the calculator will begin execution from line 00. Now complete the table below using the new program.

Height	Radius	Area of Base	Volume
20	15	?	?
10	5	?	?

Keystrokes	Display	
20 ENTER↑	20.0000	Enter the height into the Y-register.
15 R/S	706.8583	Area of base is displayed for one second.
	14,137.1669	Program stops, displaying the volume.
10 ENTER↑	10.0000	Enter the second height into Y.
5 R/S	78.5398	Area of base is displayed for one second.
	785.3982	Program stops, displaying the volume.

Program Stops

At times a mistake of some kind in your program will stop program execution. To help you identify why the calculator stopped in the middle of your program, possible reasons are listed below.

Executing a R/S. The execution of a [R/S] instruction in a program halts program execution at the line following the [R/S].

Executing Line 00. Whenever line 00 is executed in a program, program execution stops at line 00 unless it is part of a subroutine, in which case it does a return.

Pressing Any Key. Pressing any key halts program execution. Be careful to avoid pressing keys during program execution.

The calculator has been designed so that program execution will *not* halt in the middle of a digit entry sequence. If you press any key while a number is being placed in the X-register by a running program, the entire number will be "written" and the following step will be executed by the program before the program halts.

When a program is halted, you can resume execution by pressing [R/S] from the keyboard in RUN mode. When you press [R/S], the program resumes execution with the next step as though it had never stopped at all.

Error Stops

If the calculator attempts to execute any error-causing operation (see Error Indications) during a running program, execution immediately halts and the calculator displays the word ***Error*** and a number. To see the line number and keycode of the error-causing instruction, you can briefly set the calculator to PRGM mode.

Overflow Calculations. Your HP-33E has been designed so that by looking at the display you can always tell why the calculator stops. If program execution stops because the result of a calculation in the X-register is a number with a magnitude greater than $9.999999999 \times 10^{99}$, all 9's are displayed with appropriate sign. It is then easy to determine the operation that caused the overflow by switching to PRGM mode and identifying the keycode in the display.

If the overflow occurs in one of the storage registers, possibly the result of storage register arithmetic, the calculator will display ***Error 1*** to inform you of the overflow.

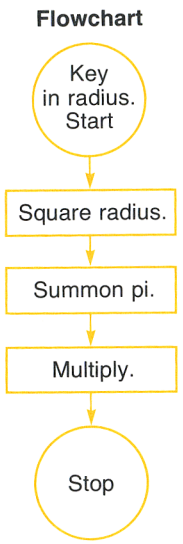
If the result of a calculation is a number with a magnitude less than $1.000000000 \times 10^{-99}$, zero will be substituted for the number and a running program will continue to execute normally.

Flowcharts

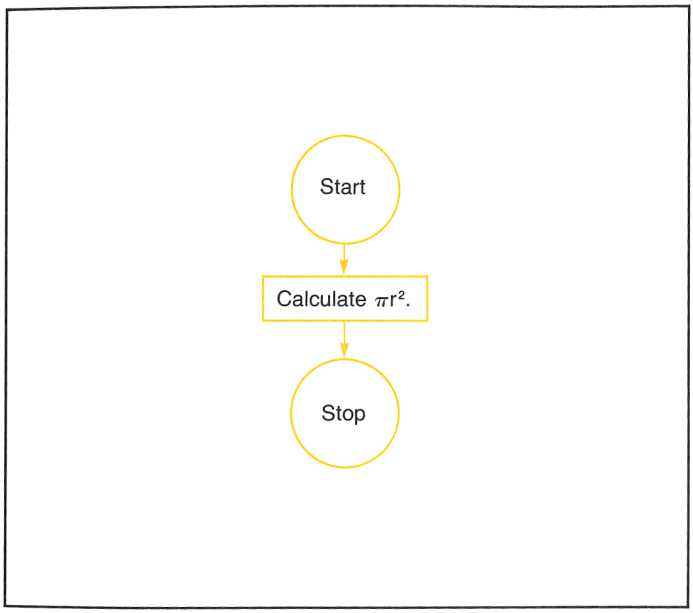
At this point, we digress for a moment from our discussion of the calculator itself to discuss a fundamental and extremely useful tool in programming—the flowchart.

A flowchart is an *outline* of the way a program solves a problem. With 49 possible instructions, it is quite easy to get "lost" while creating a long program, especially if you try to simply load the complete program from beginning to end with no breaks. A flowchart is a shorthand that can help you design your program by breaking it down into smaller groups of instructions. It is also very useful as documentation—a road map that summarizes the operation of a program.

A flowchart can be as simple or as detailed as you like. Here is a flowchart that shows the operations you executed to calculate the area of a circle according to the formula $A = \pi r^2$. Compare the flowchart to the actual instructions for the program:



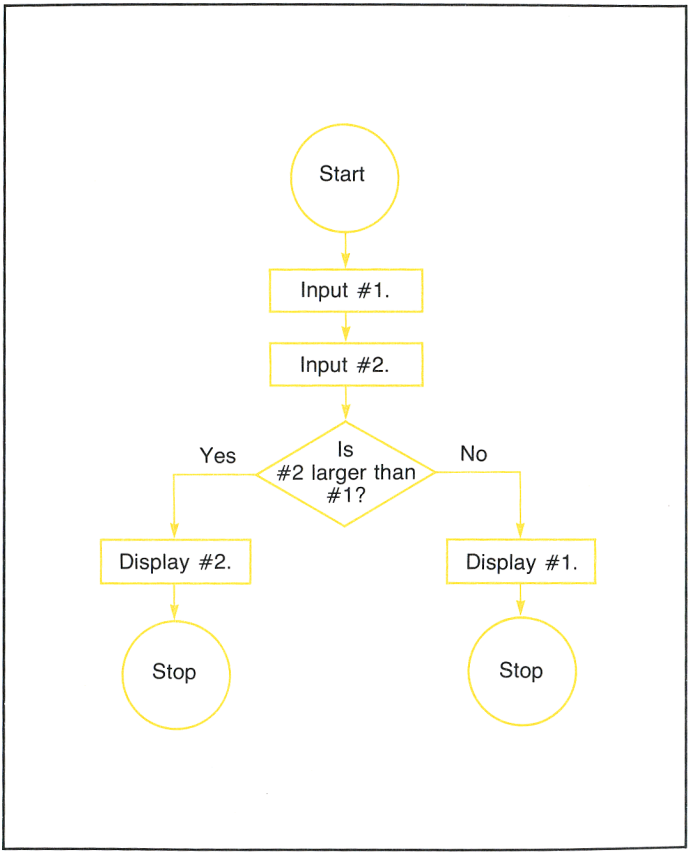
You can see the similarities. At times, a flowchart may duplicate the set of instructions exactly, as shown above. At other times, it may be more useful to have an entire group of instructions represented by a single block in the flowchart. For the program to calculate the area of a circle:



Here an entire group of instructions was replaced by one block in the flowchart. This is a common practice, and one which makes a flowchart extremely useful in visualizing a complete program.

You can see how a flowchart is drawn linearly, from the top of the page to the bottom. This represents the general flow of the program, from beginning to end. Although flowcharting symbols sometimes vary, throughout this handbook we have held to the convention of circles for the beginning and end of a program or routine, and rectangles to represent groups of functions that take an input, process it, and yield a single output. We have used a diamond to represent a *decision,* where a single input can yield either of two outputs.

For example, if you had two numbers and wished to write a program that would display only the larger, you might design your program by first drawing a flowchart that looks like the one shown on the following page:



After drawing the flowchart, you would go back and substitute groups of instructions for each element of the flowchart. When the program was loaded into the calculator and run, if #2 was larger than #1, the answer to the question "Is #2 larger than #1?" would be YES, and the program would take the left-hand path, display #2, and stop. If the answer to the question was NO, the program would execute the right-hand path, and #1 would be displayed. You will see later the many decision-making instructions available on your HP-33E.

As you work through this handbook, you will become more familiar with flowcharts. Use the flowcharts that illustrate the examples and problems to help you understand the many features of the calculator, and draw your own flowcharts to help you create, edit, eliminate errors in, and document your programs.

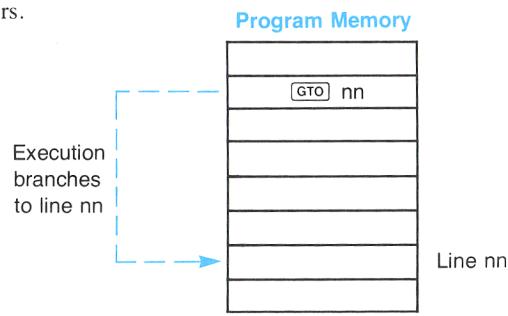
Section 4

Branching

Unconditional Branching and Looping

You have seen how the nonrecordable operation [GTO] . nn can be used from the keyboard to transfer execution to any line number of program memory. You can also use the [GTO] *(go to)* instruction as part of a program.

When the calculator is executing a program and encounters a [GTO] nn instruction, for example, it goes immediately to that line number. By using a [GTO] instruction in your program, you can transfer execution to any part of the program that you choose. When the calculator executes a [GTO] instruction, it begins execution again at the specified line it encounters.



