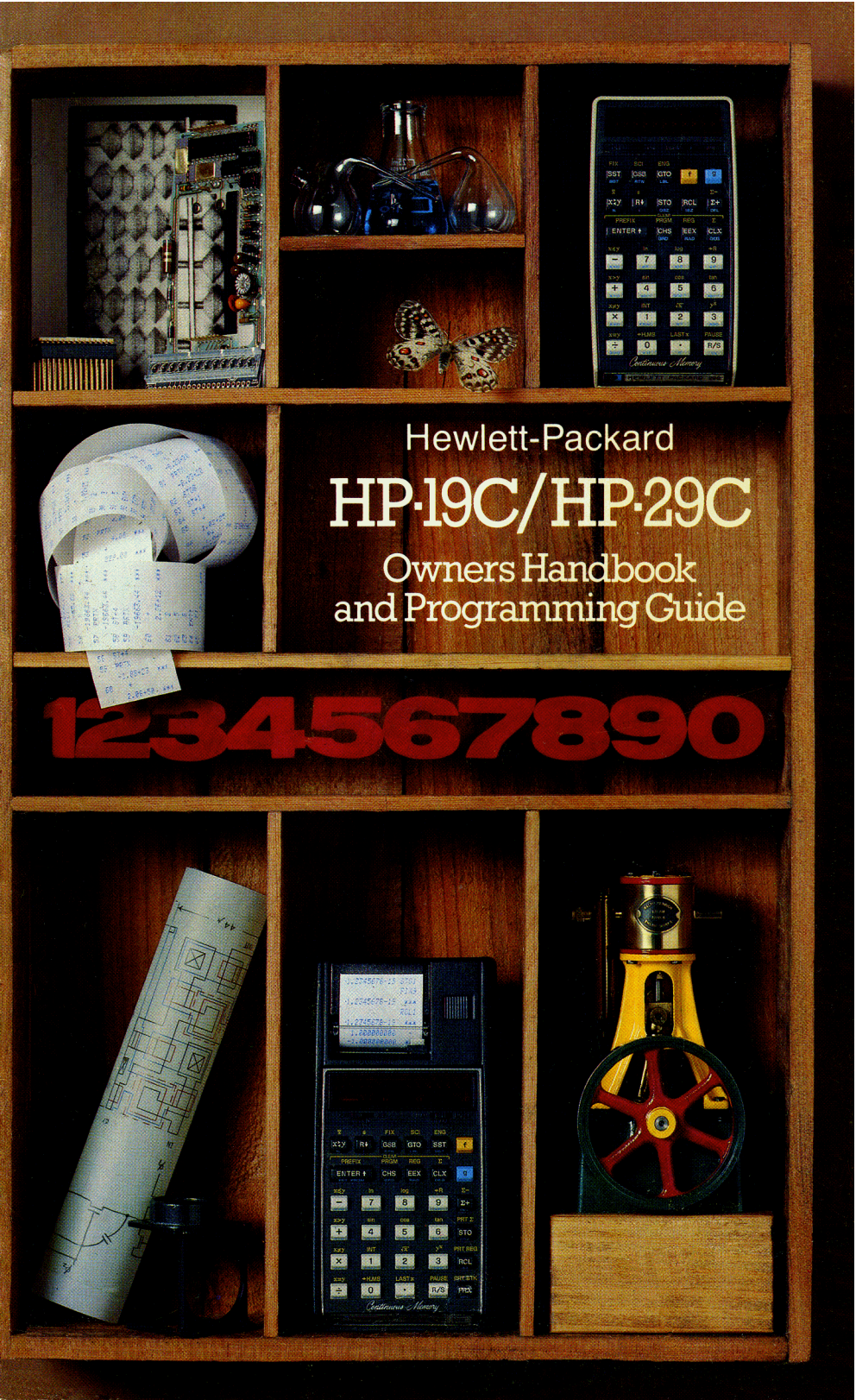

Hewlett-Packard
HP-19C/HP-29C
Owners Handbook
and Programming Guide
1234567890

"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

We wish to thank the following for their generous contributions
to the memory box pictured on the front cover of this handbook:

Mario Boschetto, Department of Chemistry,
Oregon State University, Corvallis, Oregon

Entomology Museum, Department of Entomology,
Oregon State University, Corvallis, Oregon

Ralph Henrich, Ralph Henrich Engine Works,
Corvallis, Oregon

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 world-wide, both pocket and desktop models. Their owners include Nobel laureates, astronauts, mountain climbers, businessmen, doctors, students, and housewives.

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-19C Printing and HP-29C Programmable Scientific Calculators

Owner's Handbook and Programming Guide

July 1977

5955-2110

Printed in U.S.A.

Contents

HP-19C/HP-29C
Programmable Scientific Calculator

Fold Out

Primary storage registers, program memory, displayed X-register, and display format are maintained by Continuous Memory.

Primary Storage Registers

		Address
R_0		0
R_1		1
R_2		2
R_3		3
R_4		4
R_5		5
R_6		6
R_7		7
R_8		8
R_9		9
$R_{.0}$		10 n
$R_{.1}$		11 Σx
$R_{.2}$		12 Σx^2
$R_{.3}$		13 Σy
$R_{.4}$		14 Σy^2
$R_{.5}$		15 Σxy

Statistical Registers

Indirect Storage Registers

$R_{(16)}$		16
$R_{(17)}$		17
$R_{(18)}$		18
$R_{(19)}$		19
$R_{(20)}$		20
$R_{(21)}$		21
$R_{(22)}$		22
$R_{(23)}$		23
$R_{(24)}$		24
$R_{(25)}$		25
$R_{(26)}$		26
$R_{(27)}$		27
$R_{(28)}$		28
$R_{(29)}$		29

Program Memory

	HP-19C	HP-29C
00		
01	64	74
02	64	74
03	64	74
04	64	74
96	64	74
97	64	74
98	64	74

Automatic Memory Stack

Registers

T .....
Z .....
Y .....
X .....

Display

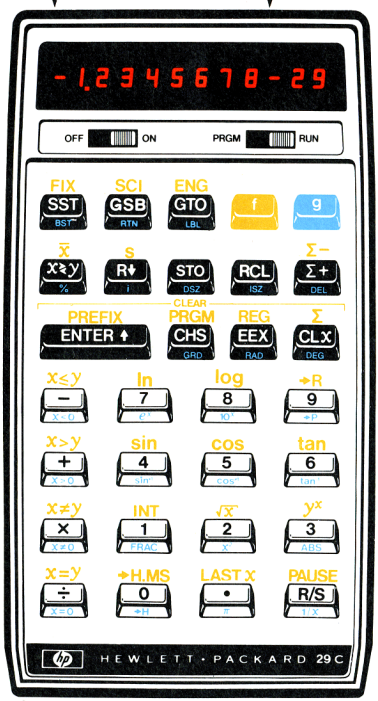
0.0000000 00

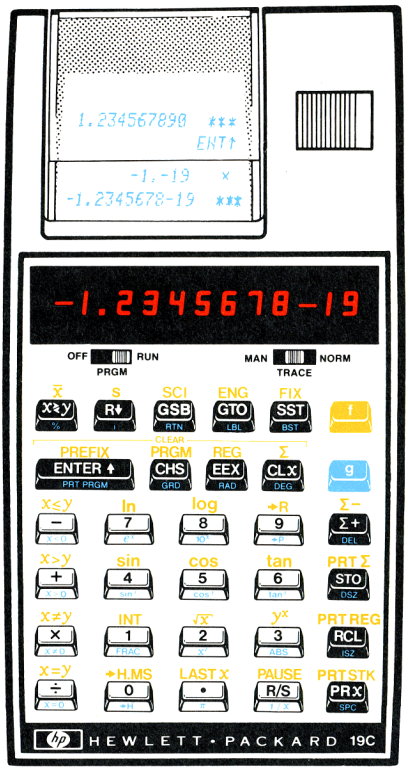
LAST X

0.0000000 00

Exponent of 10 sign

Sign | Mantissa | Exponent of 10





Function Key Index

HP-19C:
Manual RUN Mode. OFF-PRGM-RUN Switch OFF PRGM RUN set to RUN.
HP-29C:
Manual RUN Mode. PRGM-RUN Switch PRGM RUN set to RUN.

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. All references to HP-19C printer functions are printed in brown.

OFF PRGM RUN HP-19C
OFF-PRGM-RUN Switch **(page 20).**

OFF ON HP-29C
OFF-ON Switch **(page 20).**

MAN TRACE NORM HP-19C
Print Mode switch. Selects printing option **(page 22).**

f Pressed before function key, selects gold function printed above key **(page 20).**

g Pressed before function key, selects blue function printed on lower face of key **(page 20).**

CLEAR PREFIX after f, g, STO, RCL, or GTO calcels that key **(page 112).**

HP-19C Printing Functions

SPC advances paper one or more spaces without printing **(page 111).**

PRT REG Prints contents of all storage registers **(page 62).**

PRT STK Prints contents of automatic memory stack **(page 44).**

PRX Prints contents of displayed X-register **(page 36).**

PRT Σ Prints contents of statistical registers (storage registers $R_{.0}$ through $R_{.5}$) **(page 85).**

PRT PRGM Print program. Prints contents of program memory, beginning with current step and continuing until two consecutive R/S instructions are encountered or step 98 is printed **(page 111).**

Digit Entry

ENTER↑ Enters a copy of number displayed in X-register into Y-register. Used to separate numbers **(page 47).**

CHS Changes sign of mantissa or exponent of 10 in displayed X-register **(page 21).**

EEX Enter exponent. After pressing, next numbers keyed in are exponents of 10 **(page 39).**

0 through 9 Digit keys **(page 21).**

• Decimal point **(page 21).**

Number Manipulation

R↓ Rolls down contents of stack for viewing in displayed X-register **(page 45).**

X≷Y Exchanges contents or X- and Y-registers of stack **(page 46).**

CLX Clears contents of displayed X-register to zero **(page 47).**

Display Control

FIX Fixed point display. Followed by a number key, selects fixed point notation display **(page 33).**

SCI Scientific display. Followed by a number key, selects scientific notation display **(page 34).**

ENG Engineering display Followed by a number key, selects engineering notation display **(page 34).**

Number Alteration

ABS Gives absolute value of number in displayed X-register **(page 66).**

INT Leaves only integer portion of number in displayed X-register by truncating fractional portion **(page 66).**

FRAC Leaves only fractional portion of number in displayed X-register by truncating integer portion **(page 67).**

Manual Storage

STO Store. Followed by number key, decimal point and number key, or i stores displayed number in storage register specified. Also used to perform storage register arithmetic **(page 61).**

RCL Recall. Followed by number key, decimal point and number key, or i recalls value from storage register specified into the displayed X-register **(page 61).**

CLEAR REG Clears contents of all storage registers **(page 63).**

LAST X Recalls number displayed before the previous operation back into the displayed X-register **(page 55).**

Mathematics

$\sqrt{x}$ Computes square root of number in displayed X-register **(page 68).**

x^2 Computes square of number in displayed X-register **(page 68).**

$1/x$ Computes reciprocal of number in displayed X-register **(page 67).**

π Places value of pi (3.141592654) into displayed X-register **(page 69).**

$+$ $-$ $\times$ $\div$ Arithmetic operators **(page 24).**

Trigonometry

DEG Sets decimal degrees mode for trigonometric functions **(page 70).**

RAD Sets radians mode for trigonometric functions **(page 70).**

GRD Sets grads mode for trigonometric functions **(page 70).**

sin cos tan Computes sine, cosine, or tangent of value in displayed X-register **(page 71).**

$\sin^{-1}$ $\cos^{-1}$ $\tan^{-1}$ Computes arc sine, arc cosine, or arc tangent of number in displayed X-register **(page 71).**

→H.MS Converts decimal hours or degrees to *hours, minutes, seconds* or *degrees, minutes, seconds* **(page 71).**

→H *Converts hours, minutes, seconds* or *degrees, minutes, seconds* to decimal hours or degrees **(page 72).**

Polar/Rectangular Conversion

→P Converts x, y rectangular coordinates placed in X- and Y-registers to polar magnitude r and angle θ **(page 75).**

→R Converts polar magnitude r and angle θ in X- and Y-registers to rectangular x and y coordinates **(page 75).**

Logarithmic and Exponential

y^x Raises number in Y-register to power of number in displayed X-register **(page 80).**

10^x Common antilogarithm. Raises 10 to power of number in displayed X-register **(page 79).**

e^x Natural antilogarithm. Raises e (2.718281828) to power of number in displayed X-register **(page 79).**

log Computes common logarithm (base 10) of number in displayed X-register **(page 79).**

ln Computes natural logarithm (base e, 2.718281828) of number in displayed X-register **(page 79).**

Statistics

Σ+ Accumulates numbers from X- and Y-registers into storage registers $R_{.0}$ through $R_{.5}$ **(page 82).**

Σ- Subtracts x and y values from storage registers $R_{.0}$ through $R_{.5}$ for correcting Σ+ accumulations **(page 89).**

$\bar{x}$ Computes mean (average) of x and y values accumulated by Σ+ **(page 85).**

CLEAR Σ Clears storage registers used for accumulations ($R_{.0}$ through $R_{.5}$) to zero **(page 82).**

s Computes sample standard deviations of x and y values accumulated by Σ+ **(page 87).**

Percentage

% Computes x% of y **(page 69).**

Indirect Control

i When preceded by GTO, GSB, STO, or RCL, the address or control value for that function is specified by the current number in R_0 **(page 164).**

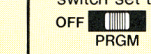
ISZ Increment R_0, skip if zero. Adds 1 to contents of R_0. Skips one step if contents are then zero **(page 156).**

DSZ Decrement R_0, skip if zero. Subtracts 1 from contents of R_0. Skips one step if contents are then zero **(page 156).**

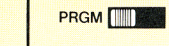
Programming Key Index

PROGRAM Mode

HP-19C OFF-PRGM-RUN switch set to PRGM.



HP-29C: PRGM-RUN switch set to PRGM.



All function keys except the functions shown below are loaded into program memory when pressed.

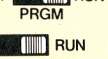
Active Keys:

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 • n n positions calculator to step n n of program memory. No instructions are executed **(page 118).**

Automatic RUN Mode

HP-19C OFF-PRGM-RUN switch set to RUN.



HP-29C PRGM-RUN switch set 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.

All references to HP-19C Printer functions are printed in brown.

Pressed from keyboard:

GTO Go to. Followed by • n n sets calculator to step n n of program memory without executing instructions. Followed by label designator (0 through 9 or i) causes calculator to search downward through program memory to first designated label. No instructions are executed. **(page 118).**

GSB Go to subroutine. Followed by label designator, 0 through 9, i, causes calculator to start executing instructions, beginning with designated label **(page 101).**

Executed as a recorded program instruction:

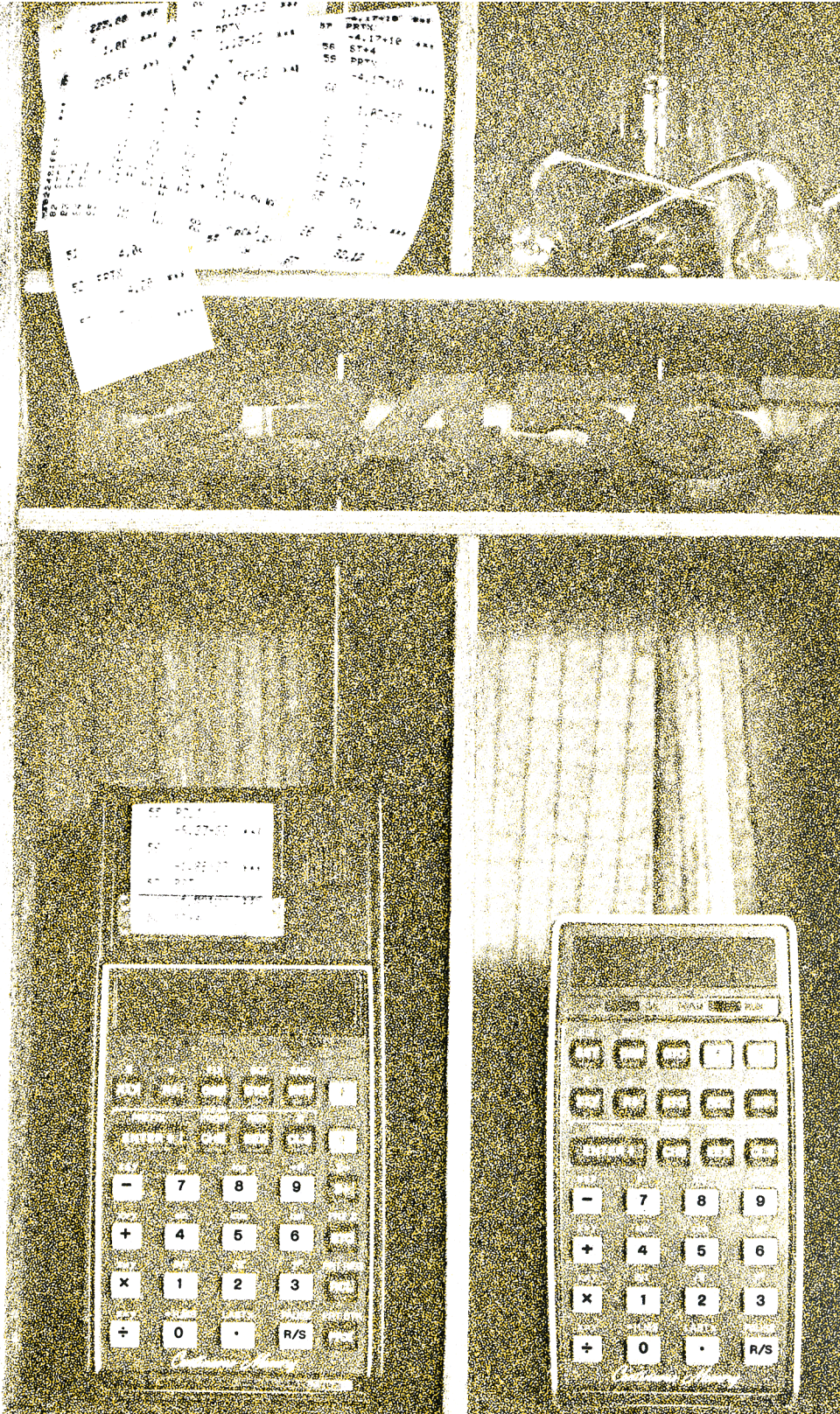
0 1 2 3 4 5 6 7 8 9 Label designators. When preceded by LBL, define beginning of routine. When preceded by GTO or GSB, cause calculator to stop execution, search downward through program memory to first designated label, and resumes execution there **(page 146).**