A [GTO] instruction used this way is known as an *unconditional branch*. It *always* unconditionally branches execution from the [GTO] instruction to the specified line number. Later, you will see how a conditional instruction can be used in conjunction with a [GTO] instruction to create a *conditional* branch—a branch that depends on the outcome of a test.

A common use of a branch is to create a "loop" in a program. For example, the following program calculates and displays the square roots of consecutive whole numbers beginning with the number 1. Your HP-33E continues to compute the square root of the next consecutive whole number until you press [R/S] to stop program execution (or until the calculator overflows).

To key in the program:

1. Slide the PRGM-RUN switch PRGM RUN to PRGM.
2. Press f CLEAR PRGM to clear program memory and reset the calculator to line 00.

Keystrokes	Display	
f CLEAR PRGM	00	
0	01– 0	
STO 1	02– 23 1	
1	03– 1	
STO + 1	04–23 51 1	
RCL 1	05– 24 1	
f PAUSE	06– 14 74	Allows viewing of number before taking its square root.
f √x	07– 14 0	
f PAUSE	08– 14 74	Allows viewing of square root.
GTO 03	09– 13 03	

To run the program, slide the PRGM-RUN switch PRGM RUN to RUN and press g RTN R/S. The program will begin displaying whole numbers and their square roots and will continue to do so until you press R/S from the keyboard or until the calculator overflows.

How it works: The GTO 03 instruction causes the calculator to go to the line 03 instruction in the program. When it encounters that line, execution begins again from that point.

Since execution is transferred to the instruction in line 03 each time the calculator executes the GTO 03 instruction in line 09, the calculator will remain in this "loop," continually adding one to the number in storage register R_1 and displaying the new number and its square root.

Looping techniques like the one illustrated here are common and extraordinarily useful in programming. By using loops, you take advantage of one of the most powerful features of your HP-33E—the ability to update data and perform calculations automatically, quickly, and if you so desire, endlessly.

You can use unconditional branches to create a loop, as shown on the previous page or in any part of a program where you wish to transfer execution to another program line.

Problem: When modified, the following program calculates the square of consecutive whole numbers beginning with 2 each time it is run. Key in the program with the PRGM-RUN switch PRGM RUN set to PRGM. Then switch to RUN and run the program a few times to see how it works. Finally, modify the program by adding a GTO 03 instruction after the f PAUSE instruction found in line 08. Do this by re-keying in the program and adding the GTO 03 instruction as indicated above. This should create a loop that will continually display a new number and its square, then increment the number by one and compute *its* square and so on. To load the original program, before modification, slide the PRGM-RUN switch PRGM RUN to PRGM and perform the following keystrokes:

Keystrokes	Display
f CLEAR PRGM	00
1	01– 1
STO 1	02– 23 1
1	03– 1
STO + 1	04– 23 51 1
RCL 1	05– 24 1
f PAUSE	06– 14 74
g x^2	07– 15 0
f PAUSE	08– 14 74
GTO 00	09– 13 00

Run the program to generate a table of squares.

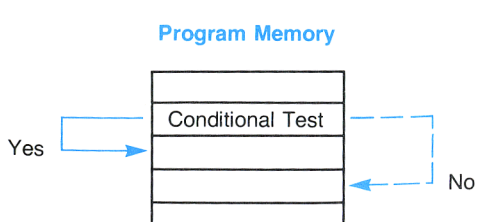
Set the PRGM-RUN switch PRGM RUN to RUN and run the program to generate a table of squares. Remember to return to top of program memory by pressing g RTN.

Conditional Tests and Conditional Branches

Often there are times when you want to make a decision. Your HP-33E makes use of conditionals and conditional branches to increase your programming capabilities. The *conditional* operations on your HP-33E keyboard are useful as program instructions to allow your calculator to make decisions. The eight conditionals that are available on your HP-33E are:

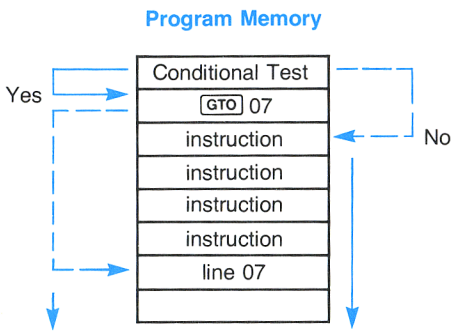
- f x≠y — Tests to see if the value in the X-register is not equal to the value in the Y-register.
- f x=y — Tests to see if the value in the X-register is equal to the value in the Y-register.
- f x>y — Tests to see if the value in the X-register is greater than the value in the Y-register.
- f x≤y — Tests to see if the value in the X-register is less than or equal to the value in the Y-register.
- g x≠0 — Tests to see if the value in the X-register is not equal to zero.
- g x=0 — Tests to see if the value in the X-register is equal to zero.
- g x>0 — Tests to see if the value in the X-register is greater than zero.
- g x<0 — Tests to see if the value in the X-register is less than zero.

Each conditional essentially asks a question when it is encountered as an instruction in a program. If the answer is YES, program execution continues sequentially downward with the next line in program memory. If the answer is NO, the calculator branches *around* the next line. For example:



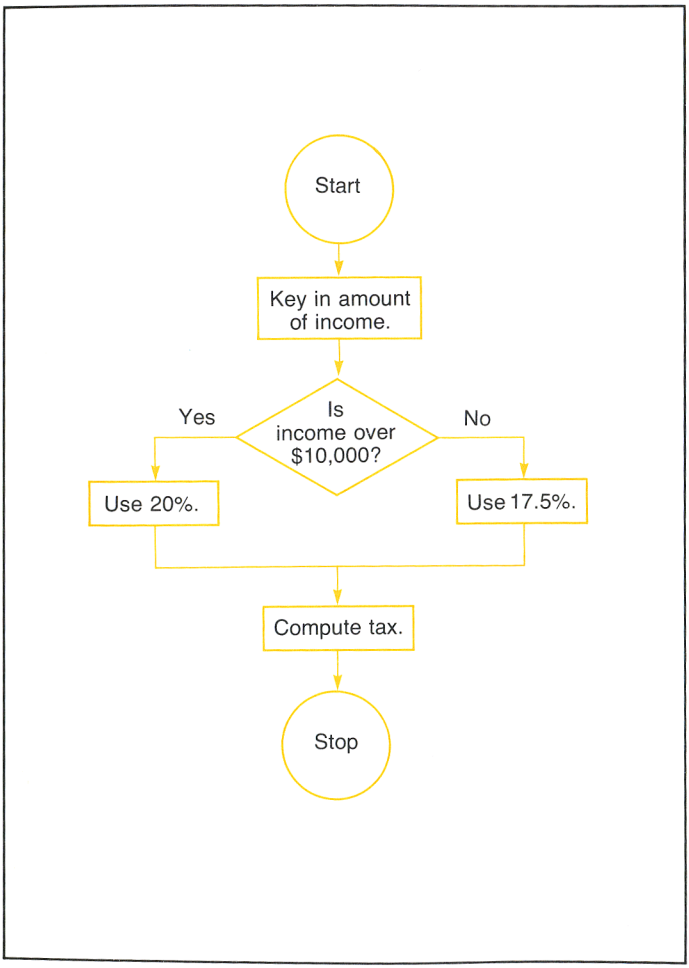
You can see that after it has made the conditional test, the calculator will do the next instruction if the test is *true*. This is the "DO IF TRUE" rule.

The line immediately following the conditional test can contain any instruction. The most commonly used instruction, you'll find, will be a [GTO] instruction. This will branch program execution to another section of program memory if the conditional test is true.



Example: Certified Public Accountant Howard Preparer knows that persons with incomes over $10,000 pay a tax of 20% and persons with incomes of $10,000 or less pay a tax of 17.5%. To make his job easier, Preparer wants to write a program which will allow him to compute tax rates for all his clients in the simplest way possible. He will use a program containing conditional branches.

The flowchart for the program might look like this:



To key in the program, slide the PRGM-RUN switch PRGM RUN to PRGM and clear program memory.

Keystrokes	Display	
f CLEAR PRGM	00	
EEX	01- 33	$10,000 placed in Y-register.
4	02- 4	
x≷y	03- 21	
f x>y	04- 14 51	If income greater than $10,000 go to line 11.
GTO 11	05- 13 11	
1	06- 1	Tax percentage for this portion of program—17.5.
7	07- 7	
.	08- 73	
5	09- 5	
GTO 13	10- 13 13	
2	11- 2	Tax percentage for this portion of program—20.
0	12- 0	
g %	13- 15 74	
GTO 00	14- 13 00	

To run the program to compute taxes on incomes of $15,000 and $7,500:

Slide the PRGM-RUN switch PRGM RUN to RUN and press g RTN to return to the top of program memory.

Keystrokes	Display
15000 R/S	3,000.0000
7500 R/S	1,312.5000

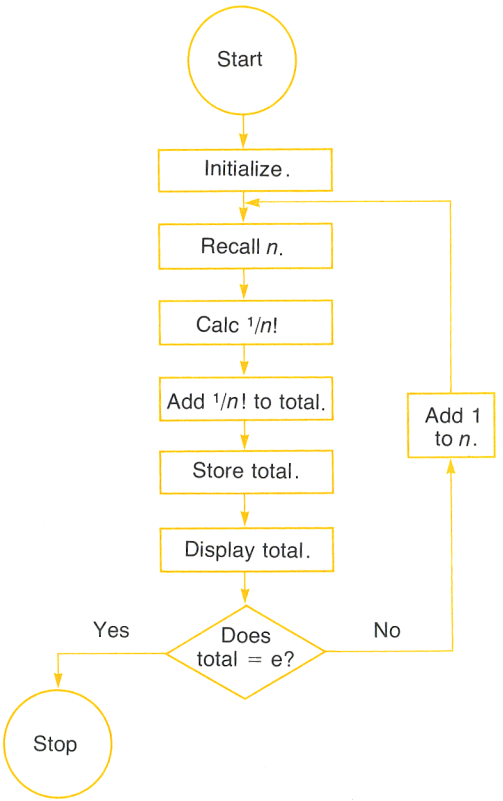
All Preparer has to do to compute tax rates for his other clients is key in their incomes and press R/S. The calculator automatically determines the clients' tax bracket and computes the tax.

Another place where you often want a program to make a decision is within a loop. The loops that you have seen to this point have been *infinite* loops—that is, once the calculator begins executing a loop, it remains locked in that loop, executing the same set of instructions over and over again, until you halt the running program by pressing [R/S] or until the calculator overflows.

You can use the decision-making power of the conditional instructions to shift program execution out of a loop. A conditional instruction can shift execution out of a loop after you have executed the loop a specified number of times or when a certain value has been reached within the loop.

Example: Your HP-33E contains a value for e, the base of the natural logarithms. (You can display the calculator's value for e by pressing 1 [g] [e^x].) The following program uses the series $e = 1 + 1/1! + 1/2! + \ldots + 1/n!$ to approximate the value for e. After each execution through the loop, the latest approximation is compared to the *calculator's* value for e. When the two values are equal, the execution is transferred out of the loop to stop the program.

The flowchart shown on the next page may help in understanding how the program works.


Start
Initialize.
Recall n.
Calc 1/n!
Add 1/n! to total.
Store total.
Display total.
Does total = e?
Yes
No
Stop
Add 1 to n.

To load the program into the calculator:

Slide the PRGM-RUN switch PRGM RUN to PRGM.

Keystrokes	Display	
f CLEAR PRGM	00	
f FIX 9	01-14 11 9	To view entire number.
1	02- 1	
STO 0	03- 23 0	R_0 will hold n.
STO 1	04- 23 1	R_1 will hold $n!$
STO 2	05- 23 2	R_2 will hold the sum of the series.
RCL 0	06- 24 0	
STO × 1	07-23 61 1	$n!$
RCL 1	08- 24 1	
g 1/x	09- 15 3	$1/n!$
RCL 2	10- 24 2	
+	11- 51	Adds $1/n!$ to previous total.
STO 2	12- 23 2	
f PAUSE	13- 14 74	Pause to see value up to that point.
1	14- 1	
g e^x	15- 15 1	Calculate e.
f x=y	16- 14 71	Is total equal to e?
GTO 00	17- 13 00	If it is, go to line 00.
1	18- 1	If it is not, add 1 to n.
STO + 0	19-23 51 0	
GTO 06	20- 13 06	Loop back to line 06 and find new approximation.
GTO 00	21- 13 00	If total equals e, display e and halt.

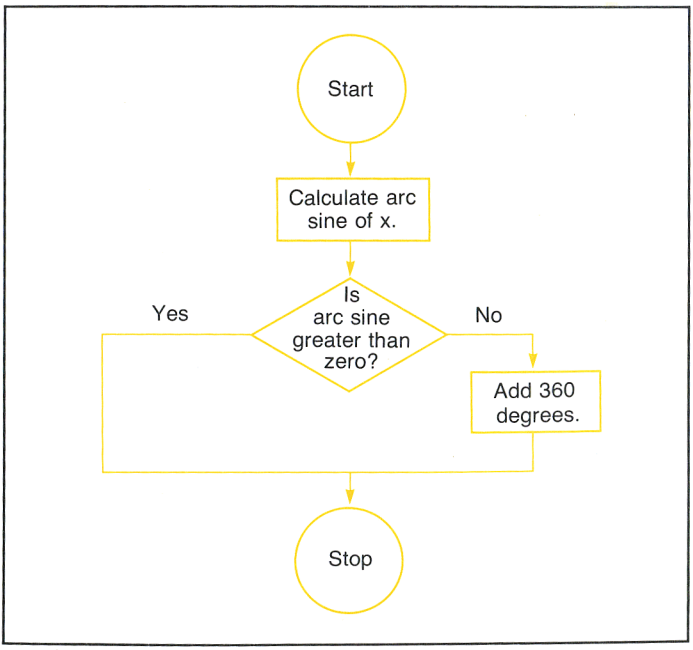
To intialize the program, first slide the PRGM-RUN switch PRGM  RUN to RUN. Press g RTN to go to top of program memory. Then press R/S to run the program.

Keystrokes	Display	
g RTN	0.0000	Assumes stack is cleared to zeros.
R/S	2.718281828	

You can see that execution continues within the loop until the approximation for e equals the calculator's value for e. When the instruction [x=y] in line 16 is finally true, execution is transferred out of the loop by the subsequent [GTO] 00 instruction and halted.

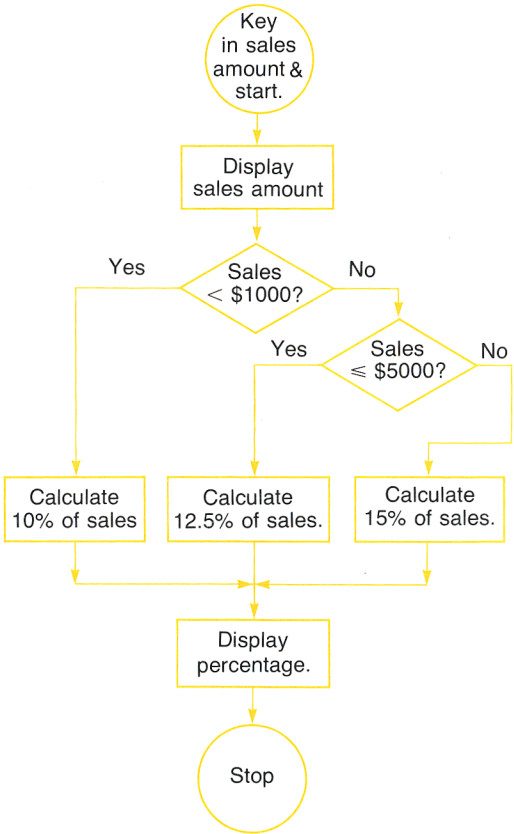
Problems

1. Write a program that will calculate the arc sine (that is, $\sin^{-1}$) of a value that has been keyed into the displayed X-register. Test the resulting angle with a conditional, and if it is negative or zero, add 360 degrees to it to make the angle positive. Use the flowchart below to help you write the program:



2. Use this flowchart to help you write a program that will allow a salesman to compute his commissions at the rates of 10% of sales of up to $1,000, 12.5% for sales of $1,000 to $5,000, and 15% for sales of over $5,000. The program should display the amount of sales and the amount of commission.

Load the program and run it for sales amounts of $500, $1,000, $1,500, $5,000, and $6,000. (Answers: $50.00, $125.00, $187.50, $625.00, $900.00)


Key in sales amount & start.
Display sales amount
Sales < $1000?
Yes
No
Sales ≤ $5000?
Yes
No
Calculate 10% of sales
Calculate 12.5% of sales.
Calculate 15% of sales.
Display percentage.
Stop

Answers

1.

Keystrokes	Display	Keystrokes	Display
f CLEAR PRGM	00		
g SIN⁻¹	01– 15 7	6	05– 6
g x>0	02– 15 51	0	06– 0
R/S	03– 74	+	07– 51
3	04– 3	GTO 00	08– 13 00

User instructions: After keying in program, set PRGM-RUN switch PRGM RUN to RUN and press g RTN to return to top of program memory. Input number and press R/S.

2.