GTO Go to. Followed by label designator 0 through 9 or i, causes calculator to stop execution, search through program memory to first designated label, and resume execution there **(page 146).**

GSB Go to subroutine. Followed by label designator 0 through 9 or i, causes calculator to search through program memory to first designated label and execute that section of program memory as a subroutine **(page 146).**

PROGRAM Mode	Automatic RUN Mode	
Active keys:	**Pressed from the keyboard:**	**Executed as a recorded program instruction:**
	[RTN] Return. Sets calculator to step 00 of program memory **(page 15).**	[RTN] Return. If executed as a result of pressing [GSB] and a label designator or execution of a [GTO] instruction, stops execution and returns control to keyboard. If executed as a result of a [GSB] instruction, returns control to next step after the [GSB] instruction **(page 146).**
CLEAR [PRGM] Clear program. Clears program memory to all [R/S] instructions, sets calculator to step 00 **(page 98).**	CLEAR [PRGM] After [f] prefix key, cancels that key. After other keys, does nothing. Does not disturb program memory or calculator status **(page 112).**	
		[PAUSE] Stops program execution and displays contents of X-register for 1 second, then resumes program execution **(page 142).**
[BST] Back step. Moves calculator back one step in program memory **(page 120).**	[BST] Back step. Sets calculator to and displays step number and keycode of previous program memory step when pressed; displays original contents of X-register when released. No instructions are executed **(page 120).**	
		[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 step of program memory. If false, calculator skips one step before resuming execution **(page 133).**

PROGRAM Mode	Automatic RUN Mode	
Active keys:	**Pressed from the keyboard:**	**Executed as a recorded program instruction:**
SST Single step. Moves calculator forward one step in program memory **(page 117).**	SST Single step. Displays step number and key-code of current program memory step when pressed; executes instruction, displays result, and moves calculator to next step when released **(page 115).**	
	R/S Run/stop. Begins execution from current step of program memory. Stops execution if program is running **(page 140).**	R/S Run/stop. Stops program execution **(page 140).**
DEL Delete. Deletes current instruction from program memory. All subsequent instructions move up one step **(page 121).**	DEL After g prefix key, cancels that key. After other keys, does nothing. Does not disturb program memory or calculator status **(page 122).**	
CLEAR PREFIX, After f, g, STO, RCL, or GTO, cancels that key **(page 112).**	CLEAR PREFIX, After f, g, STO, RCL, or GTO, cancels that key **(page 112).**	
PRT PRGM. Print program. Prints contents of program memory beginning with current step and continuing until two consecutive R/S instructions are encountered or step 98 is printed **(page 111).**	PRT PRGM. Print program. Prints contents of program memory beginning with current step and continuing until two consecutive R/S instructions are encountered or step 98 is printed **(page 111).**	
	Any key. Pressing any key on the keyboard stops execution of a running program.	


7
8
9
4
5
6
1
2
3
0
R/S

Meet the HP-19C and HP-29C

Congratulations!

With your purchase of the HP-19C/HP-29C Programmable Scientific Calculator with Continuous Memory, you have acquired a truly versatile and unique calculating instrument. Using the Hewlett-Packard RPN logic system that slices with ease through the most difficult equations, the calculator is without parallel:

As a scientific calculator. The HP-19C/HP-29C 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 the simple, step-by-step instructions in the *HP-19C/HP-29C Applications Book,* 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-19C/HP-29C is so easy to program and use that it requires no prior programming experience or knowledge of arcane programming languages. Yet even the most sophisticated computer experts marvel at the programming features of the calculator:

- Continuous Memory means that programs can be remembered by the calculator—even when the power switch is OFF.
- 16 non-volatile data storage registers.
- 14 volatile indirect data storage registers.
- 98 steps of non-volatile program memory.
- Fully merged prefix and function keys that mean more programming per step.
- Easy-to-use editing features for correcting and modifying programs.
- Powerful unconditional and conditional branching.
- Three levels of subroutines and 10 labels.
- Indirect storage, recall, branching, and subroutine calls.
- Printer to record results, list programs, or trace executing programs (HP-19C).

And in addition, the HP-19C/HP-29C can be operated from its rechargeable battery pack for *complete portability,* anywhere.

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

Manual Problem Solving

To get the feel of your new calculator, try a few simple calculations. First, set the switches that are located directly below the display window as follows:

HP-19C: Set the OFF-PRGM-RUN switch OFF PRGM RUN to RUN.
Set the Print Mode switch MAN TRACE NORM to MAN.

HP-29C: Set the OFF-ON switch OFF ON to ON.
Set the PRGM-RUN switch OFF PRGM RUN to RUN.

If ***Error*** is displayed when you turn the calculator on, don't worry, it simply means power has been interrupted at some time. Just press any key to clear the ***Error*** and proceed. If the display is blank when you turn it on, refer to AC Line Operation, page 182.

Press f FIX 2 so your display will match the displays in the following problems.

To solve:	Press:	Display:
$5 + 6 = 11$	5 ENTER↑ 6 +	*11.00*
$8 \div 2 = 4$	8 ENTER↑ 2 ÷	*4.00*
$7 - 4 = 3$	7 ENTER↑ 4 −	*3.00*
$9 \times 8 = 72$	9 ENTER↑ 8 ×	*72.00*
$\frac{1}{5} = 0.20$	5 g 1/x	*0.20*
Sine of $30° = 0.50$	30 f sin	*0.50*

Now let's try something a little more involved. To calculate the surface area of a sphere, the formula $A = \pi d^2$ can be used, where:

A is the surface area of the sphere,
d is the diameter of the sphere,
π is the value of pi, 3.141592654.

Ganymede, one of Jupiter's 12 moons, has a diameter of 3200 miles. You can use the calculator to manually compute the area of Ganymede. Merely press the following keys in order.

Press	Display	
3200	*3200.*	Diameter of Ganymede.
g x^2	*10240000.00*	Square of the diameter.
g π	*3.14*	The quantity π.
×	*32169908.78*	Area of Ganymede in square miles.

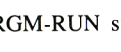
Programmed Problem Solving

After calculating the surface area of Ganymede, one of Jupiter's 12 moons, suppose you decided you wanted 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.

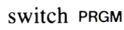
To calculate the area of a sphere using a program, you should first write the program, then you must load the program into the calculator, and finally you *run* the program to calculate each answer.

Writing the Program. You have already written it! A program is nothing more than the series of keystrokes you would execute to solve the same problem manually. Two additional operations, a *label* and a *return* are used to define the beginning and end of the program.

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

1. HP-19C: Slide the OFF-PRGM-RUN switch 

to PRGM (program).

   HP-29C: Slide the PRGM-RUN switch 

to PRGM (program).

2. Press f CLEAR PRGM to clear the calculator.

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.)

Press	
g LBL 0	Defines the beginning of the program.
g x^2 g π ×	These are the same keys you pressed to solve the problem manually.
g RTN	Defines the end of the program.

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. HP-19C: Slide the OFF-PRGM-RUN switch 

to RUN.

   HP-29C: Slide the PRGM-RUN switch

to RUN.

2. Key in the value of the diameter.
3. Press GSB 0 to run the program.

When you press GSB 0, 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 area of Ganymede, with a diameter of 3200 miles:

Press	Display	
3200	*3200.*	
GSB 0	*32169908.78*	Square miles.

With the program you have loaded, you can now calculate the area of any of Jupiter's moons—in fact, of *any* sphere—using its diameter. You have only to leave the calculator in RUN mode and key in the diameter of each sphere for which you want the area, then press GSB 0. For example, to compute the surface area of Jupiter's moon Io, with a diameter of 2310 miles:

Press	Display	
2310 GSB 0	*16763852.56*	Square miles.

For the moons Europa, diameter 1950 miles, and Callisto, diameter 3220 miles:

Press	Display	
1950 GSB 0	*11945906.07*	Area of Europa in square miles.
3220 GSB 0	*32573289.27*	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 the HP-19C/HP-29C can remember up to 98 separate operations (and many more keystrokes, since many operations require two or three keystrokes) and execute them simply by pressing GSB followed by the label number (0 through 9). By using, say, LBL 0 for one program, LBL 1 for another, etc., your calculator can contain many different programs at one time.

What Continuous Memory Means to You

Your calculator contains Continuous Memory—one of the newest, most advanced memory systems available in a scientific calculator. Continuous Memory means the program memory and primary storage registers stay "on" when your calculator is turned off. You can store your favorite program (or programs) for days or weeks!

Continuous Memory is especially convenient when you want to retain data, save battery life, or customize your calculator (e.g., if you use 20% of your programs to solve 80% of your problems). You save considerable time because you don't have to key in those common programs again and again—they are stored in your calculator. Continuous Memory reduces human entry errors too; fewer keystrokes mean fewer chances of making inadvertent errors.

Perhaps the most important advantage of Continuous Memory is that it enables you to customize or personalize your calculator. The easiest way to customize your calculator is to make a list of the problems you encounter most frequently, rate them in order of priority, then write and save the specialized programs to solve those problems. Whenever you encounter a repetitive problem set, you just write the program once, then use it at different times. You can even preserve one or two favorite programs in the calculator.

Besides saving programs, Continuous Memory lets you store data in the general-purpose storage registers. Constants, accumulations, and intermediate answers can be retrieved whenever you need them. In addition, even the number in the displayed X-register is saved, along with the display format!

Continuous Memory also helps save battery life in many situations. If your calculator is left off, Continuous Memory can store your programs for 1 month or longer. When you do use your calculator, keying in fewer programs means less time that the display is on—hence, less battery drain.

Using this Handbook

New to Hewlett-Packard calculators? Part One of this handbook has been designed to teach you to use your HP-19C/HP-29C as a powerful scientific calculator. By working through these sections of the handbook, you'll learn every function that you can use to calculate answers manually, and you'll come to appreciate the calculating efficiency of the Hewlett-Packard logic system with RPN. And since the programmability of the calculator stems from its ability to remember a series of manual keystrokes, Part One, Using Your HP-19C/HP-29C Calculator, is invaluable in laying the groundwork for Part Two, Programming The HP-19C/HP-29C.

Previous HP user? If you've already used Hewlett-Packard pocket or desktop calculators with RPN, you may want to familiarize yourself with the unique printer options of the HP-19C by reading page 22, and then turn directly to Part Two, Programming The HP-19C/HP-29C. Later, though, you will undoubtedly wish to peruse Part One at your leisure in order to discover the many calculating advantages of your calculator.

Whether an old hand or a novice, you'll find the Function and Programming Key Index beginning on page 6 invaluable as a quick reference guide, a programming guide, or even to illustrate the features of your calculator to your friends.

Please note that all references to the calculator printer are applicable only to the HP-19C. All other *operations* are identical.

Note: References to printer functions are printed in brown and are applicable only to the HP-19C.

PART ONE
Using Your HP-19C/HP-29C Calculator

Section 1

Getting Started

Your HP-19C/HP-29C is shipped fully equipped, including a battery pack. Although the calculator is completely portable, if you want to use it on battery power alone, you should connect the ac adapter/recharger and charge the battery for 6 hours first. Whether you operate from battery power or from the ac adapter/recharger, *the battery pack must always be in the calculator*. The battery pack is never in danger of being overcharged.

To begin:

HP-19C: Slide the OFF-PRGM-RUN switch OFF PRGM RUN to RUN.

Slide the Print Mode switch MAN TRACE NORM to MAN.

HP-29C: Slide the PRGM-RUN switch PRGM RUN to RUN.

Display

Numbers that you key into the calculator and intermediate and final answers are always seen in the bright red display. Should you see the display ***Error*** the first time you turn the calculator on, do not worry—it means that power to the calculator has been interrupted at some time. Merely press any key on the keyboard to clear the display, then continue.

The displays illustrating most examples in this handbook are shown to two decimal places, like this: ***0.00***. As you will see, numbers can be seen in a wide variety of formats with your calculator, but if you want the display on your calculator to look like the ones shown on the next few pages, press f FIX 2 now.

Keyboard

Each key on the keyboard can perform as many as three different functions. One function is indicated on the flat plane of the key face, while the second and third functions may be indicated by printed symbols in gold and blue, respectively: gold above the key, and blue on the lower face of the key.

There are two *prefix* keys, f and g. By pressing one of these prefix keys before pressing a function key, you select the function printed in gold or blue.

To select the function printed on the face of the key, press the key.

To select the function printed in gold above the key, first press the gold prefix key f, then press the function key.

To select the function printed in blue on the lower face of the key, first press the blue prefix key g, then press the function key.

In this handbook, the key functions appear in the appropriate color outlined by a box, like this: ÷, R↓, P→. Digit keys (0 through 9) appear like this: 1, 2, 3... .

Keying In Numbers

Key in numbers by pressing the number keys in sequence, just as though you were writing on a piece of paper. The decimal point must be keyed in if it is part of the number (unless it is to be right of the last digit). Remember, if you want your display to be the same as those shown here, press f FIX 2.

For example to key in 148.84:

Press	Display
148.84	*148.84*

The number, 148.84 is seen in the display.

Negative Numbers

To key in a negative number, press the keys for the number, then press CHS *(change sign)*. The number, preceded by a minus (–) sign, will appear in the display. For example, to change the sign of the number now in the display:

Press	Display
CHS	*–148.84*

You can change the sign of either a negative or a positive nonzero number in the display. For example, to change the sign of the –148.84 now in the display back to positive:

Press	Display
CHS	*148.84*

Notice that only negative numbers are given a sign in the display.

Clearing

You can clear any numbers that are in the display by pressing CLx *(clear X)*. This key erases the number in the display and replaces it with *0.00*.

Press	Display
CLx	*0.00*

If you make a mistake while keying in a number, clear the entire number string by pressing CLx. Then key in the correct number.

Printer (HP-19C)

The printer has three modes of operation, which you control using the Print Mode switch MAN / TRACE / NORM:

- With the Print Mode switch MAN / TRACE / NORM set to MAN *(manual)*, the printer is idle and does not print unless you press the PRX key or one of the other PRINT functions. This mode gives greatest economy of paper and battery power.

- With the Print Mode switch MAN / TRACE / NORM set to NORM *(normal)*, the calculator records a history of the calculation sequence so that you can reconstruct your problem. In this mode you see digit entries and functions, but intermediate and final answers are not printed unless you press the PRX key.

- With the Print Mode switch MAN / TRACE / NORM set to TRACE, the calculator prints numbers, functions, and intermediate and final answers, just as they are seen in the display. The results of functions are printed with the symbol *** to the right of the number. In addition, when in PRGM *(program)* mode, the printer will record mnemonics of your keystrokes as you enter your program into program memory.

To advance the printer paper, press g SPC and simply hold the SPC key down until the paper has advanced the desired amount. To replace the paper roll, refer to Your HP-19C Printer in appendix A of this handbook.

No matter what print mode you choose, you seldom have to worry about "overrunning" the printer when you are calculating. Your HP-19C contains a key buffer that "remembers" up to seven keystrokes—no matter how fast you press the keys.

Functions

In spite of the dozens of functions available on the keyboard, you will find the calculator functions simple to operate by using a single, all-encompassing rule: *When you press a function key, the calculator immediately executes the function written on the key.*

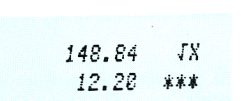
> Pressing a function key causes the calculator to immediately perform that function, and display the result.