Keystrokes	Display	Keystrokes	Display
f CLEAR PRGM	00	×	13– 61
EEX	01– 33	GTO 00	14– 13 00
3	02– 3	.	15– 73
f x>y	03– 14 51	1	16– 1
GTO 21	04– 13 21	2	17– 2
5	05– 5	5	18– 5
×	06– 61	×	19– 61
x≷y	07– 21	GTO 00	20– 13 00
f x≤y	08– 14 41	x≷y	21– 21
GTO 15	09– 13 15	.	22– 73
.	10– 73	1	23– 1
1	11– 1	×	24– 61
5	12– 5	GTO 00	25– 13 00

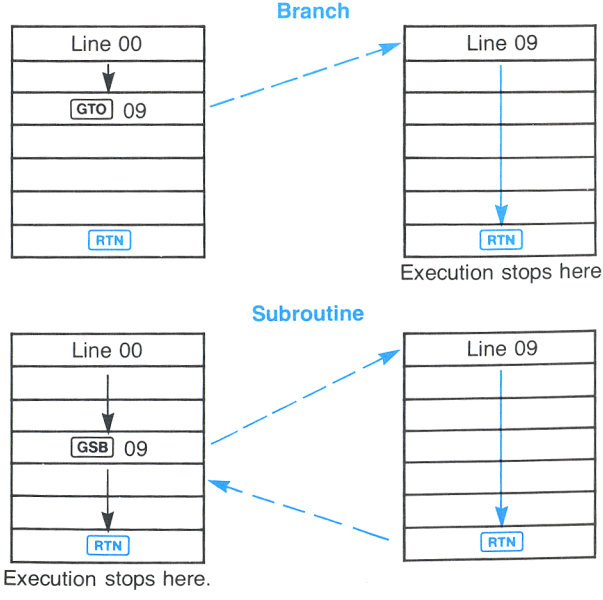
User instructions: After keying in program, set PRGM-RUN switch PRGM RUN to RUN and press g RTN to return to top of program memory. Input sales dollars and press GSB 01.

Section 5

Subroutines

Often, a program contains a certain series of instructions that are executed several times throughout the program. When the same set of instructions occurs more than once in a program, it can be executed as a subroutine. A subroutine is selected by the GSB *(go to subroutine)* operation, followed by a line address (01 through 49).

A GSB instruction transfers execution to the routine specified by the line address, just like a GTO instruction. However, after a GSB instruction has been executed and the running program executes a RTN *(return)*, execution is then transferred back to the next instruction after the GSB. Execution then continues sequentially downward through program memory. The illustration below should make the distinction between GTO and GSB more clear.



In the branching illustration on the left when you pressed [R/S] from the keyboard, the program executes instructions sequentially downward through program memory. When it encountered the [GTO] 09 instruction, it went to line 09 and continued execution from there until it encountered a [RTN]. When it executed the [RTN] instruction, execution stopped.

As with the [GTO] instruction, when a running program encounters a [GSB] 09 (go to subroutine 09) instruction, as shown on the right, it goes to line 09 and resumes execution. However, when it encounters a [RTN] *(return)*, program execution is transferred *back* to the next instruction after the [GSB] 09, and execution resumes.

Example: A quadratic equation is of the form $ax^2 + bx + c = 0$. Its two roots may be found by the formulas $r_1 = \dfrac{-b + \sqrt{b^2 - 4ac}}{2a}$ and $r_2 = \dfrac{-b - \sqrt{b^2 - 4ac}}{2a}$. Notice the similarity between the solutions for r_1 and r_2.

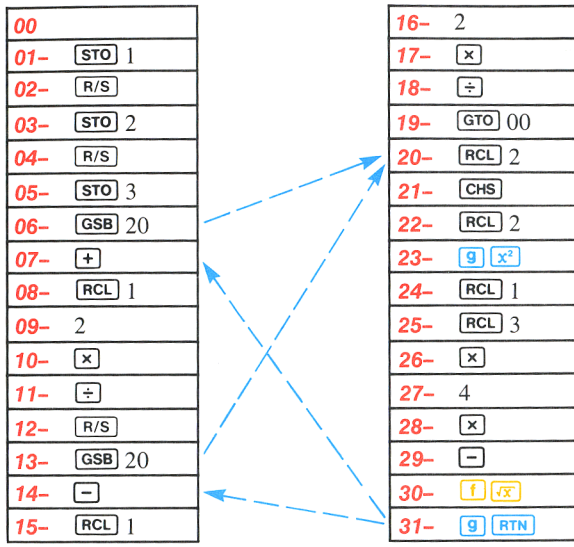
The program below permits you to enter the value of a, b, and c as follows:

1. Input a, press [R/S].
2. Input b, press [R/S].
3. Input c.
4. Press [R/S] to calculate r_1.
5. Press [R/S] again to calculate r_2.

00	f CLEAR PRGM	
01–	STO 1	Stores a in R_1.
02–	R/S	
03–	STO 2	Stores b in R_2.
04–	R/S	
05–	STO 3	Stores c in R_3.
06–	RCL 2	Calculate r_1.
07–	CHS	
08–	RCL 2	
09–	g x^2	
10–	RCL 1	These sections of program memory are identical.
11–	RCL 3	
12–	×	
13–	4	
14–	×	
15–	−	
16–	f $\sqrt{x}$	
17–	+	
18–	RCL 1	
19–	2	
20–	×	
21–	÷	
22–	R/S	
23–	RCL 2	Calculate r_2.
24–	CHS	
25–	RCL 2	
26–	g x^2	
27–	RCL 1	
28–	RCL 3	
29–	×	
30–	4	
31–	×	
32–	−	
33–	f $\sqrt{x}$	
34–	−	
35–	RCL 1	
36–	2	
37–	×	
38–	÷	
39–	GTO 00	

Since the routine for calculating r_1 contains a large section of program memory that is identical to a large section in the routine for calculating r_2, you can simply create a *subroutine* that will execute this section of instructions. The subroutine is then called up and executed in both the solution for r_1 and the solution for r_2.

This illustrates how the subroutine is used in the program.



With the modified program, when you press [R/S] to begin calculation of r_1, execution begins with the instruction in line 05. The value of c is stored in register R_3. At the next line, [GSB] 20 transfers execution to line 20 and computes the quantities $-b$ and $\sqrt{b^2 - 4ac}$, placing them in the X- and Y-registers of the stack, ready for addition or subtraction. When the [RTN] instruction in line 31 is encountered, execution transfers back to the main routine and continues with the [+] instruction in line 07. Thus the root of r_1, is computed and displayed, and the routine stops with the [R/S] in line 12.

When you press [R/S] again to calculate r_2, execution begins with line 13, transfers out to execute the subroutine beginning at line 20, and returns to line 14. This time $\sqrt{b^2 - 4ac}$ is subtracted from $-b$, and root r_2 is computed. By using a subroutine eight steps of program memory are saved!

To key in the program and subroutine:

Slide the PRGM-RUN switch PRGM RUN to PRGM.

Keystrokes	Display	
f CLEAR PRGM	00	
STO 1	01– 23 1	Stores a in R_1.
R/S	02– 74	
STO 2	03– 23 2	Stores b in R_2.
R/S	04– 74	
STO 3	05– 23 3	Stores c in R_3.
GSB 20	06– 12 20	
+	07– 51	
RCL 1	08– 24 1	
2	09– 2	
×	10– 61	
÷	11– 71	
R/S	12– 74	
GSB 20	13– 12 20	Calculates $r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$
−	14– 41	
RCL 1	15– 24 1	
2	16– 2	
×	17– 61	
÷	18– 71	
R/S	19– 74	
RCL 2	20– 24 2	Subroutine places $-b$ in X-register and $\sqrt{b^2 - 4ac}$ in X-register, ready for addition or subtraction.
CHS	21– 32	
RCL 2	22– 24 2	
g x^2	23– 15 0	
RCL 1	24– 24 1	
RCL 3	25– 24 3	
×	26– 61	
4	27– 4	
×	28– 61	
−	29– 41	
f $\sqrt{x}$	30– 14 0	
g RTN	31– 15 12	

To initialize the program you key in a and press R/S, key in b and press R/S, and key in c. Then, to find root r_1 press R/S. To find r_2 press R/S again.

Run the program now to find the roots of the equations:

$x^2 + x - 6 = 0; 3x^2 + 2x - 1 = 0.$

To run the program:

Slide the PRGM-RUN switch PRGM RUN to RUN and press [g] [RTN] to return to top of program memory.

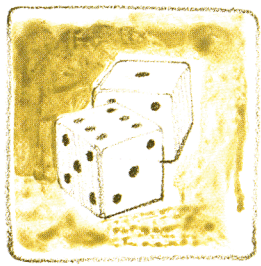
Keystrokes	Display	
1 [R/S]	1.0000	
1 [R/S]	1.0000	Program initialized.
6 [CHS]	-6.	
[R/S]	2.0000	Calculates the first root, r_1.
[R/S]	-3.0000	Calculates the second root, r_2.
[g] [RTN]	-3.0000	Returns to top of program memory
3 [R/S]	3.0000	
2 [R/S]	2.0000	Program initialized.
1 [CHS]	-1.	
[R/S]	0.3333	Calculates r_1.
[R/S]	-1.0000	Calculates r_2.

If the quantity $b^2 = 4ac$ is a negative number, the calculator will display **Error 0** and the running program will stop, because the square root of a negative number results in an error.

Routine-Subroutine Usage

Subroutines give you extreme versatility in programming. A subroutine can contain a loop, or it can be executed as part of a loop. Another

common and space-saving trick is to use the same routine both as a subroutine and as part of the main program.



Example: The program below simulates the throwing of a pair of dice, pausing to display first the value of one die (an integer from one to six) and then that of the second die (another integer from one to six). The "heart" of the program is a random number generator (actually a pseudorandom number generator) that is executed first as a subroutine and then as part of the main program. When you key in a starting number (called a "seed") and press [GSB] 01, the digit for the first die is generated using the routine as a subroutine. Then the digit for the second die is generated using the same routine as part of the main program.

To key in the program, slide the PRGM-RUN switch PRGM RUN to PRGM and press the following keys:

Keystrokes	Display	
[f] CLEAR [PRGM]	00	
[STO] 0	01– 23 0	
[GSB] 04	02– 12 04	
[f] [PAUSE]	03– 14 74	Routine following line 03 executed first as a subroutine.
[RCL] 0	04– 24 0	
9	05– 9	
9	06– 9	

Keystrokes	Display	
7	07– 7	Then executed as a routine.
×	08– 61	
g FRAC	09– 15 33	
STO 0	10– 23 0	
6	11– 6	
×	12– 61	
1	13– 1	
+	14– 51	
g INT	15– 15 32	
f FIX 0	16–14 11 0	
g RTN	17– 15 12	
+	18– 51	

Now slide the PRGM-RUN switch PRGM RUN to RUN and "roll" the dice using your HP-33E. To roll the dice, key in a decimal "seed" (that is, $0 < n < 1$). Then press GSB 01. The calculator will display the number rolled by the first die. Press R/S and the number rolled by the second die will be added to the first. To make another roll, press GSB 02 and then R/S.

You can play a game with your friends using the "dice." If your first "roll" is 7 or 11, you win. If it is another number, that number becomes your "point." You then keep "rolling" (pressing GSB 02 then R/S) until the dice again total your point (your win) or you roll a 7 or an 11 (you lose).

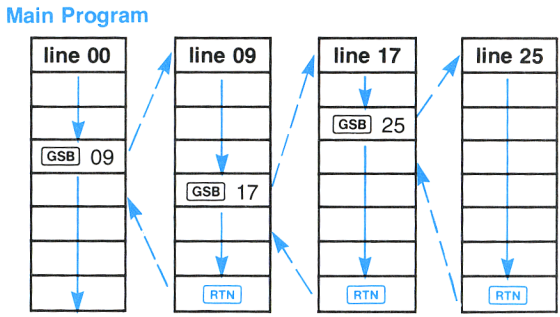
To run the program, slide the PRGM-RUN switch PRGM RUN to RUN. Press g RTN to return to top of program memory.

Keystrokes	Display	
.2315478 GSB 01	4.	The 6 that flashed earlier plus the 4 in the display determine your point—10.
R/S	10.	Total point.
GSB 02	5.	
R/S	8.	Your total is 8. You missed your point.
GSB 02	1.	
R/S	5.	Your total is 5. You missed it again.
GSB 02	3.	
R/S	7.	Your total is 7. You lose.

Subroutine Limits

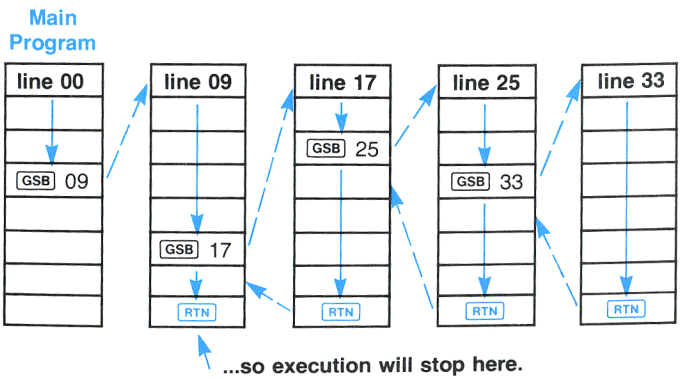
A subroutine can call up another subroutine, and that subroutine can call up yet another. Subroutine branching is limited only by the number of *returns* that can be held pending by the HP-33E. Three subroutine returns can be held pending at any one time in the HP-33E. The diagram below should make this more clear.

Three returns can be pending.



The calculator can return back to the main program from subroutines that are three deep, as shown. However, if you attempt to call up subroutines that are four deep, the calculator will execute only three returns:

Only three returns can be pending...



If program control goes to line 00, either as a result of a GTO or by incrementing from line 49, the calculator stops execution unless in a subroutine. In this case the calculator executes a RTN and continues execution at the line number after the GSB. Naturally, the calculator can execute the RTN instruction as a *stop* any number of times.

If you are executing a program one step at a time with the SST key and encounter a GSB instruction, the calculator will execute the entire subroutine before continuing to the next step. However, only one RTN instruction may be executed as the result of a GSB instruction during single-step execution; so if a program contains a subroutine within a subroutine, execution will not return to the main program during SST execution.

Section 6

Program Editing

Even the most experienced programmer finds errors in his programs. These errors range from mistakes in the original equations to mistakes in recording the program. Wherever they occur they need to be found and corrected, and the HP-33E is designed to make this error-checking process as easy as possible.

Finding the Error

One of the easiest ways to find out if your program is working properly is to work a test case in which you either know the answer or the answer can be easily determined. For example, if you have a program that calculates the area of a circle using the formula $Area = \pi \times r^2$, you can easily determine that an input value of 1 for r will give an answer of π.

[SST] **Execution.** In longer programs a wrong test-case answer will seldom pinpoint the mistake. For these cases, you can slow down the program execution by using the [SST] key in RUN mode. In RUN mode, the [SST] key will execute your program instructions one at a time. When you hold the [SST] key down in RUN mode, the program line number and keycode are displayed. When you release the [SST] key, the instruction is executed. Use [SST] on the simple area of a circle program shown below to familiarize yourself with its operation.

Example: This program calculates the area of a circle using the formula: $A = \pi r^2$ where r is the radius. Set the PRGM-RUN switch to PRGM PRGM RUN and press [f] CLEAR [PRGM] to clear program memory and display line 00. Then key in the list of keys shown below.

Keystrokes	Display
[f] CLEAR [PRGM]	00
[g] [x^2]	01- 15 0
[g] [π]	02- 15 73
[×]	03- 61

The program assumes that a value for r has been keyed into the X-register. To run the program, set the PRGM-RUN switch PRGM RUN back to RUN and press [g] [RTN] to return to the top of program memory. Also press [f] [FIX] 4 to return to normal FIX4 display. Now step through the program in slow motion using a value of 10 for r.

Keystrokes	Display	
10	*10.*	
[SST]	*01– 15 0*	When you hold [SST] down, the first instruction is displayed.
	100.0000	When you release [SST], the first instruction is executed.
[SST]	*02– 15 73*	Again holding [SST] down displays the second instruction.
	3.1416	Again releasing [SST] executes the second instruction.
[SST]	*03– 61*	Holding [SST] down displays the third instruction this time.
	314.1593	And releasing [SST] executes the third instruction.

You can see that it would be easy to spot a mistake in your program using [SST] key.

When you hold the [BST] key down in RUN mode, the program line number and keycode for the previous line are displayed. When you release [BST], the X-register is again displayed. However if you switch back to PRGM mode, you will find that the previous line is now displayed. And if you press [R/S] in RUN mode after pressing [BST], the calculator will begin execution from the previous line in program memory. Now press [BST] in RUN mode to review the program instructions of the above program.

Keystrokes	Display	
g BST	*03– 61*	Holding BST down in RUN mode displays the previous instruction.
	314.1593	Releasing the BST key displays the original contents of the X-register.
g BST	*02– 15 73*	Again holding BST down displays the previous line in program memory.
	314.1593	And releasing BST displays the original contents of the X-register again.

If you now switch to PRGM mode the second line will be displayed.

02– 15 73

Cued Stops. If you have a program that is halted several times during execution for data entries, you may want to "identify" each stop by recording a familiar number into the program just before each R/S instruction. Then when the calculator stops execution because of the R/S instruction in the program, you can look at the displayed X-register to see the "identification number" for the required input. For example, if your program contains eight stops for data inputs, it may be helpful to have the numbers 1 through 8 appear so you know which input is required each time. These identification numbers are helpful in editing a program.

If you key in data after the program has stopped running, remember that resuming program execution terminates digit entry.

Changing One Instruction

Changing or correcting one line of your program is easy with your calculator because of the features built into it. Once the error has been found, use SST, g BST, or GTO . nn in PRGM mode or GTO in RUN mode to display the line preceding the line to be changed. For example,

to change the instruction in line 06, you need to display line 05. If you wish to change the line, simply press the correct key or keys for line 06. They will write over and replace the incorrect information already stored in that line.

If line 06 is an extra line in your program, press [g] [NOP] *(no operation)*. This instruction tells the calculator not to perform any operation here.

Example: The program represented below is designed to take the cube root of a number.