The best way to see how simple functions operate on your HP-19C is with the Print Mode switch set to TRACE to give you a complete record of inputs, functions, and answers.

Slide the HP-19C Print Mode switch MAN / TRACE / NORM to TRACE now and make sure there is a roll of paper in the printer. If there is no paper, refer to Your HP-19C Printer in appendix A (page 186).

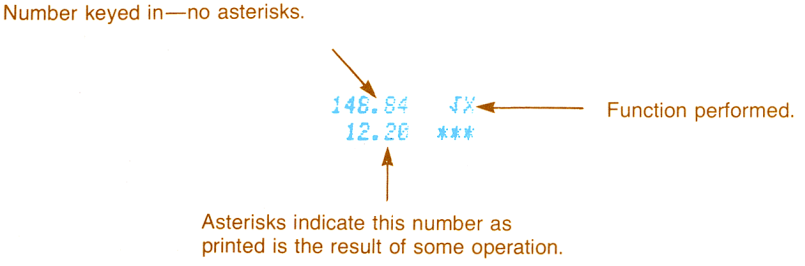
For example, to calculate the square root of 148.84 merely:

Press	Display
148.84	***148.84***
f √x	***12.20***



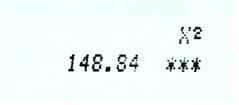
Let's look briefly at the printed copy of that problem to see the simple way that the HP-19C printer duplicates your calculations.

The paper tapes are printed just as you read, from left to right and top to bottom. The number, 148.84, is printed exactly as you keyed it in. A symbol for the function performed, [√x] is printed next to it. The answer, 12.20, is printed with a three-asterisk label to its right, indicating that the HP-19C performed some operation in order to obtain the number as it is printed.



Now let's continue: To square the result of the previous calculation:

Press	Display
g [x²]	*148.84*



[√x], and [x²] are examples of one-number fucntion keys; that is, keys that execute upon a single number. All function keys in the calculator operate upon either one number or two numbers at a time (except for statistics keys like [Σ+] and [s]—more about these later).

Function keys operate upon either one number or two numbers.

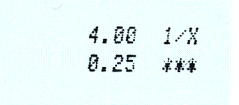
One-Number Functions

To use any one-number function key:

1. Key in the number.
2. Press the prefix key, then the function key.

For example, to use the function [1/x], you first key in the number represented by x, press the prefix key [g], then press the function key. To calculate ¼, key in 4 (the x-number), press [g], and then press [1/x].

Press	Display
4	*4.*
g [1/x]	*0.25*



Now try these other one-number function problems. Remember, *first key in the number,* press the appropriate prefix key, *then press the function:*

$\frac{1}{25}$	= **0.04**	
$\sqrt{2500}$	= **50.00**	
10^5	= **100000.00**	(Use the [10^x] key.)
$\sqrt{3204100}$	= **1790.00**	
log 12.58925411	= **1.10**	
71^2	= **5041.00**	

Two-Number Functions

Two-number functions are functions that must have two numbers present in order for the operation to be performed. [+], [−], [×], and [÷] are examples of two-number function keys. You cannot add, subtract, multiply, or divide unless there are two numbers present in the calculator. Two-number functions work the same way as one-number functions—that is, the operation occurs when the function key is pressed. Therefore, *both numbers must be in the calculator before the function key is pressed.*

When more than one number must be keyed into the calculator before performing an operation, the [ENTER↑] key is used to separate the two numbers.

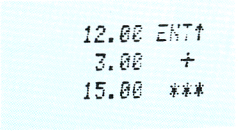
> Use the [ENTER↑] key whenever more than one number must be keyed into the calculator before pressing a function.

If you key in only one number, you never need to press [ENTER↑]. To place two numbers into the calculator and perform an operation:

1. Key in the first number.
2. Press [ENTER↑] to separate the first number from the second.
3. Key in the second number.
4. Press the function key to perform the operation.

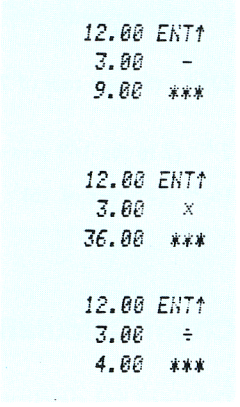
For example, to add 12 and 3:

Press	Display	
12	**12.**	The first number.
[ENTER↑]	**12.00**	Separates the first number from the second.
3	**3.**	The second number.
[+]	**15.00**	The function, and answer.



Other arithmetic functions are performed the same way:

To perform	Press	Display
12 − 3	12 ENTER↑ 3 −	9.00
12 × 3	12 ENTER↑ 3 ×	36.00
12 ÷ 3	12 ENTER↑ 3 ÷	4.00



The y^x key is also a two-number operation. It is used to raise numbers to powers, and you can use it in the same simple way that you use every other two-number function key:

1. Key in the first number.
2. Press ENTER↑ to separate the first number from the second.
3. Key in the second number (power).
4. Perform the operation (press f y^x).

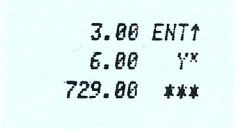
When working with any function key (including y^x), you should remember that the displayed number is always designated by x on the function key symbols.

The number *displayed* is always x.

So $\sqrt{x}$ means square root of the displayed number, $1/x$ means $\frac{1}{\text{displayed number}}$, etc.

Thus, to calculate 3^6:

Press	Display	
3	3.	
ENTER↑	3.00	
6	6.	x, the displayed number, is now six.
f y^x	729.00	The answer.



Now try the following problems using the y^x key, keeping in mind the simple rules for two-number functions:

16^4 (16 to the 4th power)	= 65536.00	
81^2 (81 squared)	= 6561.00	(You could also have done this as a one-number function using x^2.)
$225^{.5}$ (Square root of 225)	= 15.00	(You could also have done this as a one-number function using $\sqrt{x}$.)
2^{16} (2 to the 16th power)	= 65536.00	
$16^{.25}$ (4th root of 16)	= 2.00	

Chain Calculations

The speed and simplicity of operation of the Hewlett-Packard logic system become most apparent during chain calculations. Even during the longest of calculations, you still perform only one operation at a time, and you see the results as you calculate—the Hewlett-Packard automatic memory stack stores up to four intermediate results inside the calculator until you need them, then inserts them into the calculation. This system makes the process of working through a problem as natural as it would be if you were working it out with pencil and paper but the calculator takes care of the hard part.

For example, solve (12 + 3) × 7.

If you were working the problem with a pencil and paper, you would first calculate the intermediate result of (12 + 3)...

~~(12 + 3)~~ × 7 =

15

...and then you would multiply the intermediate result by 7.

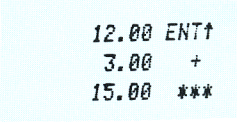
~~(12 + 3)~~ × 7 =

15 × 7 = 105

You work through the problem exactly the same way with the calculator, one operation at a time. You solve for the intermediate result first...

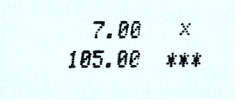
(12 + 3)

Press	Display	
12	12.	
ENTER↑	12.00	
3	3.	
+	15.00	Intermediate result.



... and then solve for the final answer. You don't need to press ENTER↑ to store the intermediate result—the calculator automatically stores it when you key in the next number. To continue...

Press	Display	
7	*7.*	The intermediate result from the preceding operation is automatically stored inside the calculator when you key in this number.
⊠	*105.00*	Pressing the function key multiplies the new number and the intermediate result, giving you the final answer.

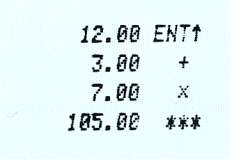


Because the calculator stores intermediate results automatically, you don't need to print or remember them. You can slide the Print Mode switch to NORM to preserve a record of your calculations, and then press PRX to print the final answer. The final answer will be in the displayed X-register on both the HP-19C and HP-29C.

For example, when you solved the above problem in TRACE mode, you preserved *all* intermediate and final results. To solve the same problem and preserve only a history of the calculations:

Slide the HP-19C Print Mode switch MAN TRACE NORM to NORM.

Press	Display	
12	*12.*	
ENTER↑	*12.00*	
3	*3.*	
⊞	*15.00*	
7	*7.*	
⊠	*105.00*	
PRX (HP-19C)	*105.00*	Preserves the final answer in your printed record.



Now try these problems. Notice that for each problem you only have to press ENTER↑ to insert a pair of numbers into the calculator—each subsequent operation is performed using a new number and an automatically stored intermediate result.

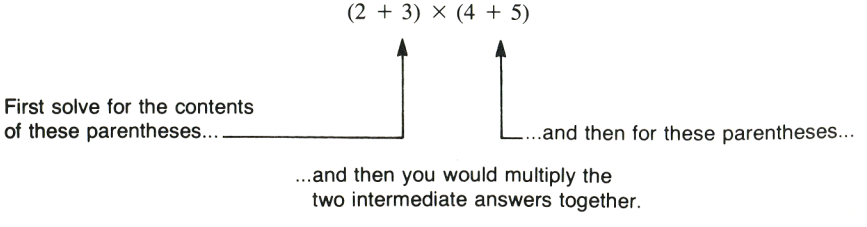
To solve	Press	Display
$\frac{(2 + 3)}{10}$	2	2.
	ENTER↑	2.00
	3	3.
	⊞	5.00
	10	10.
	÷	0.50
	PRX (HP-19C)	0.50
$3\ (16 - 4)$	16	16.
	ENTER↑	16.00
	4	4.
	⊟	12.00
	3	3.
	×	36.00
	PRX (HP-19C)	36.00
$\frac{14 + 7 + 3 - 2}{4}$	14	14.
	ENTER↑	14.00
	7	7.
	⊞	21.00
	3	3.
	⊞	24.00
	2	2.
	⊟	22.00
	4	4.
	÷	5.50
	PRX (HP-19C)	5.50

```
2.00 ENT↑
3.00   +
10.00  ÷
0.50   ***
```

```
16.00 ENT↑
4.00   -
3.00   x
36.00  ***
```

```
14.00 ENT↑
7.00   +
3.00   +
2.00   -
4.00   ÷
5.50   ***
```

Problems that are even more complicated can be solved in the same simple manner, using the automatic storage of intermediate results. For example, to solve $(2 + 3) \times (4 + 5)$ with a pencil and paper, you would:



You work through the problem the same way with your HP-19C/HP-29C. First you solve for the intermediate result of (2 + 3)...

Press	Display	
2	2.	
ENTER↑	2.00	
3	3.	
+	5.00	Intermediate result.

```
2.00 ENT↑
3.00   +
```

Then add 4 and 5:

(Since you must now key in another *pair* of numbers before you can perform a function, you use the ENTER↑ key again to separate the first number of the pair from the second.)

Procedure	Press	Display
(2 + 3) × (4 + 5) 5 9	4 ENTER↑ 5 +	9.00

```
4.00 ENT↑
5.00   +
```

Then multiply the intermediate answers together for the final answer:

Procedure	Press	Display
(2 + 3) × (4 + 5) 5×9=45	×	45.00
	PRX (HP-19C)	45.00

```
45.00   ×
        ***
```

Notice that you didn't need to write down or key in the intermediate answers from inside the parentheses before you multiplied—the calculator automatically stacked up the intermediate results for you and brought them out on a last-in, first-out basis when it was time to multiply.

No matter how complicated a problem may look, it can always be reduced to a series of one- and two-number operations. Just work through the problem in the same logical order you would use if you were working it with a pencil and paper.

For example, to solve:

$$\frac{(9 + 8) \times (7 + 2)}{(4 \times 5)}$$

Press	Display	
9 ENTER↑ 8 +	17.00	Intermediate result of (9 + 8).
7 ENTER↑ 2 +	9.00	Intermediate result of (7 + 2).
×	153.00	(9 + 8) multiplied by (7 + 2).
4 ENTER↑ 5 ×	20.00	Intermediate result of (4 × 5).
÷	7.65	The final answer.
PRX (HP-19C)	7.65	

```
9.00 ENT↑
8.00   +
7.00 ENT↑
2.00   +
         ×
4.00 ENT↑
5.00   ×
         ÷
7.65   ***
```

Now try these problems. Remember to work through them as you would with a pencil and paper, but don't worry about intermediate answers—they're handled automatically by the calculator.

$(2 \times 3) + (4 \times 5) = 26.00$

$$\frac{(14 + 12) \times (18 - 12)}{(9 - 7)} = 78.00$$

$$\frac{\sqrt{16.38 \times 5}}{.05} = 181.00$$

$4 \times (17 - 12) \div (10 - 5) = 4.00$

$\sqrt{(2 + 3) \times (4 + 5)} + \sqrt{(6 + 7) \times (8 + 9)} = 21.57$

A Word about the HP-19C/HP-29C

Now that you've learned how to use the calculator, you can begin to fully appreciate the benefits of the Hewlett-Packard logic system. With this system, you enter numbers using a parenthesis-free, unambiguous method called RPN.

It is this unique system that gives you all these calculating advantages whether you're writing keystrokes for a program or using the calculator under manual control:

- You never have to work with more than one function at a time. The calculator cuts problems down to size instead of making them more complex.
- Pressing a function key immediately executes the function. You work naturally through complicated problems, with fewer keystrokes and less time spent.
- Intermediate results appear as they are calculated. There are no "hidden" calculations, and you can check each step as you go.
- Intermediate results are automatically handled. Using the HP-19C, you don't even have to print out long intermediate answers when you work a problem. (Of course, if you want intermediate answers, the HP-19C printer will record them in TRACE mode.)
- Intermediate answers are automatically inserted into the problem on a last-in, first-out basis. You don't have to remember where they are and then summon them.
- You can calculate in the same order that you do with pencil and paper. You don't have to think the problem through ahead of time.

The HP system takes a few minutes to learn. But you'll be amply rewarded by the ease with which your calculator solves the longest most complex equations. With HP, the investment of a few moments of learning yields a lifetime of mathematical dividends.

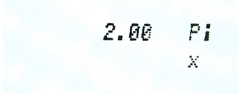
Section 2

Printer and Display Control

In the HP-19C/HP-29C, you can select many different options for display and printing of numbers. But regardless of the display options in effect, the calculator always operates internally using each number as a 10-digit mantissa and a two-digit exponent of 10. Thus, when the calculator is set to display only two digits past the decimal point, the fixed constant pi, which is represented internally as $3.141592654 \times 10^{00}$, will appear in the display as 3.14.

For example, when you compute $2 \times \pi$, you might *see* the answer to only two decimal places:

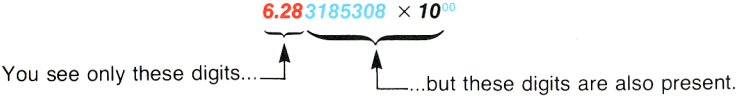
Press	Display
2 g π ×	6.28



However, inside the calculator all numbers have 10-digit mantissas and two-digit exponents of 10. So the calculator *actually* calculates using full 10-digit numbers:

$$2.000000000 \times 10^{00} \; g \; \pi \; 3.141592654 \times 10^{00} \; \times$$

yields an answer that is actually carried to full 10 digits internally:



The Continuous Memory of the calculator maintains values that are in the displayed X-register. Any number that was in the display before you turn the calculator off will return to the display when you turn the calculator back on.

Display Control Keys

There are three functions, FIX, SCI, and ENG, that allow you to control the manner in which numbers appear in the calculator display.

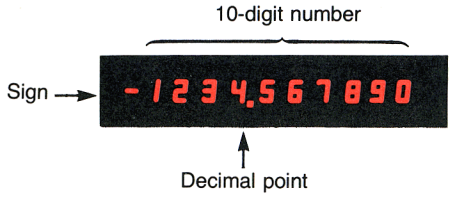
FIX displays and prints numbers in fixed decimal point format, while SCI permits you to view numbers in a scientific notation format. ENG displays numbers in engineering notation, with exponents of 10 shown in multiples of three (e.g., 10^{3}, 10^{-6}, 10^{12}). By pressing a digit key (0 through 9) after any of these display control functions, you specify the number of digits displayed.

No matter which format or how many displayed digits you choose, display control alters only the manner in which a number is displayed and printed. The actual number itself is not altered by any of the print options or the display control keys.

Display mode is one of the items that is maintained by the Continuous Memory of the HP-19C/HP-29C, so even though you may turn the calculator off, when you turn it back on again

you will see numbers displayed in the manner you selected earlier. Most of the examples in this handbook use two digits past the decimal point, but as you will soon see, you can work problems with the display set to any mode you desire.

Fixed Point Display



Using fixed point display, you can specify the number of places to be shown after the decimal point. It is selected by pressing [FIX] followed by a number key to specify the number of decimal places (0 through 9) to which the display is to be rounded. The displayed number begins at the left side of the display (or the right side of the printed tape on the HP-19C) and includes trailing zeros within the setting selected. Note that when you turn your calculator OFF, then back ON (RUN on the HP-19C), the calculator remains in the same display number mode and the previous value in the displayed X-register is retained.