Keystrokes	Display
[f] CLEAR [PRGM]	*00*
[ENTER↑]	*01– 31*
3	*02– 3*
[g] [1/x]	*03– 15 3*
[f] [y^x]	*04– 14 3*

Suppose that upon reviewing the program with the [SST] key, however, you discover you have keyed in the following mistake-ridden program:

Keystrokes	Display	
[f] CLEAR [PRGM]	*00*	
[ENTER↑]	*01– 31*	
3	*02– 3*	
[g] [%]	*03– 15 74*	Oops! You pressed the wrong key.
[x≷y]	*04– 21*	You made another mistake.
[f] [y^x]	*05– 14 3*	

Set the PRGM-RUN switch PRGM RUN to PRGM, press [f] CLEAR [PRGM], and key in this mistake-ridden second program now.

To correct the program, press [g] [BST] three times to display line 02. Then correct the first mistake by keying in the correct keys for line 03.

Keystrokes	Display	
	02– 3	First display this line.
[g] [1/x]	*03– 15 3*	Then press the correct keys for line 03.

With line 03 displayed you are ready now to correct line 04. Since this is an unwanted extra line, use the [g] [NOP] function to replace its contents.

Keystrokes	Display	
	03– 15 3	Display line 03 to correct line 04.
[g] [NOP]	04– 15 13	Press [g] [NOP] so that the calculator will not perform an operation here.

Now set the PRGM-RUN switch PRGM RUN back to RUN and press [g] [RTN] to reset the calculator to line 00. The example below will help you determine whether or not you have corrected the program.

Example: Find the cube root of 8 and then of 125.

Keystrokes	Display
8 [R/S]	2.
125 [R/S]	5.

Adding Instructions

If you have recorded a medium-sized program and have left out a crucial sequence of keystrokes right in the middle, you do not have to start over. The missing sequence of keystrokes can be recorded in the available lines following your program. You can then use the [GSB] key to make a subroutine to the sequence when it is needed and then make it return to the main part of your program at the end of the sequence.

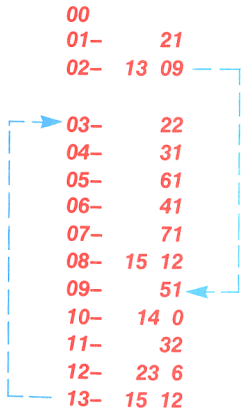
The sample inoperative program shown below should make this more clear. Three lines are missing between lines 02 and 03.

Keycodes	Keystrokes	
00		
01– 21	[x≷y]	
02– 51	[+]	
		Missing three lines ([f] [√x], [CHS], and [STO] 6) here.

Keycodes	Keystrokes
03– 22	R↓
04– 31	ENTER↑
05– 61	×
06– 41	−
07– 71	÷
08– 15 12	9 RTN

In order to add the missing lines we need to branch to one of the unused lines in program memory.

To do this line 02 must be changed to a GSB 09 operation enabling you to form a subroutine. Line 09 must contain the instruction formerly found in line 02. The missing keystrokes are then stored in lines 10 through 12 and the 9 RTN instruction stored in line 13 returns execution back to line 03 in the main program. The corrected program is shown below:



Goes to subroutine beginning at line 09.

Former line 02.

Returns to line 03 and continues execution.

Appendix A

Service and Maintenance

Your Hewlett-Packard Calculator

Your calculator is another example of the award-winning design, superior quality, and attention to detail in engineering and construction that have marked Hewlett-Packard electronic instruments for more than 30 years. Each Hewlett-Packard calculator is precision crafted by people who are dedicated to giving you the best possible product at any price.

After construction, every calculator is thoroughly inspected for electrical or mechanical flaws.

When you purchase a Hewlett-Packard calculator, you deal with a company that stands behind its products.

AC Line Operation

Your calculator contains a rechargeable battery pack consisting of nickel-cadmium batteries. When you receive your calculator, the battery pack inside may be discharged, but you can operate the calculator immediately by using the ac adapter/recharger.

> **Note:** Do not attempt to operate the calculator from the ac line with the battery pack removed.

The procedure for using the ac adapter/recharger is as follows:

1. You need *not* turn the calculator off.
2. Insert the ac adapter/recharger plug into the rear connector of the calculator.
3. Insert the power plug into a live ac power outlet.

> **Note:** It is normal for the ac adapter/recharger to be warm to the touch when it is plugged into an ac outlet.

> **CAUTION**
>
> The use of a charger other than the HP recharger supplied with the calculator may result in damage to your calculator.

Battery Charging

The rechargeable batteries in the battery pack are charged while you operate the calculator from the ac adapter/recharger. Batteries will charge with the calculator on or off, provided batteries are in place and recharger is connected. Normal charging times between the fully discharged state and the fully charged state are (depending on ac line voltage):

Calculator off: 5 to 9 hours
Calculator on: 17 hours

Shorter charging periods will reduce the operating time you can expect from a single battery charge. Whether the calculator is off or on, the calculator battery pack is never in danger of becoming overcharged.

Note: The ac adapter/recharger is a sealed unit and is not repairable. Return it to Hewlett-Packard if service is required.

Battery Operation

To operate the calculator from battery power alone, simply disconnect the recharger plug from the rear of the calculator. (Even when not connected to the calculator, the ac adapter/recharger may be left plugged into the ac outlet.)

Using the calculator on battery power gives the calculator full portability, allowing you to carry it nearly anywhere. A fully charged battery pack typically provides 3 hours of continuous operation. By turning the power off when the calculator is not in use, the charge on the battery pack should easily last throughout a normal working day.

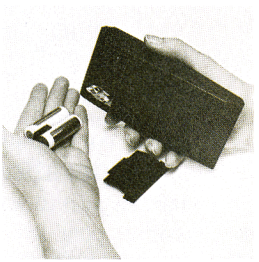
Battery Pack Replacement

To replace the battery pack use the following procedure:

1. Set calculator ON-OFF switch to OFF and disconnect the battery ac adapter/recharger from the calculator.

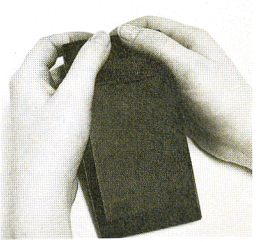
2. Press down on the short ridges of the battery door, close to the edge, until the door release snaps open. Slide the door open.

3. When door is removed, turn calculator over and gently shake, allowing the battery pack to fall into the palm of your hand.



4. Place the new battery pack into the calculator. Your calculator will only turn on if the battery pack is inserted correctly.

5. Insert battery door and slide door back into place.



6. Turn calculator over and turn power on to assure proper battery installation. If the display does not light, make sure the battery pack is correctly placed in the calculator.

Battery Care

When not being used, the batteries in your calculator have a self-discharge rate of approximately 1 percent of available charge per day. After 30 days, a battery pack might have only 50 to 75 percent of its charge remaining, and the calculator might not even turn on. If a calculator fails to turn on, you should substitute a charged battery pack, if available, for the one in the calculator or plug into the ac adapter/recharger. The discharged battery pack should be charged for at least 12 hours.

If a battery pack will not hold a charge and seems to discharge very quickly in use, it may be defective. If the one-year warranty on the battery pack has not expired, return the defective pack to Hewlett-Packard according to the shipping instructions. (If you are in doubt about the cause of the problem, return the complete calculator along with its battery pack and ac adapter/recharger.) If the battery pack is out of warranty, see your nearest dealer to order a replacement.

WARNING

Do not attempt to incinerate or mutilate the battery pack—the pack may burst or release toxic materials.

Do not connect together or otherwise short-circuit the battery terminals—the pack may melt or cause serious burns.

Service

Low Power

When you are operating from battery power and the batteries get low, a raised decimal is turned on at the far left of the display to warn you that you have between 1 minute and 25 minutes of operating time left.

•1.23 49

If the display contains the low power indication, a negative number looks like half a divide sign.

±1.23 49

To return to full power either connect the ac adapter/recharger to the calculator as described under AC Line Operation, or substitute a fully charged battery pack for the one in the calculator.

Blank Display

If the display blanks out, turn the calculator off, then on. If a display of numbers does not appear in the display, check the following:

1. If the ac adapter/recharger is attached to the calculator, make sure it is plugged into an ac outlet.
2. Examine the battery pack to see if the contacts are dirty.
3. Substitute a fully charged battery pack, if available, for the one that was in the calculator.
4. If the display is still blank, try operating the calculator using the ac adapter/recharger (with the batteries in the calculator).
5. If, after step 4, the display is still blank, service is required. (Refer to Warranty.)

Temperature Range

Temperature ranges for the calculator are:

Operating	0° to 45°C	32° to 113°F
Charging	15° to 40°C	59° to 104°F
Storage	−40° to 55°C	−40° to 131°F

Limited One-Year Warranty

What We Will Do

The HP-33E and its accessories are warranted by Hewlett-Packard against defects in materials and workmanship for one year from date of original purchase. If you sell your calculator or give it as a gift, the warranty is automatically transferred to the new owner and remains in effect for the original one-year period. During the warranty period we will repair or, at our option, replace at no charge a product that proves to be defective provided that you return the product, shipping prepaid, to a Hewlett-Packard repair center.

How to Obtain Repair Service

Hewlett-Packard maintains repair centers in most major countries throughout the world. You may have your calculator repaired at a Hewlett-Packard repair center anytime it needs service, whether the unit is under warranty or not. There is a charge for repairs after the one-year warranty period. Please refer to the Shipping Instructions in this handbook.