For example:

Slide the HP-19C Print Mode switch MAN TRACE NORM to MAN now so that you can concentrate on the display changes.

Press	Display	
f [FIX] 2	*6.28*	[FIX] 2 display mode used in this handbook. Result remains from previous example.
123.4567	*123.4567*	
f [FIX] 0	*123.*	Display is rounded off to 0 decimal places. Internally, however, the number maintains its original value of $123.4567\ 000 \times 10^{00}$.
f [FIX] 4	*123.4567*	
(Turn the calculator off and back on)		
	123.4567	Calculator remains in the same display mode; number in X-register is retained.
f [FIX] 5	*123.45670*	
f [FIX] 1	*123.5*	Notice that the display rounds if the first *hidden* digit is 5 or greater.
f [FIX] 2	*123.46*	Normal [FIX] 2 display.

Scientific Notation Display

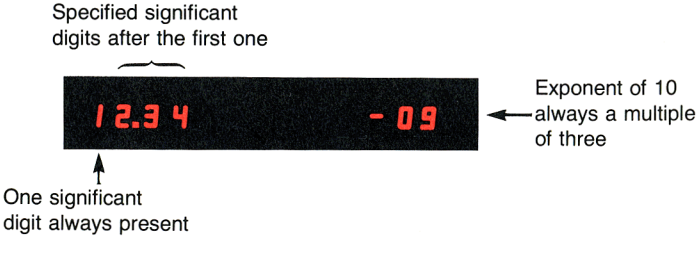
In scientific notation each number is displayed with a single digit to the left of the decimal point followed by a specified number of digits (up to seven) to the right of the decimal point and multiplied by a power of 10. Scientific notation is particularly useful when working with very large or small numbers.

Scientific notation is selected by pressing f SCI followed by a digit key to specify the number of decimal places to which the number is rounded. The display is left-justified and includes trailing zeros within the selected setting. The HP-19C printed copy is right-justified, with a sign to identify the exponent of 10.

Press	Display		
123.4567	*123.4567*		
f SCI 2	*1.23*	*02*	Indicates 1.23×10^2.
f SCI 4	*1.2346*	*02*	Indicates 1.2346×10^2. Notice that the display rounds if the first *hidden* mantissa digit is 5 or greater.
f SCI 7	*1.2345670*	*02*	Indicates 1.2345670×10^2.

Note: You can easily key in numbers in scientific notation format by using the EEX *(enter exponent)* key—more about this later.

Engineering Notation Display



Engineering notation allows all numbers to be shown with exponents of 10 that are multiples of three (e.g., 10^{3}, 10^{-6}, 10^{12}).

This is particularly useful in scientific and engineering calculations, where units of measure are often specified in multiples of three. Refer to the prefix chart below.

Multiplier	Prefix	Symbol
10^{12}	tera	T
10^{9}	giga	G
10^{6}	mega	M
10^{3}	kilo	k
10^{-3}	milli	m
10^{-6}	micro	μ
10^{-9}	nano	n
10^{-12}	pico	p
10^{-15}	femto	f
10^{-18}	atto	a

Engineering notation is selected by pressing f ENG followed by a number key. The first significant digit is always present in the display, and the number key specifies the number of additional significant digits to which the display is rounded. The decimal point always appears in the display. For example:

Press	Display		
.0123456	*0.0123456*		
f ENG 1	*12.*	*-03*	Engineering notation display. Number appears in the display rounded off to one significant digit after the omnipresent first one. Power of 10 is proper multiple of three.
f ENG 3	*12.35*	*-03*	Display is rounded off to third significant digit after the first one.
f ENG 7	*12.345600*	*-03*	
f ENG 0	*10.*	*-03*	Display rounded off to first significant digit.

Notice that rounding can occur to the *left* of the decimal point, as in the case of ENG 0 specified above.

When engineering notation has been selected, the decimal point shifts to show the mantissa as units, tens, or hundreds in order to maintain the exponent of 10 as a multiple of three. For example, multiplying the number now in the calculator by 10 causes the decimal point to shift to the right without altering the exponent of 10:

Press	Display	
f ENG 2	*12.3*	*-03*
10 ×	*123.*	*-03*

However, multiplying again by 10 causes the exponent to shift to another multiple of three. Since you specified ENG 2 earlier, the calculator maintains two significant digits after the first one when you multiply by 10.

Press	Display		
10 ×	*1.23*	*00*	Decimal point shifts. Power of 10 shifts to 10^0. Display maintains two significant digits after the first one.

Format of Printed Numbers (HP-19C)

When using the printer, whether you are in MAN or NORM mode (where you must press PRX to see answers) or in TRACE (where the HP-19C automatically prints answers as they are calculated), printed numbers can be shown in any display format—fixed point, scientific notation, or engineering notation. By selecting the display format, you also select the print format.

Results from your HP-19C are always displayed and printed in the format that you have chosen. The three-asterisk label that you see printed next to a result is a guarantee that it is in the chosen display format. Although numbers in the display are left-justified, printed numbers are right-justified.

Numbers that you key in—this is, numbers that are *not* the results of operations—are also printed by the HP-19C. When you key in a number with the Print Mode switch set to NORM or TRACE, the HP-19C does not print it until you change display format or press a function key. Then the number is printed exactly *as you keyed it in*. (One case is an exception to this rule—more about that later.) A number that you keyed in is not the result of an operation, and no asterisks are printed to its right. Subsequent *results*, of course, are printed in the selected format with a three-asterisk label. For example:

Slide the Print Mode switch MAN TRACE NORM to NORM.

Press	Display		
.0012345	*0.0012345*		
f SCI 3	*1.235*	*–03*	When you press any function, the number is first printed just as you keyed it in.
PRX	*1.235*	*–03*	Results of functions, including display formatting, are printed in the selected format.
123456789	*123456789.*		
f ENG 6	*123.4568*	*06*	The number is printed as you keyed it in.
PRX	*123.4568*	*06*	The three-asterisk label guarantees that the number is now in the selected format.

```
0.0012345 SCI3
1.235-03  ***

123456789.0 ENG6
123.4568+06 ***
```

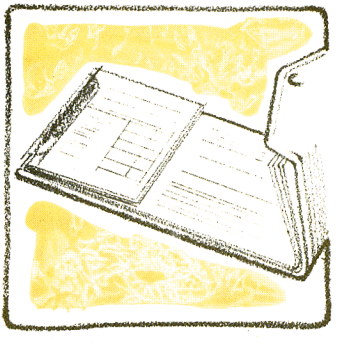
Notice that the HP-19C *prints* a + (plus) sign to show you positive exponents of 10.

Thus, whenever you key in a number, the HP-19C prints it just as you keyed it in; *then* the format is changed. It is easy for you to reconstruct your calculation because your exact inputs are identifiable from your printed copy.

When you have keyed in a number, there is one time that the HP-19C will change its format *before* printing. If you have specified fixed point notation the HP-19C will attempt to align the decimal points for easy readability on your printed copy. It will do this in fixed point notation by printing the number that you keyed in in the *specified* format (if the number can be printed without truncating), adding trailing zeros and exponent if necessary.

This feature permits you to key in numbers in fixed point notation and line up the decimal points in the printed record of your calculations.

Example: You begin the month with a balance of $735.43 in your checking account. During the month, you write checks for $235, $79.95, $5, $1.44, $17.83, $50, and $12.43. Calculate the closing balance for the account and preserve a printed record of your calculations.



First, ensure that the HP-19C Print Mode switch MAN TRACE NORM is set to NORM.

Press	Display	
f FIX 2	*123.45*	Sets FIX 2 display mode. (Result remains from previous problem.)
735.43 ENTER↑	*735.43*	
235 −	*500.43*	Two extra zeros printed so that decimal points will line up.
79.95 −	*420.48*	The number is printed exactly as you keyed it in.
5 −	*415.48*	Two extra zeros printed.
1.44 −	*414.04*	
17.83 −	*396.21*	
50 −	*346.21*	Two extra zeros printed.
12.43 −	*333.78*	
PRX	*333.78*	Closing balance.

```
                FIX2
735.43 ENT↑
235.00   -
 79.95   -
  5.00   -
  1.44   -
 17.83   -
 50.00   -
 12.43   -
333.78  ***
```

You need not worry about "losing" digits on the printed copy. The HP-19C printer will never truncate digits (not even extra zeros) that you have keyed in. For example, if you wanted to set aside 5/10000 of the closing balance of your account for a present for your sister-in-law:

Press	Display	
.0005	*0.0005*	
×	*0.17*	Entire number is printed—not rounded to FIX 2.
PRX	*0.17*	Amount set aside for sister-in-law's gift. Result of function is rounded to FIX 2.

```
0.0005   ×
  0.17  ***
```

Automatic Display Switching

The HP-19C/HP-29C switches the display from fixed point notation to scientific notation SCI with the same number of decimal places as previously set by FIX whenever the number is too large or too small to be seen with a fixed decimal point. This feature keeps you from missing unexpectedly large or small answers. For example, if you try to solve $(.05)^3$ in normal FIX 2 display, the answer is automatically shown in scientific notation.

Press	Display		
CLx	0.00		Normal FIX 2 display.
.05 ENTER↑	0.05		
3 f y^x	1.25	-04	Display automatically switched to SCI to show answer.
PRx	1.25	-04	

```
                CLX
0.05 ENT↑
3.00            Yˣ
1.25-04         ***
```

After automatically switching from fixed point to scientific display, when a new number is keyed in or CLx is pressed, the display automatically reverts back to the fixed point display originally selected.

The calculator also switches to scientific notation if the answer is too large ($\geq 10^{10}$) for fixed point display. For example, the display will not switch from fixed point if you solve 1582000 × 1842:

Press	Display		
1582000	1582000.		
ENTER↑	1582000.00		
1842 ×	2914044000.		Fixed point format.
PRx	2914044000.		

```
1582000.00 ENT↑
1842.00         ×
2914044000.     ***
```

However, if you multiply the result by 10, the answer is too large for fixed point notation, and the calculator display switches automatically to scientific notation:

Press	Display		
10 ×	2.91	10	Scientific notation format.
PRx	2.91	10	

```
10.00           ×
2.91+10         ***
```

Notice that automatic switching is between fixed and scientific notation display modes only—engineering notation display must be selected from the keyboard.

Keying in Exponents of Ten

You can key in numbers multiplied by powers of 10 by pressing EEX *(enter exponent of 10)* followed by number keys to specify the exponent of 10. For example, to key in 15.6 trillion (15.6×10^{12}), and multiply it by 25:

Press	Display		
15.6	15.6		
EEX	15.6	00	
12	15.6	12	This means 15.6×10^{12}.

Now Press	Display	
ENTER↑	1.56	13
25 ×	3.90	14
PRX	3.90	14

```
15.6+12 ENT↑
   25.00   ×
3.90+14  ***
```

You can save time when keying in exact powers of 10 by merely pressing EEX and then pressing the desired power of 10. For example, key in 1 million (10^6) and divide by 52.

Press	Display		
EEX	1.	00	You do not have to key in the number 1 before pressing EEX when the number is an exact power of 10.
6	1.	06	
ENTER↑	1000000.00		Since you have not specified scientific notation, the display reverts to fixed point notation when you press ENTER↑.
52 ÷	19230.77		
PRX	19230.77		

```
1.+06 ENT↑
   52.00   ÷
19230.77  ***
```

To see your answer in scientific notation with six decimal places:

Press	Display	
f SCI 6	1.923077	04
PRX	1.923077	04

```
          SCI6
1.923077+04  ***
```

To key in negative exponents of 10, key in the number, press EEX, press CHS to make the exponent negative, then key in the power of 10. For example, key in Planck's constant (h)—roughly, 6.625×10^{-27} erg sec.—and multiply it by 50.

Press	Display		
CL x	0.000000	00	
f FIX 2	0.00		
6.625 EEX	6.625	00	
CHS	6.625	-00	
27	6.625	-27	
ENTER↑	6.63	-27	
50 ×	3.31	-25	Erg sec.
PRX	3.31	-25	

```
          CLX
          FIX2
6.625-27 ENT↑
   50.00   ×
 3.31-25  ***
```

Calculator Overflow and Underflow

When the number in the calculator would be greater than $9.999999999 \times 10^{99}$, the calculator displays all 9's to indicate that the problem has exceeded the calculator's range. For example, if you solve $(1 \times 10^{49}) \times (1 \times 10^{50})$, the calculator will display the answer:

Press	Display		
CLx	0.00		
EEX 49 ENTER↑	1.00	49	
EEX 50 ×	1.00	99	
PRX	1.00	99	

```
                CLX
1.+49 ENT↑
1.+50           ×
1.00+99         ***
```

But if you attempt to multiply the above result by 100, the calculator display indicates overflow by showing you all 9's:

Press	Display		
100 ×	9.9999999	99	Overflow indication.
PRX	9.9999999	99	

```
100.00          ×
9.9999999+99    ***
```

Numbers 10^{-100} and smaller are too small for the calculator to display or hold internally. When a number 10^{-100} or smaller is calculated the HP-19C/HP-29C substitutes a zero for the result.

Error Display

If you happen to key in an improper operation the word *Error* will appear in the display.

In addition, if the HP-19C Print Mode switch MAN NORM TRACE is set to NORM or TRACE, the printer will print ERROR

For example, if you attempt to calculate the square root of –4, the calculator will recognize it as an improper operation:

Ensure that the HP-19C Print Mode switch MAN NORM TRACE is set to NORM.

Press	Display
4 CHS	–4.
f √x	Error

```
-4.00           √x
                ERROR
```

Pressing *any key* clears the error and is *not* executed. The number that was in the display before the error-causing function is returned to the display so that you can see it. Sliding the HP-19C OFF-PRGM-RUN switch or the HP-29C PRGM-RUN switch to PRGM also clears the error.

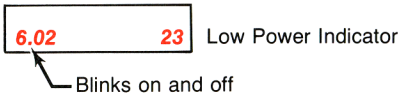
When the switch is then returned to the RUN position, the number that was in the display before the error-causing function is again returned there. The rest of the calculator remains unchanged. To clear the error:

Press	Display
CL x	-4.00

All those operations that cause an error condition are listed in appendix B.

Low Power Display

When you are operating from battery power in RUN mode, the decimal point blinks on and off to warn you that you have a minimum of one minute of operating time left.



In PRGM mode, a blinking decimal point will appear between the step number and the keycode. You must then either operate the calculator from the battery charger/ac adapter as described under AC Line Operation, or you can substitute a fully charged battery pack for the one in the calculator.

Section 3

The Automatic Memory Stack

The Stack

Automatic storage of intermediate results is the reason that the HP-19C/HP-29C slides so easily through the most complex equations. And automatic storage is made possible by the Hewlett-Packard automatic memory stack.

Display

The display format used in this section is obtained by pressing f FIX 2. You can work through this section with the HP-19C Print Mode switch at any setting you desire. However, the printed tapes that illustrate the examples in this section were created with the HP-19C Print Mode switch MAN TRACE NORM set to NORM.

When you see decimal digits like *0.00* in the display, the number represents the contents of the "X-register" in the calculator.

Basically, numbers are stored and manipulated in the machine "registers." Each number, no matter how few digits (e.g., 0, 1, or 5) or how many (e.g., 3.14159265, –23.28362, or 2.8714890×10^{27}), occupies one entire register.

The displayed X-register, which is the only visible register, is one of four registers inside the calculator that are positioned to form the automatic memory stack. We label these registers X, Y, Z, and T. They are "stacked" one on top of the other with the displayed X-register on the bottom. When the calculator is switched on, the Y, Z, and T registers are cleared to 0.00. The X-register is maintained by the Continuous Memory.

Switch the calculator OFF, then ON (RUN on the HP-19C).

Press	Display
CL x	*0.00*

```
CLX
```

Name	Register	
T	*0.00*	
Z	*0.00*	
Y	*0.00*	
X	*0.00*	Always displayed.

You can view the contents of the entire stack at any time on the HP-19C by printing them using the PRT STK *(print stack)* key.

Press	Display
f PRT STK	*0.00*

```
PRST
0.00   T
0.00   Z
0.00   Y
0.00   X
```

Notice that f PRTSTK, like PRX and the other print functions, operates regardless of the position of the Print Mode switch.

Manipulating Stack Contents

The R↓ *(roll down)* and x≷y *(x exchange y)* keys allow you to review the stack contents or to shift data within the stack for computation at any time.

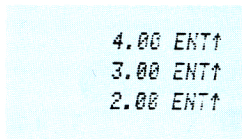
Reviewing the Stack

To see how the R↓ key works, first load the stack with numbers 1 through 4 by pressing:

4 ENTER↑ 3 ENTER↑ 2 ENTER↑ 1

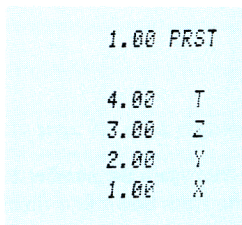
The numbers that you keyed in are now loaded into the stack, and its contents look like this:

T	*4.00*	
Z	*3.00*	
Y	*2.00*	
X	*1.*	Display



To print the contents of the stack now:

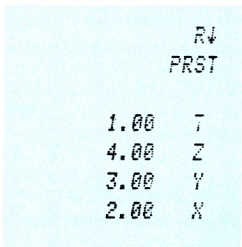
Press	Stack Contents
	T *4.00*
	Z *3.00*
f PRTSTK	Y *2.00*
	X *1.00*



When you press the R↓ key, the stack contents shift downward one register. So the last number that you have keyed in will be rotated around to the T-register when you press R↓ When you press R↓ again, the stack contents again roll downward one register.