The Hewlett-Packard United States Repair Center for handheld and portable printing calculators is located at Corvallis, Oregon. The mailing address is:

HEWLETT-PACKARD COMPANY
CORVALLIS DIVISION SERVICE DEPT.
P.O. BOX 999
CORVALLIS, OREGON 97330

What Is Not Covered

This warranty does not apply if the product has been damaged by accident or misuse, or as a result of service or modification by other than an authorized Hewlett-Packard repair center.

No other express warranty is given. The repair or replacement of a product is your exclusive remedy. **ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS IS LIMITED TO THE ONE-YEAR DURATION OF THIS WRITTEN WARRANTY.** Some states do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you. **IN NO EVENT SHALL HEWLETT-PACKARD COMPANY BE LIABLE FOR CONSEQUENTIAL DAMAGES.** Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Obligation to Make Changes

Products are sold on the basis of specifications applicable at the time of sale. Hewlett-Packard shall have no obligation to modify or update products once sold.

Warranty Information Toll-Free Number

If you have any questions concerning this warranty please call **800/648-4711.** (In Nevada call 800/992-5710.)

Repair Policy

Hewlett-Packard calculators are normally repaired and reshipped within five (5) working days of receipt at any repair center. This is an average time and could possibly vary depending upon time of year and work load at the repair center.

Shipping Instructions

The calculator should be returned, along with completed Service Card, in its shipping case (or other protective package) to avoid in-transit damage. Such damage is not covered by warranty and Hewlett-Packard suggests that the customer insure shipments to the repair center. A calculator returned for repair should include the ac adapter/recharger and the battery pack. Send these items to the address shown on the Service Card. *Remember to include a sales slip or other proof of purchase with your unit.*

Whether the unit is under warranty or not, it is your responsibility to pay shipping charges for delivery to the Hewlett-Packard repair center.

After warranty repairs are completed, the repair center returns the unit with postage prepaid. On out-of-warranty repairs, the unit is returned C.O.D. (covering shipping costs and the service charge).

Programming and Applications Assistance

Should you need technical assistance concerning programming, calculator applications, handbook corrections, etc., call Hewlett-Packard Customer Support at 503/757-2000. This is not a toll-free number, and we regret that we cannot accept collect calls. As an alternative, you may write to:

Hewlett-Packard
Corvallis Division Customer Support
1000 N.E. Circle Boulevard
Corvallis, OR 97330

A great number of our users submit program applications or unique program key sequences to share with other HP owners. Hewlett-Packard will only consider using ideas given freely to us. Since it is the policy of Hewlett-Packard not to accept suggestions given in confidence, the following statement must be included with your submittal:

"All information stated within is submitted to Hewlett-Packard Company without any confidentiality or obligation. I understand that by forwarding this information, no expressed, implied, or confidential relationship is established with Hewlett-Packard. Hewlett-Packard may copyright, distribute, publish, reproduce, dispose of, or use any or all of the information in any way without compensation to me."

Further Information

Service contracts are not available. Calculator circuitry and design are proprietary to Hewlett-Packard, and service manuals are not available to customers.

Should other problems or questions arise regarding repairs, please call your nearest Hewlett-Packard sales office or repair center.

Note: Not all Hewlett-Packard repair centers offer service for all models of HP calculators. However, you can be sure that service may be obtained in the country where you bought your calculator.

If you happen to be outside of the country where you bought your calculator, you can contact the local Hewlett-Packard repair center to see if service capability is available for your model. If service is unavailable, please ship your calculator to the following address:

Hewlett-Packard
1000 N.E. Circle Boulevard
Corvallis, Oregon 97330
U.S.A.

All shipping and reimportation arrangements are your responsibility.

Appendix B

Error Indications

If you attempt a calculation containing an improper operation—say, division by zero—the display will show Error and a number. To clear, press any number.

The following operations will display Error plus a number:

Error 0

[÷], where $x = 0$.
[y^x], where $y = 0$ & $x \leq 0$.
[y^x], where $y < 0$ & x is non-integer.
[$\sqrt{x}$] where $x < 0$.
[$1/x$], where $x = 0$.
[LOG], where $x \leq 0$.
[LN], where $x \leq 0$.
[SIN^{-1}], where $|x|$ is > 1.
[COS^{-1}], where $|x|$ is > 1.
[STO] [÷], where $x = 0$.

Error 1

Storage register overflow.

Error 2

[STO], [STO] [+], [STO] [×], [STO] [−], [STO] [÷], [RCL], where next digit entered is ≥ 8.

Error 3

Improper statistical operation.
[$\bar{x}$], where $n = 0$.
[s], where $n \leq 1$.
[r], where $n \leq 1$.
[$\hat{x}$], where $n \leq 1$.
[$\hat{y}$], where $n \leq 1$.
[L.R.], where $n \leq 1$

Note: ***Error 3*** is also displayed if division by zero or the square root of a negative number would be required during computation with any of the following formulas:

$$s_x = \sqrt{\frac{M}{n(n-1)}}$$

$$s_y = \sqrt{\frac{N}{n(n-1)}}$$

$$r = \frac{P}{\sqrt{M \cdot N}}$$

$$A = \frac{P}{M}$$

$$B = \frac{M\Sigma y - P\Sigma x}{nM}$$

A and B are the values returned by the operation [L.R.], where $y = Ax + B$.

$$\hat{x} = \frac{P\Sigma x + M(ny - \Sigma y)}{nP}$$

$$\hat{y} = \frac{M\Sigma y + P(nx - \Sigma x)}{nM}$$

where:

$$M = n\Sigma x^2 - (\Sigma x)^2$$

$$N = n\Sigma y^2 - (\Sigma y)^2$$

$$P = n\Sigma xy - \Sigma x\Sigma y$$

Error 4

[GTO], when using a branch to an illegal number.
[GSB], when using a subroutine to an illegal number.

Error 9

Self check failure. When ***Error 9*** is displayed, press any key and another number will show in the display. This number indicates to service personnel what is wrong with your calculator.

Appendix C

Stack Lift and LAST X

Your calculator has been designed to operate in a natural, normal manner. As you have seen as you worked through this handbook, you are seldom required to think about the operation of the automatic memory stack—you merely work through calculations in the same way you would with a pencil and paper, performing one operation at a time.

There may be occasions, however, particularly as you program the calculator, when you wish to know the effect of a particular operation upon the stack. The following explanation and table should help you.

Digit Entry Termination

Most operations on the calculator, whether executed as instructions in a program or pressed from the keyboard, terminate digit entry. This means that the calculator knows that any digits you key in after any of these operations are part of a new number.

Stack Lift

There are three types of operations on the calculator, depending upon how they affect the stack lift. These are stack *disabling* operations, stack *enabling* operations and *neutral* operations.

Disabling Operations

There are only four stack disabling operations on the calculator. These operations disable the stack lift, so that a number keyed in after one of these disabling operations writes over the current number in the displayed X-register and the stack does not lift. These special disabling operations are:

[ENTER↑] [CLx] [Σ+] [Σ−]

Enabling Operations

The bulk of the operations on the keyboard, including one- and two-number mathematical functions like [x^2] and [×], are stack enabling operations. These operations enable the stack lift, so that a number keyed in after one of the enabling operations lifts the stack.

Neutral Operations

Some operations are neutral; that is they do not alter the previous status of the stack lift. Thus, if you have previously disabled the stack lift by pressing [ENTER↑], then press [EEX] and key in a new number. That number will write over the number in the X-register and the stack will not lift. Similiarly, if you have previously enabled the stack lift by executing, say, [x^2], then execute a [GTO] 03 instruction followed by a digit entry sequence, the stack will lift.

The table below lists all legal operations. Enabling operations are designated by a code of "E" disabling operations by "D", and neutral operations by "N". The table also indicates those operations that save the number from the X-register in the LAST X register.