To see how the R↓ key operates, press f PRTSTK on the HP-19C to list the stack contents after each press of the R↓ key:

Press	Stack Contents	
	T *1.00*	
R↓	Z *4.00*	
f PRTSTK	Y *3.00*	
	X *2.00*	Display.



Press	Stack Contents	
	T 2.00	
	Z 1.00	
R↓	Y 4.00	
f PRT STK	X 3.00	Display.
R↓	T 3.00	
f PRT STK	Z 2.00	
	Y 1.00	
	X 4.00	Display.
R↓	T 4.00	
f PRT STK	Z 3.00	
	Y 2.00	
	X 1.00	Display.

```
        R↓
      PRST

2.00    T
1.00    Z
4.00    Y
3.00    X

        R↓
      PRST

3.00    T
2.00    Z
1.00    Y
4.00    X

        R↓
      PRST

4.00    T
3.00    Z
2.00    Y
1.00    X
```

Once again the number 1.00 is in the displayed X-register. Four presses of the R↓ key roll the stack down four times, returning the contents of the stack to their original registers.

Exchanging x and y

The x≷y *(x exchange y)* key exchanges the contents of the X- and the Y-registers without affecting the Z- and T-registers. If you press x≷y with data intact from the previous example, the numbers in the X- and Y-registers will be changed...

... from this ...		... to this.	
T 4.00		T 4.00	
Z 3.00		Z 3.00	
Y 2.00		Y 1.00	
X 1.00	Display	X 2.00	Display.

You can verify this on the HP-19C by first listing the stack contents and then prsssing x≷y. To see the results, list the stack contents again:

Press	Stack Contents		
f PRT STK	T 4.00 Z 3.00 Y 2.00 X 1.00	Display.	PRST 4.00 T 3.00 Z 2.00 Y 1.00 X
x≷y			X≷Y PRST
f PRT STK	T 4.00 Z 3.00 Y 1.00 X 2.00	Display.	4.00 T 3.00 Z 1.00 Y 2.00 X

Notice that whenever you move numbers in the stack using one of the data manipulation keys, the actual stack registers maintain their positions. Only the *contents* of the registers are shifted. The contents of the X-register are always displayed.

Clearing the X-Register

When you press CL x *(clear x)*, the displayed X-register is cleared to zero. No other register is affected when you press CL x.

Press CL x now, and the stack contents are changed...

... from this ...		... to this.	
T 4.00 Z 3.00 Y 1.00 X 2.00	Display.	T 4.00 Z 3.00 Y 1.00 X 0.00	CLX

Although it may be comforting, *it is never necessary to clear the displayed X-register when starting a new calculation.* This will become obvious when you see how old results in the stack are automatically lifted by new entries.

The ENTER↑ Key

When you key a number into the calculator, its contents are written into the displayed X-register. For example, if you key in the number 314.32 now, you can see that the display contents are altered.

When you key in 314.32 with the stack contents intact from previous examples the contents of the stack registers are changed...

... from this ...		... to this.	
T 4.00		T 4.00	
Z 3.00		Z 3.00	
Y 1.00		Y 1.00	
X 0.00	Display.	X 314.32	Display.

In order to key in another number at this point, you must first terminate digit entry—i.e., you must indicate to the calculator that you have completed keying in the first number and that any new digits you key in are part of a new number.

Use the ENTER↑ key to separate the digits of the first number from the digits of the second.

When you press the ENTER↑ key, the contents of the stack registers are changed...

... from this ...		... to this.	
T 4.00		T 3.00	
Z 3.00		Z 1.00	
Y 1.00		Y 314.32	
X 314.32	Display.	X 314.32	Display.

As you can see, the number in the displayed X-register is copied into Y. The numbers in Y and Z have also been transferred to Z and T, respectively, and the number in T has been lost off the top of the stack.

Immediately after pressing ENTER↑, the X-register is prepared for a new number, and that new number writes over the number in X. For example, key in the number 543.28 and the contents of the stack registers change...

... from this ...		... to this.	
T 3.00		T 3.00	
Z 1.00		Z 1.00	
Y 314.32		Y 314.32	
X 314.32	Display.	X 543.28	Display.

CL x replaces any number in the display with zero. Any new number then writes over the zero in X.

For example, if you had meant to key in 689.4 instead of 543.28, you would press CLx now to change the stack...

... from this ...		... to this.	
T	3.00	T 3.00	
Z	1.00	Z 1.00	
Y	314.32	Y 314.32	
X	543.28 Display.	X 0.00	Display.

and then key in 689.4 to change the stack...

... from this ...		... to this.	
T	3.00	T 3.00	
Z	1.00	Z 1.00	
Y	314.32	Y 314.32	
X	0.00 Display.	X 689.4	Display.

Notice that numbers in the stack do not move when a new number is keyed in immediately after you press ENTER↑, CLx, or Σ+. However, numbers in the stack *do* lift upward when a new number is keyed in immediately after you press most other functions, including R↓ and x≷y. See appendix C, Stack Lift and LAST X, for a complete list of the operations that cause the stack to lift.

One-Number Functions and the Stack

One-number functions execute upon the number in the X-register only, and the contents of the Y-, Z-, and T-registers are unaffected when a one-number function key is pressed.

For example, with numbers positioned in the stack as in the previous example, pressing f √x changes the stack contents...

... from this ...		... to this.	
T	3.00	T 3.00	
Z	1.00	Z 1.00	
Y	314.32	Y 314.32	
X	689.4 Display.	X 26.26	Display.

The one-number function executes upon only the number in the displayed X-register, and the answer writes over the number that was in the X-register. No other stack register is affected by a one-number function.

Two-Number Functions and the Stack

Hewlett-Packard calculators do arithmetic by positioning the numbers in the stack the same way you would on paper. For instance, if you wanted to add 34 and 21 you would write 34 on a piece of paper and then write 21 underneath it, like this:

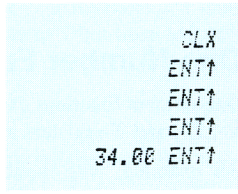
```
 34
 21
----
```

and then you would add, like this:

```
 34
+21
----
 55
```

Numbers are positioned the same way in the calculator. Here's how it is done. (As you know, it is not necessary to remove earlier results from the stack before beginning a new calculation, but for clarity, the following example is shown with the stack cleared to all zeros initially. If you want the contents of your stack registers to match the ones here, first clear the stack by using the CLx and ENTER↑ keys to fill the stack with zeros.)

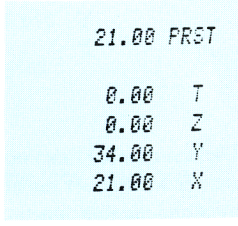
Press	Display	
CLx	*0.00*	
ENTER↑	*0.00*	
ENTER↑	*0.00*	
ENTER↑	*0.00*	Stack cleared to zeros initially.
34	*34.*	34 is keyed into X.
ENTER↑	*34.00*	34 is copied into Y.
21	*21.*	21 writes over the 34 in X.



Now 34 and 21 are sitting vertically in the stack so we can add.

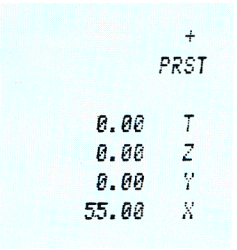
To see this on the HP-19C,

Press	Display
f PRTSTK	*21.00*



Now add:

+	*55.00*
f PRTSTK	*55.00*



The simple old-fashioned math notation helps explain how to use your calculator. Both numbers are always positioned in the stack in the natural order first, then the operation is executed when the function key is pressed. *There are no exceptions to this rule.* Subtraction, multiplication, and division work the same way. In each case, the data must be in the proper position before the operation can be performed.

Chain Arithmetic

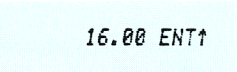
You've already learned how to key numbers into the calculator and perform calculations with them. In each case you first needed to position the numbers in the stack manually using the ENTER↑ key. However, the stack also performs many movements automatically. These automatic movements add to its computing efficiency and ease of use, and it is these movements that automatically store intermediate results. The stack automatically "lifts" every calculated number in the stack when a new number is keyed in because it knows that after it completes a calculation, any new digits you key in are a part of a new number. Also, the stack automatically "drops" when you perform a two-number operation.

To see how it works, let's solve 16 + 30 + 11 + 17 = ?

If you want the contents of your stack registers to match those shown here, first clear the stack by using the CL x and ENTER↑ keys to fill the stack with zeros.

Remember, too, that you can always monitor the contents of the stack at any time by using the f PRTSTK function on the HP-19C, or R↓ on both the HP-19C or HP-29C. However, using R↓ or x≷y to monitor the contents of the stack immediately following the ENTER↑ or CL x keys may cause an erroneous result.

Press	Stack Contents	
16	T *0.00* Z *0.00* Y *0.00* X *16.*	16 is keyed into the displayed X-register.
ENTER↑	T *0.00* Z *0.00* Y *16.00* X *16.00*	16 is copied into Y.

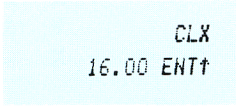


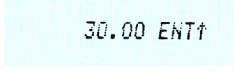
30	T 0.00 Z 0.00 Y 16.00 X 30.	30 writes over the 16 in X.	
⊞	T 0.00 Z 0.00 Y 0.00 X 46.00	16 and 30 are added together. The answer, 46, is displayed.	30.00 +
11	T 0.00 Z 0.00 Y 46.00 X 11.	11 is keyed into the displayed X-register. The 46 in the stack is automatically raised.	
⊞	T 0.00 Z 0.00 Y 0.00 X 57.00	46 and 11 are added together. The answer, 57, is displayed.	11.00 +
17	T 0.00 Z 0.00 Y 57.00 X 17.	17 is keyed into the X-register, 57 is automatically entered into Y.	
⊞ PRX	T 0.00 Z 0.00 Y 0.00 X 74.00	57 and 17 are added together for the final answer.	17.00 + 74.00 ***

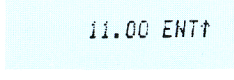
After any calculation or number manipulation, the stack automatically lifts when a new number is keyed in. Because operations are performed when the operations are pressed, the length of such chain problems is unlimited unless a number in one of the stack registers exceeds the range of the calculator (up to $9.999999999 \times 10^{99}$).

In addition to the automatic stack lift after a calculation, the stack automatically drops during calculations involving both the X- and Y-registers. It happened in the above example, but let's do the problem differently to see this feature more clearly. For clarity, first press CL X to clear the X-register. Now, again solve 16 + 30 + 11 + 17 =?

Press	Stack Contents	
16	T 0.00 Z 0.00 Y 0.00 X 16.	16 is keyed into the displayed X-register.
ENTER↑	T 0.00 Z 0.00 Y 16.00 X 16.00	16 is copied into Y.
30	T 0.00 Z 0.00 Y 16.00 X 30.	30 is written over the 16 in X.
ENTER↑	T 0.00 Z 16.00 Y 30.00 X 30.00	30 is entered into Y. 16 is lifted up to Z.
11	T 0.00 Z 16.00 Y 30.00 X 11.	11 is keyed into the displayed X-register.
ENTER↑	T 16.00 Z 30.00 Y 11.00 X 11.00	11 is copied into Y. 16 and 30 are lifted up to T and Z respectively.
17	T 16.00 Z 30.00 Y 11.00 X 17.	17 is written over the 11 in X.







⊞	T 16.00 Z 16.00 Y 30.00 X 28.00	17 and 11 are added together and the rest of the stack drops. 16 drops to Z and is also duplicated in T. 30 and 28 are ready to be added.	17.00 +
⊞	T 16.00 Z 16.00 Y 16.00 X 58.00	30 and 28 are added together and the stack drops again. Now 16 and 58 are ready to be added.	+
⊞	T 16.00 Z 16.00 Y 16.00 X 74.00	16 and 58 are added together for the final answer and the stack continues to drop.	+
PRX			74.00 ***

The same dropping action also occurs with ⊟, ⊠ and ⊡. The number in T is duplicated in T and drops to Z, the number in Z drops to Y, and the numbers in Y and X combine to give the answer, which is visible in the X-register.

This automatic lift and drop of the stack give you tremedous computing power, since you can retain and position intermediate results in long calculations without the necessity of reentering the numbers.

Order of Execution

When you see a problem like this one:

$$5 \times \left[(3 \div 4) - (5 \div 2) + (4 \times 3)\right] \div (3 \times .213)$$

you must decide where to begin before you ever press a key.

Experienced HP calculator users have determined that by starting every problem at its innermost number or parentheses and working outward, just as you would with paper and pencil, you maximize the efficiency and power of your HP calculator. Of course, with the HP-19C/HP-29C you have tremendous versatility in the order of execution.

For example, you could work the problem above by beginning at the left side of the equation and simply working through it in left-to-right order. All problems cannot be solved using left-to-right order, however, and the best order for solving any problem is to begin with the innermost parentheses and work outward. So, to solve the problem above:

Press	Display	
3	3.	
ENTER↑	3.00	
4	4.	
÷	0.75	Intermediate answer for (3 ÷ 4).
5	5.	
ENTER↑	5.00	
2	2.	
÷	2.50	Intermediate answer for (5 ÷ 2).
−	−1.75	Intermediate answer for (3 ÷ 4) − (5 ÷ 2).
4	4.	
ENTER↑	4.00	
3	3.	
×	12.00	Intermediate answer for (4 × 3).
+	10.25	Intermediate answer for (3 ÷ 4) − (5 ÷ 2) + (4 × 3).
3	3.	
ENTER↑	3.00	
.213	.213	
×	0.64	Intermediate answer for (3 × .213).
÷	16.04	
5	5.	The first number is keyed in.
×	80.20	The final answer.
PRX	80.20	

```
3.00 ENT↑
4.00   ÷
5.00 ENT↑
2.00   ÷
         −
4.00 ENT↑
3.00   ×
         +
3.00 ENT↑
0.213  ×
         ÷
5.00   ×
80.20  ***
```

LAST X

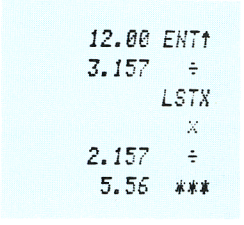
In addition to the four stack registers that automatically store intermediate results, the calculator also contains a separate automatic register, the LAST X register. This register preserves the value that was last displayed in the X-register before the performance of a function. To place the contents of the LAST X register into the display again, press f LAST X.

Recovering from Mistakes

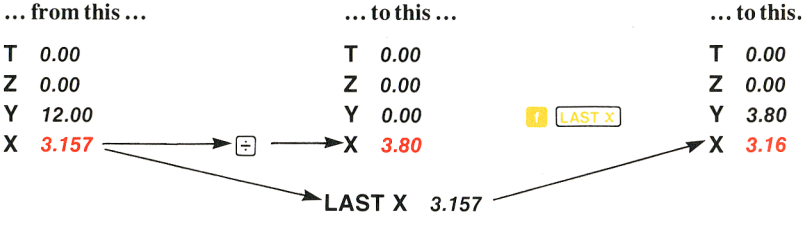
[LAST X] makes it easy to recover from keystroke mistakes, such as pressing the wrong function key or keying in the wrong number.

Example: Divide 12 by 2.157 after you have mistakenly divided by 3.157.

Press	Display	
12	*12.*	
[ENTER↑]	*12.00*	
3.157 [÷]	*3.80*	Oops! You made a mistake.
[f] [LAST X]	*3.16*	Retrieves that last entry (3.157).
[×]	*12.00*	You're back at the beginning.
2.157 [÷]	*5.56*	The correct answer.
[PRX]	*5.56*	



In the above example, when the first [÷] is pressed, followed by [f] [LAST X], the contents of the stack and LAST X registers are changed...



This makes possible the correction illustrated in the example above.

Recovering a Number for Calculation

The LAST X register is useful in calculations where a number occurs more than once. By recovering a number using [LAST X], you do not have to key that number into the calculator again.

Example: Calculate

$$\frac{7.32 + 3.6501123}{3.6501123}$$

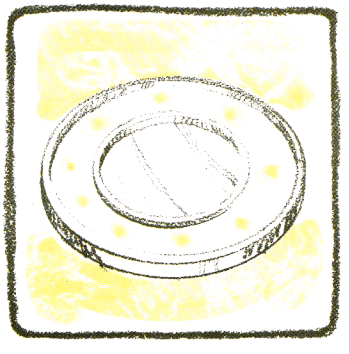
Press	Display	
7.32	*7.32*	
ENTER↑	*7.32*	
3.6501123	*3.6501123*	Intermediate answer.
+	*10.97*	Recalls 3.6501123 to X-register.
f LAST X	*3.65*	
÷	*3.01*	The answer.
PRX	*3.01*	

```
7.32 ENT↑
3.6501123  +
        LSTX
           ÷
3.01    ***
```

Constant Arithmetic

You may have noticed that whenever the stack drops because of a two-number operation (not because of R↓), the number in the T-register is reproduced there. This stack operation can be used to insert a constant into a problem.

Example: A bacteriologist tests a certain strain whose population typically increases by 15% each day. If he starts a sample culture of 1000, what will be the bacteria population at the end of each day for six consecutive days?



Method: Put the growth factor (1.15) in the Y-, Z-, and T-registers and put the original population (1000) in the X-register. Thereafter, you get the new population whenever you press ×. Try working this problem with the HP-19C Print Mode switch set to TRACE so that you'll have a record of all the answers without pressing PRX each time.

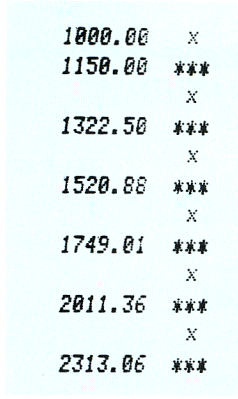
Slide the HP-19C Print Mode switch MAN ▮ NORM to TRACE.

Press	Display	
1.15	*1.15*	Growth factor.
ENTER↑	*1.15*	
ENTER↑	*1.15*	
ENTER↑	*1.15*	Growth factor now in T.

```
1.15 ENT↑
     ENT↑
     ENT↑
```

For example, to recall the number of persons carried daily by the Japanese National Railway:

1000	*1000.*	Starting population.
⊠	*1150.00*	Population after 1st day.
⊠	*1322.50*	Population after 2nd day.
⊠	*1520.88*	Population after 3rd day.
⊠	*1749.01*	Population after 4th day.
⊠	*2011.36*	Population after 5th day.
⊠	*2313.06*	Population after 6th day.



When you press ⊠ the first time, you calculate 1.15×1000. The result (1150.00) is displayed in the X-register and a new copy of the growth factor drops into the Y-register. Since a new copy of the growth factor is duplicated from the T-register each time the stack drops, you never have to reenter it.

Notice that performing a two-number operation such as ⊠ causes the number in the T-register to be duplicated there each time the stack is dropped. However, the R↓ key, since it rotates the contents of the stack registers, does not rewrite any number, but merely shifts the numbers that are already in the stack.

Section 4

Storing and Recalling Numbers

You have learned about the calculating power that exists in the four-register automatic memory stack and the LAST X register of your HP-19C/HP-29C calculator. In addition to the automatic storage of intermediate results that is provided by the stack, however, the calculator also contains 30 *addressable* data storage registers that are unaffected by operations within the stack. These registers allow you to manually store and recall constants or to set aside numbers for use in later calculations. Like all functions, you can use these storage registers either from the keyboard or as part of a program. The primary registers are part of the Continuous Memory of the calculator and maintain their contents even though the calculator is turned OFF.

The diagram below shows all storage registers. The addresses of the primary storage registers are indicated by the numbers 0 through 9 and by .0 through .5. The address of the indirect storage registers are indicated by the numbers (16) through (29).

Storing and recalling numbers in the 14 indirect storage registers is explained in section 12 (page 164).

Automatic Memory Stack	Primary Storage Registers	Indirect Storage Registers
T	R_0	$R_{(16)}$
Z	R_1	$R_{(17)}$
Y	R_2	$R_{(18)}$
X	R_3	$R_{(19)}$
LAST X	R_4	$R_{(20)}$
	R_5	$R_{(21)}$
	R_6	$R_{(22)}$
	R_7	$R_{(23)}$
	R_8	$R_{(24)}$
	R_9	$R_{(25)}$
	$R_{.0}$	$R_{(26)}$
	$R_{.1}$	$R_{(27)}$
	$R_{.2}$	$R_{(28)}$
	$R_{.3}$	$R_{(29)}$
	$R_{.4}$	
	$R_{.5}$	

Primary Storage Registers

Storing Numbers

To store a displayed number in any of storage registers R_0 through R_9.

1. Press STO *(store)*.
2. Press the number key of the applicable register address (0 through 9).

For example, to store Avogadro's number (approximately 6.02×10^{23}) in register R_2:

Slide the HP-19C Print Mode switch to NORM MAN TRACE NORM if you want your printed tape to match the ones shown here.

Press	Display	
6.02 EEX 23	6.02	23
STO 2	6.02	23

```
6.02+23 STO2
```

Avogadro's number is now stored in register R_2. Notice that when a number is stored, it is merely copied into the storage register, so 6.02×10^{23} also remains in the displayed X-register.

To store a displayed number in any of storage registers $R_{.0}$ through $R_{.5}$.

1. Press STO .
2. Press the decimal point key ⊡.
3. Press the number key of the applicable register address (0 through 5).

For example, to store 16,495,000 (the number of persons carried daily by the Japanese National Railway) in register $R_{.4}$:

Press	Display
16495000	16495000.
STO ⊡ 4	16495000.00

```
16495000.00 ST.4
```

The number has been copied into storage register $R_{.4}$ and also remains in the displayed X-register.

Recalling Numbers

Numbers are recalled from storage registers back into the displayed X-register in much the same way as they are stored. To recall a number from any of storage registers R_0 through R_9:

1. Press RCL *(recall)*.
2. Press the number key of the applicable register address (0 through 9).

For example, to recall Avogadro's number from register R_2:

Press	Display	
RCL 2	6.02	23

```
RCL2
```

To recall a number from any of registers $R._0$ through $R._5$:

1. Press RCL.
2. Press the decimal point key ⊡.
3. Press the number key of the applicable register address (0 through 5).

For example, to recall the number of persons carried daily by the Japanese National Railway:

Press	Display
RCL ⊡ 4	*16495000.00*

RC.4

Recalling a number causes the stack to lift unless the preceding keystroke was ENTER↑, CLx, or Σ+ (more about Σ+ later).

Printing the Storage Registers (HP-19C)

You can see the contents of all storage registers at any time with the PRT REG key. Simply press f PRT REG to print a listing of the contents of all the numbered storage registers. For example, if you have worked through the examples as shown above, printing the contents of the storage registers should give you a listing like the one shown below.

Press	Display
f PRT REG	*16495000.00*

```
                PREG
        0.00     0
        0.00     1
    6.02+23      2
        0.00     3
        0.00     4
        0.00     5
        0.00     6
        0.00     7
        0.00     8
        0.00     9
        0.00    .0
        0.00    .1
        0.00    .2
        0.00    .3
16495000.00     .4
        0.00    .5
        0.00    16
        0.00    17
        0.00    18
        0.00    19
        0.00    20
        0.00    21
        0.00    22
        0.00    23
        0.00    24
        0.00    25
        0.00    26
        0.00    27
        0.00    28
        0.00    29
```

If you want only a partial listing of storage registers, you can stop the printing of them at any time by pressing any key. The contents of the X-register prior to pressing PRT REG are returned to the displayed X-register.

Clearing Storage Registers

Even though you have recalled the contents of a storage register into the displayed X-register, the number also remains in the storage register. You can clear storage registers in either of two ways:

- To replace a number in a single storage register, merely store another number there. To clear a storage register, replace the number in it with zero. For example, to clear storage register R_2, press 0 STO 2.

- To clear *all* storage registers back to zero at one time, press f CLEAR REG. This clears all storage registers, while leaving the automatic memory stack unchanged.

Remember that because of the Continuous Memory of the calculator the primary storage registers *retain* their contents even though the calculator is turned OFF. When you turn the calculator back on again, you can summon and use the contents of the primary storage registers.

You can also clear storage registers $R_{.0}$ through $R_{.5}$ while leaving the remaining storage registers and the stack intact by using the CLEAR Σ function.

- Press f CLEAR Σ to clear storage registers $R_{.0}$ through $R_{.5}$ only.

Storage Register Arithmetic

Arithmetic can be performed *upon* the contents of storage registers R_0 through R_9 and $R_{.0}$ through $R_{.5}$ by pressing STO followed by the arithmetic function key followed in turn by the register address. For example:

Press	Result
STO + 1	Number in displayed X-register added to contents of storage register R_1, and sum placed into R_1; $(r_1 + x \rightarrow R_1)$.
STO − 2	Number in displayed X-register subtracted from contents of storage register R_2, and difference placed into R_2; $(r_2 - x \rightarrow R_2)$.
STO × . 3	Number in displayed X-register multiplied by contents of storage register R_3, and the product placed into $R_{.3}$; $[(r_{.3})x \rightarrow R_{.3}]$.
STO ÷ . 4	Contents of storage register R_4 divided by number in displayed X-register, and quotient placed into register $R_{.4}$; $(r_{.4} \div x \rightarrow R_{.4})$.

When storage register arithmetic operations are performed, the answer is written into the selected storage register, while the contents of the displayed X-register and the rest of the stack remain unchanged.

Here is an example of storage register arithmetic.

Example: During harvest, farmer Flem Snopes trucks tomatoes to the cannery for three days. On Monday and Tuesday he hauls loads of 25 tons, 27 tons, 19 tons, and 23 tons, for which the cannery pays him $55 per ton. On Wednesday the price rises to $57.50 per ton, and Snopes ships loads of 26 tons and 28 tons. If the cannery deducts 2% of the price on Monday and Tuesday because of blight on the tomatoes, and 3% of the price on Wednesday, what is Snopes' total net income?

Method: Keep total amount in a storage register while using the stack to add tonnages and calculate amounts of loss.

Press	Display	
25 ENTER↑	25.00	
27 +	52.00	
19 + 23 +	94.00	Total of Monday's and Tuesday's tonnage.
55 ×	5170.00	Gross amount for Monday and Tuesday.
STO 5	5170.00	Gross placed in storage register R_5.
2 g %	103.40	Deductions for Monday and Tuesday.
STO − 5	103.40	Deductions subtracted from total in storage register R_5.
26 ENTER↑	26.00	
28 +	54.00	Wednesday's tonnage.
57.50 ×	3105.00	Gross amount for Wednesday.
STO + 5	3105.00	Wednesday's gross amount added to total in storage register R_5.
3 g %	93.15	Deduction for Wednesday.
STO − 5	93.15	Wednesday deduction subtracted from total in storage register R_5.
RCL 5	8078.45	Snopes' total net income from his tomatoes.
PRX	8078.45	

```
25.00 ENT↑
27.00   +
19.00   +
23.00   +
55.00   ×
        STO5
 2.00   %
        ST-5
26.00 ENT↑
28.00   +
57.50   ×
        ST+5
 3.00   %
        ST-5
        RCL5
8078.45 ***
```

(You could also work this problem using the stack alone, but doing it as shown here illustrates how storage register arithmetic can be used to maintain and update different running totals.)

Storage Register Overflow

If you attempt a storage register arithmetic operation that would cause the magnitude of a number in any of the storage registers to exceed 9.999999999 10^{99}, the operation is not performed and the calculator display immediately indicates ***Error***. In addition, if the HP-19C Print Mode switch MAN NORM TRACE is set to NORM or TRACE, the printer will also print *ERROR*

When you then press any key, the error condition is cleared and the last value in the X-register before the error is again displayed. The storage registers all contain the values they held before the error-causing operation was attempted.

For example, if you store 7.33 × 10^{52} in register R_1 and attempt to use storage register arithmetic to multiply that value by 10^{50}, the display will show ***Error***.

Slide the HP-19C Print Mode switch MAN NORM TRACE to NORM.

Press	Display		
7.33	*7.33*		
EEX 52	*7.33*	*52*	
STO 1	*7.33*	*52*	
EEX 50	*1.*	*50*	
STO × 1	*Error*		

```
7.33+52 STO1
1.+50 ST×1
ERROR
```

To clear the error and display the contents of the X-register, press any key. The original contents of storage register R_1 are still present there.

Press	Display		
CL x	*1.00*	*50*	Contents of X-register.
RCL 1	*7.33*	*52*	Contents of storage register R_1.

```
RCL1
```

Section 5

Function Keys

The HP-19C/HP-29C has dozens of internal functions that allow you to compute answers to problems quickly and accurately. Each function operates the same way, regardless of whether you press the function key manually or the function is executed as part of a program.

In this section, each function key is explained as it is used manually, with the Program Mode switch set to RUN. To save HP-19C printing time and paper, you might wish to learn how to use the functions with the Print Mode switch set to MAN. Or you might wish to see every intermediate and final answer by setting the switch to TRACE. Except where indicated, however, all examples in this section are illustrated with the Print Mode switch set to NORM.

If you want your displays and printed copy to match the ones shown here, then:

HP-19C: Set the OFF-PRGM-RUN switch OFF / PRGM / RUN to RUN.
Set the Print Mode switch MAN / TRACE / NORM to NORM.

HP-29C: Set the PRGM-RUN switch PRGM / RUN to RUN.

Number Alteration Keys

Besides CHS, there are three keys provided for altering numbers in the calculator. These keys are ABS, INT, and FRAC, and you will find them most useful when performing operations as part of a program.

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 shift key followed by the ABS *(absolute value)* key. For example, to calculate the absolute value of –3:

Press	Display	
3 CHS	–3.	
g ABS	3.00	|–3|
PRX	3.00	

```
-3.00  ABS
 3.00  ***
```

To see the absolute value of +3:

Press	Display	
g ABS	3.00	|+3|
PRX	3.00	

```
       ABS
 3.00  ***
```

Integer Portion of a Number

To extract and display the integer portion of a number, press the f prefix key followed by

the [INT] *(integer)* key. For example, to display only the integer portion of the number 123.456:

Press	Display	
123.456	123.456	
f [INT]	123.00	Only the integer portion of the number remains.
PRX	123.00	

```
123.456  INT
123.00   ***
```

When f [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:

Press	Display	
f [LAST X]	123.46	Summons the original number back to the X-register.
g [FRAC]	0.46	Only the fractional portion of the number is displayed, rounded here to FIX 2 display.
PRX	0.46	

```
LSTX
FRC
0.46  ***
```

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.

Reciprocals

To calculate the reciprocal of a number in the displayed X-register, key in the number, then press g [1/x]. For example, to calculate the reciprocal of 25:

Press	Display
25 g [1/x]	0.04
PRX	0.04

```
25.00  1/X
0.04   ***
```

You can also calculate the reciprocal of a value in a previous calculation without reentering the number.

Example: In an electrical circuit, four resistors are connected in parallel. Their values are 220 ohms, 560 ohms, 1.2 kilohms, and 5 kilohms. What is the total resistance of the circuit?

$$R_T = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}} = \frac{1}{\frac{1}{220} + \frac{1}{560} + \frac{1}{1200} + \frac{1}{5000}}$$

Press	Display	
220 g 1/x	4.55	-03
560 g 1/x	1.79	-03
+	0.01	
1200 g 1/x	8.33	-04
+	0.01	
5000 g 1/x	2.00	-04
+	0.01	
g 1/x	135.79	
PRX	135.79	

Sum of reciprocals.

The reciprocal of the sum of the reciprocals yields the answer in ohms.

```
220.00   1/X
560.00   1/X
           +
1200.00  1/X
           +
5000.00  1/X
           +
         1/X
135.79   ***
```

Square Roots

To calculate the square root of a number in the displayed X-register, press f √x. For example, to find the square root of 16:

Press	Display
16 f √x	4.00
PRX	4.00

```
16.00   √X
 4.00   ***
```

To find the square root of the result:

Press	Display
f √x	2.00
PRX	2.00

```
        √X
2.00    ***
```

Squaring

To square a number in the displayed X-register, press g x². For example, to find the square of 45:

Press	Display
45 g x²	2025.00
PRX	2025.00

```
45.00    X²
2025.00  ***
```

To find the square of the result:

Press	Display
g x²	4100625.00
PRX	4100625.00

```
         X²
4100625.00  ***
```

Using Pi

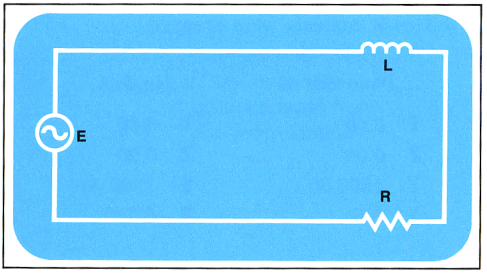
The value π accurate to 10 places (3.141592654) is provided as a fixed constant in the calculator. Merely press g π whenever you need it in a calculation. For example, to calculate 3π:

Press	Display
3 g π ×	9.42
PRX	9.42

```
3.00   Pi
       X
9.42   ***
```

Example: In the schematic diagram below, X_L is 12 kilohms, R is 7 kilohms, E is 120 volts, and f is 60 hertz. Find the inductance of the coil L in henries according to the formula:

$$L = \frac{X_L}{2\pi f}.$$



$$L = \frac{X_L}{2\pi f} = \frac{12{,}000}{2 \times \pi \times 60}$$

Press	Display	
12 EEX 3	12.	03
ENTER↑	12000.00	
2 ÷	6000.00	
g π ÷	1909.86	
60 ÷	31.83	Henries.
PRX	31.83	

```
12.+03 ENT↑
2.00     ÷
         Pi
         ÷
60.00    ÷
31.83    ***
```

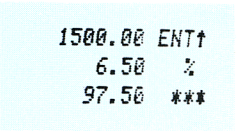
Percentages

The % key is a two-number function that allows you to compute percentages. To find the percentage of a number:

1. Key in the base number.
2. Press ENTER↑.
3. Key in the number representing percent rate.
4. Press the g prefix key.
5. Press %.