Keystrokes	Enabling, Disabling, or Neutral	Saves x in LAST X
[f] [LST x]	E	No
[f] CLEAR [PREFIX]	N	No
[f] [LN]	E	Yes
[f] [LOG]	E	Yes
[−]	E	Yes
[g] [→P]	E	Yes
[g] [%]	E	Yes
[g] [π]	E	No
[+]	E	Yes
[f] [PAUSE]	N	No
[g] [MANT]	N	No
[f] CLEAR [STK]		No
[g] [NOP]	N	No
[f] [→R]	E	Yes
[R↓]	E	No
[g] [RAD]	N	No
[RCL] 0 through 7	E	No
[RCL] [Σ+]	E	No
[R/S]	N	No
[g] [RTN]	N	No
[g] [s]	E	No
[f] [SCI] 0 through 9	N	No

Keystrokes	Enabling, Disabling, or Neutral	Saves x in LAST X
[Σ+]	D	Yes
[f] [Σ−]	D	Yes
[f] [SIN]	E	Yes
[g] [SIN⁻¹]	E	Yes
[g] [ABS]	E	Yes
[g] [BST]	N	No
[CHS] *	E	No
[f] CLEAR [REG]	N	No
[f] CLEAR [PRGM]	N	No
[CLx]	D	No
[f] [COS]	E	Yes
[g] [COS⁻¹]	E	Yes
[g] [DEG]	N	No
[f] [r]	E	No
[÷]	E	Yes
[f] [L.R.]	E	No
[EEX] *	N	
[f] [ENG] 0 through 9	N	No
[ENTER↑]	D	No
[g] [e^x]	E	Yes
[g] [FRAC]	E	Yes
[f] [FIX] 0 through 9	N	No
[g] [GRD]	N	No
[GSB] 01 through 49	N	No
[GTO] 00 through 49	N	No
[f] [→H.MS]	E	Yes
[g] [→H]	E	Yes
[f] [→RAD]	E	Yes
[g] [INT]	E	Yes
[g] [→DEG]	E	Yes
[f] [$\sqrt{x}$]	E	Yes
[SST]	N	No
[STO] [÷] 0 through 7	E	No
[STO] [−] 0 through 7	E	No
[STO] [+] 0 through 7	E	No
[STO] [×] 0 through 7	E	No

Keystrokes	Enabling, Disabling, or Neutral	Saves x in LAST X
STO 0 through 7	E	No
f $\hat{x}$	E	Yes
f $\hat{y}$	E	Yes
f TAN	E	Yes
g TAN^{-1}	E	Yes
×	E	Yes
g x=0	N	No
g x≠0	N	No
g x<0	N	No
g x>0	N	No
f x=y	N	No
f x≠y	N	No
f x≤y	N	No
f x>y	N	No
g $\bar{x}$	E	No
g x^2	E	Yes
g 1/x	E	Yes
x≷y	E	No
f y^x	E	Yes
g 10^x	E	Yes

* CHS, EEX, . , and the digits 0 through 9 are normally used as part of a digit entry sequence. However, if you press CHS after digit entry has been terminated by another operation, the stack lift will be enabled.

Information

This card must be completed and returned with your calculator and/or recharger, and batteries. Return of this card is considered authorization for Hewlett-Packard to make all repairs necessary to return the calculator to normal working order and to charge the cost of those repairs to the owner for units out of warranty.

Owner's Name _____ Date _____

Street Address _____

City _____ State _____ Zip Code _____

Home Phone _____ Work Phone _____

Date Purchased _____

What Is The Problem Area?

- ☐ Intermittent Problem
- ☐ Printer (Enclose sample)
- ☐ Keyboard
- ☐ Programming
- ☐ Display
- ☐ Recharger/Battery
- ☐ Prerecorded Program/Reader

Describe Problem: _____

Model No. _____ Serial No. _____

Preferred method of payment for out-of-warranty repairs.
If not specified, unit will be returned C.O.D.

☐ VISA ☐ Master Charge

Card No. _____ Expiration Date _____

Name appearing on credit card _____

☐ Purchase Order. Companies with established Hewlett-Packard credit only. (Include copy of purchase order with shipment.)

P.O. Number _____

Authorized Signature _____

Service Card

The warranty period for your calculator and/or accessory is one year from date of purchase. Hewlett-Packard will assume that any unit returned without a copy of proof of purchase (sales slip or validation) is out of warranty. Should service be required, please return your calculator, charger, batteries and this card protectively packaged to avoid in-transit damage. Such damage is not covered under warranty.

Inside the U.S.A.

Return items safely packaged directly to:

Hewlett-Packard
Corvallis Division • Service Department
P.O. Box 999
Corvallis, Oregon 97330

We advise that you insure your calculator and use priority (AIR) mail for distances greater than 300 miles to minimize transit times. All units will be returned by fastest practical means.

Outside the U.S.A.

Where required please fill in the validation below and return your unit to the nearest designated Hewlett-Packard Sales and Service Office. Your warranty will be considered invalid if this completed card is not returned with the calculator.

Model No. _____ Serial No. _____

Date Received _____

Invoice No./Delivery Note No. _____

Sold by:

HP-33E Registration Card

Please complete and return this postage-paid card. Owners with a U.S. address will have their name added to the Hewlett-Packard Personal Calculator Digest *mailing list, and will automatically receive future product information.*

Date Product Received _____ (Month / Day / Year)

Name _____ (First / Initial / Last)

Company _____ (Include Bldg., Division, Room No., etc.)

Street/Box/Route _____

City _____ State _____ Zip Code _____

1. Where was your calculator purchased?

101 ☐ Directly from an HP sales office or factory
102 ☐ By mail from HP
103 ☐ From any retail store

2. Check the ONE category best describing your job function

201 ☐ Top Management
202 ☐ Middle Management/ Supervisory
203 ☐ Professional/Technical
204 ☐ Student
205 ☐ Other (Specify) _____

3. Rank TWO categories of applications for which your calculator will be used. (1 for most important, 2 for second.)

301 ☐ Engineering
302 ☐ Physical Science
303 ☐ Natural Science
304 ☐ Computer Science/Data Processing
305 ☐ Aviation/Marine Navigation
306 ☐ Statistics/Mathematics
307 ☐ Financial Analysis
308 ☐ Real Estate/Lending
309 ☐ Budgeting/Forecasting

If you are outside the United States please mail this card to the nearest Hewlett-Packard Sales Office.

Useful Conversion Factors

The following factors are provided to 10 digits of accuracy where possible. Exact values are marked with an asterisk. For more complete information on conversion factors, refer to *Metric Practice Guide E380-74* by the American Society for Testing and Materials (ASTM).

Length		
1 inch	=	25.4 millimeters*
1 foot	=	0.304 8 meter*
1 mile (statute)†	=	1.609 344 kilometers*
1 mile (nautical)†	=	1.852 kilometers*
1 mile (nautical)†	=	1.150 779 448 miles (statute)†
Area		
1 square inch	=	6.451 6 square centimeters*
1 square foot	=	0.092 903 04 square meter*
1 acre	=	43 560 square feet
1 square mile†	=	640 acres
Volume		
1 cubic inch	=	16.387 064 cubic centimeters*
1 cubic foot	=	0.028 316 847 cubic meter
1 ounce (fluid)†	=	29.573 529 56 cubic centimeters
1 ounce (fluid)†	=	0.029 573 530 liter
1 gallon (fluid)†	=	3.785 411 784 liters*
Mass		
1 ounce (mass)	=	28.349 523 12 grams
1 pound (mass)	=	0.453 592 37 kilogram*
1 ton (short)	=	0.907 184 74 metric ton*
Energy		
1 British thermal unit	=	1 055.055 853 Joules
1 kilocalorie (mean)	=	4 190.02 Joules
1 watt-hour	=	3 600 Joules*
Force		
1 ounce (force)	=	0.278 013 85 Newton
1 pound (force)	=	4.448 221 615 Newtons
Power		
1 horsepower (electric)	=	746 watts*
Pressure		
1 atmosphere	=	760 mm Hg at sea level
1 atmosphere	=	14.7 pounds per square inch
1 atmosphere	=	101 325 Pascals
Temperature		
Fahrenheit	=	1.8 Celsius + 32
Celsius	=	5/9 (Fahrenheit − 32)
Kelvin	=	Celsius + 273.15
Kelvin	=	5/9 (Fahrenheit + 459.67)
Kelvin	=	5/9 Rankine

† U.S. values shown. * Exact values.

Calculator Catalog and Buying Guide Request Card

Thank you for your order.

A friend or associate might also want to know about Hewlett-Packard calculators. If you would like us to send the current issue of the *Hewlett-Packard Personal Calculator Digest* (The HP Magazine and Product Catalog), please write his/her name and address on this postage-paid Request Card.

Primary Interest:

- ☐ Scientific Calculators
- ☐ Fully-Programmable Calculators
- ☐ Business Calculators
- ☐ All

Name _____

Title _____

Company _____

Street _____

City _____ State _____ Zip _____

Valid in U.S. only

430M

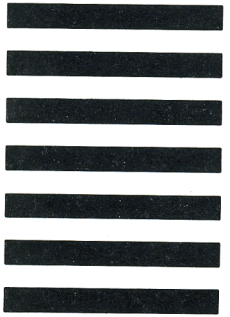
FIRST CLASS
Permit No. 33
Corvallis, Oregon

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

Postage will be paid by:

Hewlett-Packard
1000 N.E. Circle Blvd.
Corvallis, Oregon 97330



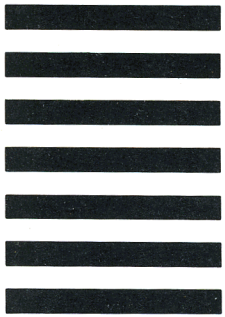
FIRST CLASS
Permit No. 33
Corvallis, Oregon

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

Postage will be paid by:

Hewlett-Packard
1000 N.E. Circle Blvd.
Corvallis, Oregon 97330



1000 N.E. Circle Blvd., Corvallis, OR 97330

For additional sales and service information contact your local Hewlett-Packard Sales Office or call 800/648-4711. (In Nevada call 800/992-5710.)

00033-90001

Printed in U.S.A.