For example, to calculate a sales tax of 6.5% on a purchase of $1500:

Press	Display	
1500 ENTER↑	*1500.00*	Base number.
6.5	*6.5*	Percent rate.
g %	*97.50*	The answer.
PRX	*97.50*	



6.5% of $1500 is $97.50.

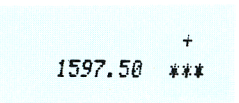
In the above example, when the % key is pressed, the calculated answer writes over the percentage rate in the X-register, and the base number is preserved in the Y-register.

When you pressed %, the stack contents were changed...

... from this ...		... to this.	
T	*0.00*	T	*0.00*
Z	*0.00*	Z	*0.00*
Y	*1500.00*	Y	*1500.00*
X	*6.5*	X	*97.50*

Since the purchase price is now in the Y-register and the amount of tax is in the X-register, the total amount can be obtained by simply adding:

Press	Display	
+	*1597.50*	Total of price and sales tax combined.
PRX	*1597.50*	



Trigonometric Functions

Your calculator provides you with six trigonometric functions, which operate in decimal degrees, radians, or grads. You can convert angles between decimal degrees and *degrees minutes, seconds,* and you can add and subtract angles in any of these forms without converting them.

Trigonometric Modes

For trigonometric functions, angles can be assumed by the calculator to be in decimal degrees, radians, or grads. To select decimal degrees mode, press g DEG *(degrees)* before using a trigonometric function. To select radians mode, press g RAD *(radians)*. Grads mode is selected with g GRD *(grads)*.

Note: 360 degrees = 400 grads = 2π radians.

Functions

The six trigonometric functions provided by the calculator are:

- f [sin] *(sine)*
- g [sin⁻¹] *(arc sine)*
- f [cos] *(cosine)*
- g [cos⁻¹] *(arc cosine)*
- f [tan] *(tangent)*
- g [tan⁻¹] *(arc tangent)*

Each trigonometric function assumes that angles are in decimal degrees, radians, or grads, depending upon the trigonometric mode selected.

All trigonometric functions are one-number functions, so to use them, you key in the number, then press the function key(s).

Example 1: Find the cosine of 35°.

Press	Display	
g [DEG]	*0.00*	Degrees mode selected. (Display assumes no results remain from previous examples.)
35	*35.*	
f [cos]	*0.82*	
[PRX]	*0.82*	

```
DEG
35.00 COS
 0.82 ***
```

Example 2: Find the arc sine in radians of .964.

Press	Display	
g [RAD]	*0.82*	Selects radians mode. (Result remains from previous example.)
0.964	*0.964*	
g [sin⁻¹]	*1.30*	Radians
[PRX]	*1.30*	

```
RAD
0.964 SIN⁻¹
 1.30 ***
```

Example 3: Find the tangent of 43.66 grads.

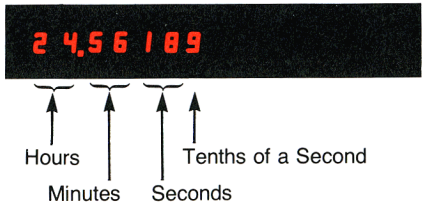
Press	Display	
g [GRD]	*1.30*	Selects grads mode. (Result remains from previous example.)
43.66	*43.66*	
f [tan]	*0.82*	Grads.
[PRX]	*0.82*	

```
GRAD
43.66 TAN
 0.82 ***
```

Hours, Minutes, Seconds/Decimal Hours Conversions

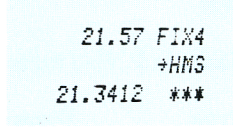
Using the HP-19C/HP-29C, you can change time specified in decimal hours to *hours, minutes, seconds* format by using the [→H.MS] *(hours to hours, minutes, seconds)* function; you can also

change from *hours, minutes, seconds* to decimal hours by using the [→H] *(hours, minutes, seconds to hours)* function. When a time is displayed or printed in *hours, minutes, seconds* format, the digits specifying *hours* occur to the left of the decimal point, while the digits specifying *minutes, seconds,* and *fractions of seconds* occur to the right of the decimal point.



To convert from decimal hours to *hours, minutes, seconds,* simply key in the value for decimal hours and press [f] [→H.MS]. For example, to change 21.57 hours to *hours, minutes, seconds:*

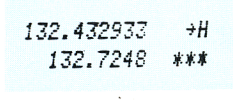
Press	Display	
21.57	*21.57*	Key in the decimal time.
[f] [FIX] 4	*21.5700*	Reset display format.
[f] [→H.MS]	*21.3412*	This is 21 hours, 34 minutes, 12 seconds.
[PRX]	*21.3412*	



Notice that the display is *not* automatically switched to show you more than the normal two digits after the decimal point ([FIX] 2), so to see the digits for *seconds,* you had to reset the display format to [FIX] 4.

To convert from *hours, minutes, seconds* to decimal hours, simply key in the value for *hours, minutes, seconds* in that format and press [g] [→H]. For example, to convert 132 hours, 43 minutes, and 29.33 seconds to its decimal degree equivalent:

Press	Display	
132.432933	*132.432933*	This is 132 hours, 43 minutes, 29.33 seconds.
[g] [→H]	*132.7248*	This is 132.7248 hours.
[PRX]	*132.7248*	



Using the [→H.MS] and [→H] operations, you can also convert angles specified in decimal degrees to *degrees, minutes, seconds,* and vice versa. The format for *degrees, minutes, seconds* is the same as for *hours, minutes, seconds.*

Example: Convert 42.57 decimal degrees to *degrees, minutes, seconds.*

Press	Display	
42.57	*42.57*	Key in the angle.
f →H.MS	*42.3412*	This means 42°34′ 12″. (Display assumes FIX 4 notation remains specified from previous example.)
PRX	*42.3412*	

```
42.5700 →HMS
42.3412  ***
```

Example: Convert 38°8′56.7″ to its decimal equivalent.

Press	Display	
38.08567	*38.08567*	Key in the angle.
g →H	*38.1491*	Answer in decimal degrees. (FIX 4 display specified from previous example.)
PRX	*38.1491*	
f FIX 2	*38.15*	Display mode reset.

```
38.08567  →H
38.1491   ***
          FIX2
```

In the HP-19C/HP-29C, trigonometric functions assume angles in decimal degrees, decimal radians, or decimal grads, so if you want to compute any trigonometric functions of an angle given in *degrees, minutes, and seconds,* you must first convert the angle to decimal degrees.

Example: Lovesick sailor Oscar Odysseus dwells on the island of Tristan da Cunha (37°03′S, 12°18′W), and his sweetheart, Penelope, lives on the nearest island. Unfortunately for the course of true love, however, Tristan da Cunha is the most isolated inhabited spot in the world. If Penelope lives on the island of St. Helena (15°55′S, 5°43′W), use the following formula to calculate the great circle distance that Odysseus must sail in order to court her.

$$\text{Distance} = \cos^{-1}\left[\sin(\text{LAT}_s)\sin(\text{LAT}_d) + \cos(\text{LAT}_s)\cos(\text{LAT}_d)\cos(\text{LNG}_d - \text{LNG}_s)\right] \times 60$$

Where:

LAT_s and LNG_s = latitude and longitude of the source (Tristan da Cunha).

LAT_d and LNG_d = latitude and longitude of the destination.

Solution: Convert all *degrees, minutes, seconds* entries into decimal degrees as you key them in. The equation for the great circle distance from Tristan da Cunha to the nearest inhabited land is:

$$\text{Distance} = \cos^{-1}\left[\sin(37°03') \sin(15°55') + \cos(37°03') \cos(15°55') \cos(5°43'\text{W} - 12°18'\text{W})\right] \times 60$$

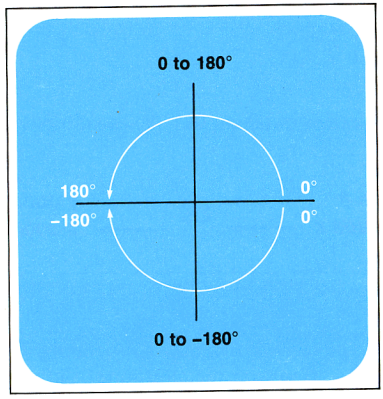
Press	Display	
g DEG	0.00	(Display assumes no results remain from previous examples.)
5.43	5.43	
g →H	5.72	
12.18	12.18	
g →H −	-6.58	
f cos	0.99	
15.55	15.55	
g →H	15.92	
STO 1	15.92	
f cos	0.96	
×	0.96	
37.03 g →H	37.05	
STO 0	37.05	
f cos	0.80	
×	0.76	
RCL 0 f sin	0.60	
RCL 1 f sin	0.27	
×	0.17	
+	0.93	
g cos⁻¹	21.92	
60 ×	1315.41	Distance in nautical miles that Odysseus must sail to visit Penelope.
PRX	1315.41	

```
                DEG
5.43            →H
12.18           →H
                -
                COS
15.55           →H
                STO1
                COS
                ×
37.03           →H
                STO0
                COS
                ×
                RCL0
                SIN
                RCL1
                SIN
                ×
                +
                COS⁻¹
60.00           ×
1315.41         ***
```

Polar/Rectangular Coordinate Conversions

Two functions, [→P] and [→R], are provided for polar/rectangular coordinate conversions. Polar angle θ is assumed in decimal degrees, radians, or grads, depending upon the trigonometric mode first selected by [DEG], [RAD], or [GRD].

In the HP-19C/HP-29C, polar angle θ is represented in the following manner:



To convert from rectangular x, y coordinates to polar r, θ coordinates (magnitude and angle, respectively):

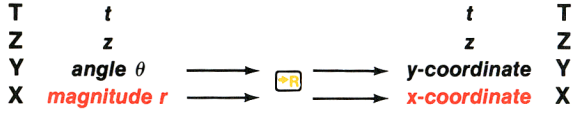
1. Key in the y-coordinate.
2. Press [ENTER↑] to raise the y-coordinate value to the Y-register of the stack.
3. Key in the x-coordinate.
4. Press [g] [→P] *(to polar)*. Magnitude r then appears in the X-register and angle θ is placed in the Y-register. (To display the value for θ, you press [x≷y].)

The following diagram shows how the stack contents change when you press [→P].

To convert from polar r, θ, coordinates to rectangular x, y, coordinates:

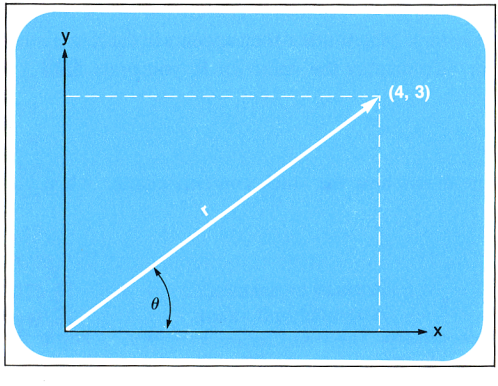
1. Key in the value for the angle θ.
2. Press ENTER↑ to raise the value for θ to the Y-register of the stack.
3. Key in the value for magnitude r.
4. Press f R→ *(to rectangular)*. The x-coordinate then appears in the displayed X-register and the y-coordinate is placed in the Y-register. (To display the value for the y-coordinate, you can press x≷y.)

The following diagram shows how the stack contents change when you press R→.



After you press P→ or R→, you can use the x≷y key to bring the calculated angle θ or the calculated y-coordinate into the X-register for viewing or further calculation.

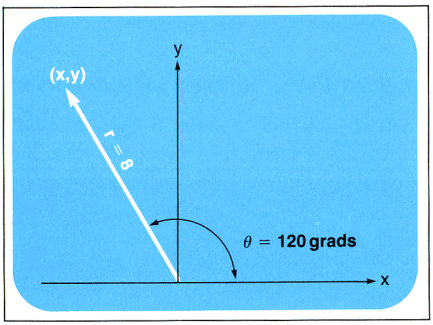
Example 1: Convert rectangular coordinates (4, 3) to polar form with the angle expressed in radians.



Press	Display	
g RAD	0.00	Radians mode selected. (Display assumes no results remain from previous examples.)
3 ENTER↑	3.00	y-coordinate entered into the Y-register.
4	4.	x-coordinate keyed into the X-register.
g →P	5.00	Magnitude r.
PRX	5.00	
x≷y	0.64	Angle θ in radians.
PRX	0.64	

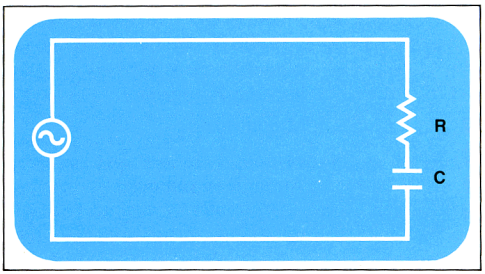
```
                RAD
3.00 ENT↑
4.00          →P
5.00          ***
                X≷Y
0.64          ***
```

Example 2: Convert polar coordinates (8, 120 grads) to rectangular coordinates.

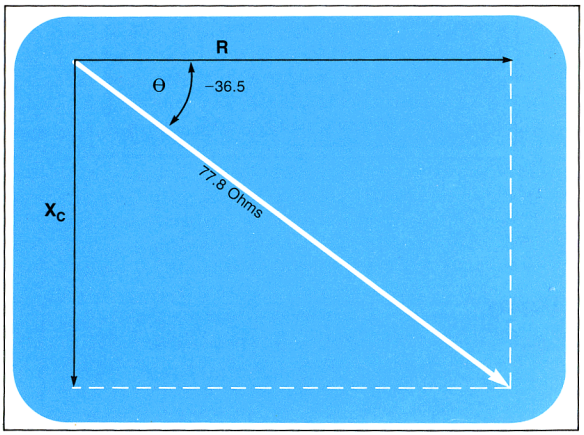


Press	Display	
g GRD	0.64	Grads mode selected. (Note that results can remain from previous examples.)
120 ENTER↑	120.00	Angle θ entered into the Y-register.
8	8.	Magnitude r placed in displayed X-register.
f →R	−2.47	x-coordinate.
x≷y	7.61	y-coordinate brought into displayed X-register for use, if desired.

```
                GRAD
120.00 ENT↑
8.00          →R
                X≷Y
```



Example 3: Engineer Trigo Slothrop has determined that in the RC circuit shown above, the total impedance is 77.8 ohms and voltage lags current by 36.5°. What are the values of resistance R and capacitive reactance X_c in the circuit?



Method: Draw a vector diagram using total impedance 77.8 ohms for polar magnitude *r* and −36.5° for angle θ. When the values are converted to rectangular coordinates, the x-coordinate value yields resistance R in ohms, and the y-coordinate value yields reactance X_c in ohms.

Solution:

Press	Display	
g DEG	*7.61*	Degrees mode selected. (Note that results can remain from previous examples.)
36.5 CHS	*-36.5*	
ENTER↑	*-36.50*	
77.8	*77.8*	
f →R	*62.54*	Resistance R in ohms.
x≷y	*-46.28*	Reactance X_c, 46.28 ohms, available in displayed X-register.

```
                DEG
-36.50 ENT↑
 77.80   →R
         X≷Y
```

Logarithmic and Exponential Functions

Logarithms

The HP-19C/HP-29C computes both natural and common logarithms as well as their inverse functions (antilogarithms):

- f ln is $\log_e$ (natural log). It takes the log of the value in the X-register to base e (2.718...).
- g e^x is antilog$_e$ (natural antilog). It raises e (2.718...) to the power of the value in the X-register. (To display the value of e, press 1 g e^x.)
- f log is $\log_{10}$ (common log). It computes the log of the value in the X-register to base 10.
- g 10^x is antilog$_{10}$ (common antilog). It raises 10 to the power of the value in the X-register.

Example 1: The 1906 San Francisco earthquake, with a magnitude of 8.25 on the Richter scale, is estimated to be 105 times greater than the Nicaragua quake of 1972. What would be the magnitude of the latter on the Richter scale? The equation is:

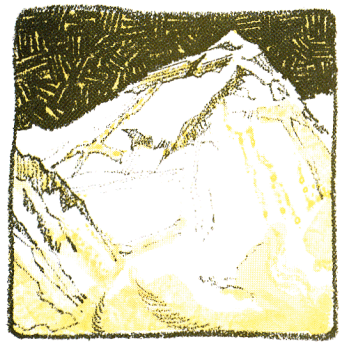
$$R_1 = R_2 - \log\frac{M_2}{M_1} = 8.25 - \left(\log\frac{105}{1}\right)$$

Solution:

Press	Display	
8.25 ENTER↑	*8.25*	
105 f log	*2.02*	
−	*6.23*	Rating on Richter scale.
PRX	*6.23*	

```
                8.25 ENT↑
              105.00  LOG
                      -
                6.23  ***
```

Example 2: Having lost most of his equipment in a blinding snowstorm, ace explorer Jason Quarmorte is using an ordinary barometer as an altimeter. After measuring the sea level pressure (30 inches of mercury) he climbs until the barometer indicates 9.4 inches of mercury. Although the exact relationship of pressure and altitude is a function of many factors, Quarmorte knows that an *approximation* is given by the formula:



$$\text{Altitude (feet)} = 25{,}000 \ln \frac{30}{\text{Pressure}}$$

$$= 25{,}000 \ln \frac{30}{9.4}$$

Where is Jason Quarmorte?

Solution:

Press	Display		
30 ENTER↑	*30.00*		30.00 ENT↑
9.4 ÷	*3.19*		9.40 ÷
f ln	*1.16*		LN
25000	*25000.*		25000.00 ×
×	*29012.19*	Altitude in feet.	29012.19 ***
PRX	*29012.19*		

Quarmorte is probably near the summit of Mount Everest (29,028 feet).

Raising Numbers to Powers

The y^x key is used to raise numbers to powers. Using y^x permits you to raise a positive real number to any real power—that is, the power may be positive or negative, and it may be an integer, a fraction, or a mixed number. y^x also permits you to raise any negative real number to the power of any integer (within the calculating range of the calculator, of course).

For example, to calculate 2^9 (that is, $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$):

Press	Display	
2 ENTER↑ 9	*9.*	2.00 ENT↑
f y^x	*512.00*	9.00 Y^x
PRX	*512.00*	512.00 ***

To calculate $8^{-1.2567}$:

Press	Display
8 ENTER↑	8.00
1.2567 CHS	-1.2567
f y^x	0.07
PRX	0.07

```
8.00 ENT↑
-1.2567  Yˣ
0.07  ***
```

To calculate $(-2.5)^5$:

Press	Display
2.5 CHS	-2.5
ENTER↑	-2.50
5 f y^x	-97.66
PRX	-97.66

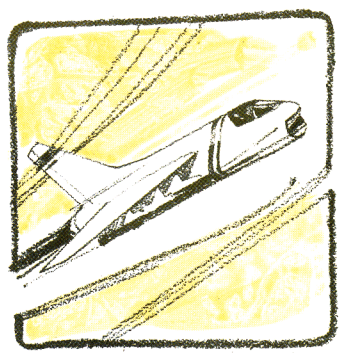
```
-2.50 ENT↑
5.00  Yˣ
-97.66  ***
```

In conjunction with $1/x$, y^x provides a simple way to extract roots. For example, find the cube root of 5. (This is equivalent to $5^{1/3}$.)

Press	Display	
5 ENTER↑	5.00	
3 g $1/x$	0.33	Reciprocal of 3.
f y^x	1.71	Cube root of 5.
PRX	1.71	

```
5.00 ENT↑
3.00  1/X
      Yˣ
1.71  ***
```

Example: In a rather overoptimistic effort to break the speed of sound, highflying pilot Ike Daedalus cranks open the throttle on his surplus Hawker Siddeley Harrier aircraft. From his instruments he reads a pressure altitude (PALT) of 25,500 feet with a calibrated airspeed (CAS) of 350 knots. What is the flight mach number



$$M = \frac{\text{speed of aircraft}}{\text{speed of sound}}$$

if the following formula is applicable?

$$M = \sqrt{5\left[\left(\left\{\left[\left(1 + 0.2\left[\frac{350}{661.5}\right]^2\right)^{3.5} - 1\right]\left[1 - (6.875 \times 10^{-6})\,25{,}500\right]^{-5.2656}\right\} + 1\right)^{0.286} - 1\right]}$$

Method: The most efficient place to begin work on this problem is at the innermost set of brackets. So begin by solving for the quantity $\left[\frac{350}{661.5}\right]^2$ and proceed outward from there.

Press	Display		
350 ENTER↑	350.00		
661.5 ÷	0.53		
g x^2	0.28		Square of bracketed quantity.
.2 × 1 +	1.06		
3.5 f y^x 1 −	0.21		Contents of left-hand set of brackets are in the stack.
1 ENTER↑	1.00		
6.875 EEX	6.875	00	
CHS 6 ENTER↑	6.88	06	
25500 × −	0.82		Contents of right-hand set of brackets are in the stack.
5.2656 CHS	−5.2656		
f y^x	2.76		
× 1 +	1.58		
.286 f y^x	1.14		
1 −	0.14		
5 × f $\sqrt{x}$	0.84		Mach number of Daedalus' Harrier.
PRX	0.84		

```
350.00 ENT↑
661.50  ÷
        X²
  0.20  X
  1.00  +
  3.50  Yˣ
  1.00  -

  1.00 ENT↑
6.875-06 ENT↑
25500.00  X
          -

-5.2656  Yˣ
         X
  1.00   +
 0.286   Yˣ
  1.00   -
  5.00   X
         √X
  0.84   ***
```

In working through complex equations like the one containing six levels of parentheses above, you really appreciate the value of the Hewlett-Packard logic system. Because you calculate one step at a time, you don't get "lost" within the problem. You see every intermediate result, and you emerge from the calculation confident of your final answer.

Statistical Functions

Accumulations

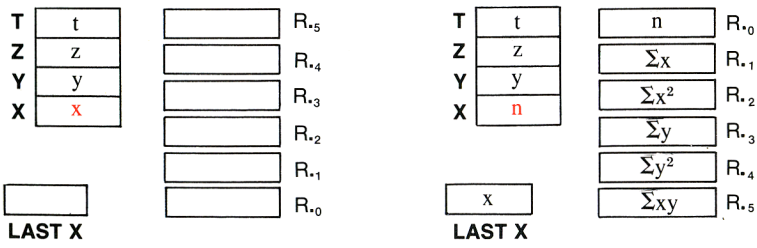
Pressing the Σ+ key automatically gives you several different sums and products of the values in the X- and Y-registers at once. In order to make these values accessible for sophisticated statistics problems, they are automatically placed by the calculator into storage registers R_0 through R_5. *The only time that information is automatically accumulated in the storage registers is when the* Σ+ (or Σ−) key is used. Before you begin any calculations using the Σ+ key, you should first clear the storage registers used in accumulations by pressing f CLEAR Σ.

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

1. The number that you keyed into the X-register is added to the contents of storage register $R_{.1}$. ($\Sigma x \rightarrow R_{.1}$)
2. The square of the number that you keyed into the X-register is added to the contents of storage register $R_{.2}$. ($\Sigma x^2 \rightarrow R_{.2}$)
3. The number in the Y-register of the stack is added to the contents of storage register $R_{.3}$. ($\Sigma y \rightarrow R_{.3}$)
4. The square of the number in the Y-register of the stack is added to the contents of storage register $R_{.4}$ ($\Sigma y^2 \rightarrow R_{.4}$)
5. The number that you keyed into the X-register is multiplied by the contents of the Y-register, and the product added to storage register $R_{.5}$. ($\Sigma xy \rightarrow R_{.5}$)
6. The number 1 is added to storage register $R_{.0}$, and the total number in $R_{.0}$ then writes over the number in the displayed X-register of the stack. The stack does not lift.

The number that you keyed into the X-register is preserved in the LAST X register, while the number in the stack Y-register remains in the Y-register.

Thus, when you press Σ+, the stack and storage register contents are changed...



Before you begin accumulating results in primary storage registers $R_{.0}$ through $R_{.5}$ using the Σ+ key, you should first ensure that the contents of these registers have been cleared to zero by pressing f CLEAR Σ.

Note: Unlike storage register arithmetic, the Σ+ function allows overflows (i.e., numbers whose magnitudes are greater than $9.999999999 \times 10^{99}$) in storage register $R_{.0}$ through $R_{.5}$ without registering ***Error*** in the display (or on the HP-19C printed copy).

To use *only* the Σx and Σy that you have accumualted in the storage registers, you can press RCL followed by $\Sigma+$. 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, like that illustrated on pages 90-91.)

To use *any* of the summations individually at any time, you can recall the contents of the desired storage register into the displayed X-register by pressing RCL . followed by the number key of the storage register address. After you have pressed $\Sigma+$, recalling storage register contents or keying in another number 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 the paired values of x and y listed below.

y	7	5	9
x	5	3	8

Press	Display	
f CLEAR Σ	*0.00*	Ensures that storage registers $R._0$ through $R._5$ are cleared to zero initially. Display assumes no results remain from previous example.
7 ENTER↑	*7.00*	
5 Σ+	*1.00*	First pair is accumulated; n = 1.
5 ENTER↑	*5.00*	
3 Σ+	*2.00*	Second pair is accumulated; n = 2.
9 ENTER↑	*9.00*	
8 Σ+	*3.00*	Third pair is accumulated; n = 3.
RCL . 1	*16.00*	Sum of x values from register $R._1$.
RCL . 2	*98.00*	Sum of squares of x values from register $R._2$.
RCL . 3	*21.00*	Sum of y values from register $R._3$.
RCL . 4	*155.00*	Sum of squares of y values from register $R._4$.
RCL . 5	*122.00*	Sum of products of x and y values from register $R._5$.
RCL . 0	*3.00*	Number of entries (n = 3) from register $R._0$.

```
          CLΣ
7.00 ENT↑
5.00   Σ+
5.00 ENT↑
3.00   Σ+
9.00 ENT↑
8.00   Σ+
       RC.1
       RC.2
       RC.3
       RC.4
       RC.5
       RC.0
```

Printing Accumulations (HP-19C)

You can see *all* of the values accumulated by the Σ+ key at any time. Simply press f PRTΣ, and the printer will print out the contents of the storage registers used for summations along with a description for each summation.

For example, to list *all* of the accumulations that are now in the storage registers from the previous example:

Press	Display
f PRTΣ	*3.00*

```
                PRTΣ
          3.00   N
         16.00   ΣX
         98.00   ΣX²
         21.00   ΣY
        155.00   ΣY²
        122.00   ΣXY
```

Mean

The x̄ *(mean)* key is the key you use to calculate the mean (arithmetic average) of x and y accumulated in registers $R_{.1}$ and $R_{.3}$, respectively.

When you press f x̄:

1. The mean ($\bar{x}$) of x is calculated using the data accumulated in register $R_{.1}$ (Σx) and $R_{.0}$ (n) according to the formula:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \left(\text{That is, } \frac{R_{.1}}{R_{.0}} = \bar{x}\right)$$

   The resultant value for $\bar{x}$ is seen in the displayed X-register.

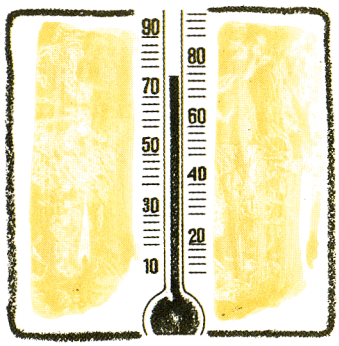
2. The mean ($\bar{y}$) of y is calculated using the data accumulated in register $R_{.3}$ (Σy) and register $R_{.0}$ (n) according to the formula:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad \left(\text{That is, } \frac{R_{.3}}{R_{.0}} = \bar{y}\right)$$

   The resultant value for $\bar{y}$ is available in the Y-register of the stack.

The easiest way to accumulate the required data in the applicable registers is through the use of the Σ+ key as described above.

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 for the week selected?

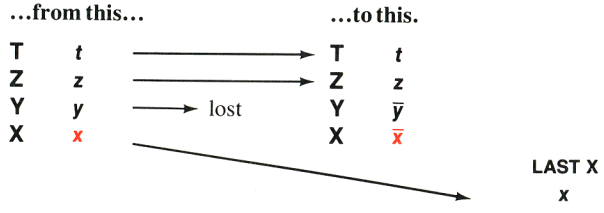


	Sun	Mon	Tues	Wed	Thurs	Fri	Sat
High	6	11	14	12	5	−2	−9
Low	−22	−17	−15	−9	−24	−29	−35

f CLEAR Σ	*0.00*	Accumulation registers cleared. (Display assumes no results remain from previous calculations.)
6 ENTER↑ 22		
CHS Σ+	*1.00*	Number of data pairs (n) is now 1.
11 ENTER↑ 17		
CHS Σ+	*2.00*	Number of data pairs (n) is now 2.
14 ENTER↑ 15		
CHS Σ+	*3.00*	
12 ENTER↑ 9		
CHS Σ+	*4.00*	
5 ENTER↑ 24		
CHS Σ+	*5.00*	
2 CHS ENTER↑	*−2.00*	
29 CHS Σ+	*6.00*	
9 CHS ENTER↑	*−9.00*	
35 CHS Σ+	*7.00*	Number of data pairs (n) is now 7.
f x̄	*−21.57*	Average low temperature.
PRX	*−21.57*	
x≷y	*5.29*	Average high temperature.
PRX	*5.29*	

```
                CLΣ
          6.00 ENT↑
        -22.00   Σ+
         11.00 ENT↑
        -17.00   Σ+
         14.00 ENT↑
        -15.00   Σ+
         12.00 ENT↑
         -9.00   Σ+
          5.00 ENT↑
        -24.00   Σ+
         -2.00 ENT↑
        -29.00   Σ+
         -9.00 ENT↑
        -35.00   Σ+
                 x̄
        -21.57  ***
                 x≷y
          5.29  ***
```

The illustration below represent what happens in the stack when you press f x̄. Press f x̄ and the contents of the stack are changed...



Standard Deviation

The [s] *(standard deviation)* key is the key you use to calculate the sample standard deviation (a measure of dispersion around the mean) of data accumulated in storage registers $R_{.0}$ through $R_{.5}$.

When you press [f] [s]:

1. Sample x standard deviation (s_x) is calculated using the data accumulated in storage register $R_{.2}$ (Σx^2), $R_{.1}$ (Σx), and $R_{.0}$ (n) according to the formula:

$$s_x = \sqrt{\frac{\Sigma x^2 - \frac{(\Sigma x)^2}{n}}{n-1}}$$

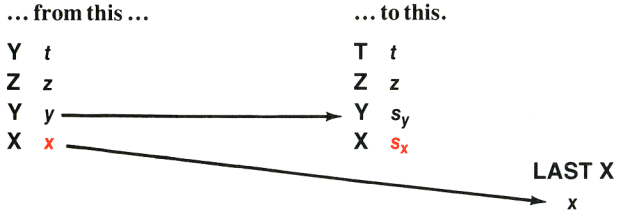
   The resultant value for standard deviation of x (s_x) is seen in the displayed X-register.

2. Sample y standard deviation (s_y) is calculated using the data accumulated in storage registers $R_{.4}$ (Σy^2), $R_{.3}$ (Σy), and $R_{.0}$ (n) according to the formula:

$$s_y = \sqrt{\frac{\Sigma y^2 - \frac{(\Sigma y)^2}{n}}{n-1}}$$

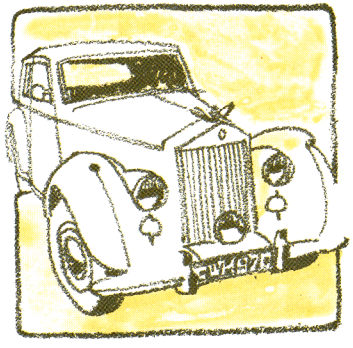
   The resultant value for standard deviation of y (s_y) is available in the Y-register of the stack.

Thus, with data accumulated in registers $R_{.0}$ through $R_{.5}$, when you press [f] [s], the contents of the stack registers are changed...



To use the value for standard deviation of y (s_y) simply use the [x⇄y] key to bring that value into the displayed X-register of the stack.

Example: In a recent survey to determine the age and net worth (in millions of dollars) of six of the 50 wealthiest persons in the United States, the following data were obtained (sampled). Calculate the average age and net worth of the sample, and calculate the standard deviations for these two sets of data.



Age	62	58	62	73	84	68
Net Worth	1200	1500	1450	1950	1000	1750

Press	Display	
f CLEAR Σ	0.00	Clears storage registers used for Σ+ (Display assumes no results remain from previous examples.)
62 ENTER↑	62.00	
1200 Σ+	1.00	Number of data pairs (n) is 1.
58 ENTER↑	58.00	
1500 Σ+	2.00	
62 ENTER↑	62.00	
1450 Σ+	3.00	
73 ENTER↑	73.00	
1950 Σ+	4.00	
84 ENTER↑	84.00	
1000 Σ+	5.00	
68 ENTER↑	68.00	
1750 Σ+	6.00	Number of data pairs (n) is 6.
f x̄	1475.00	Average value of net worth.
x≷y	67.83	Average age of the sample.
f s	347.49	Standard deviation (s_x) of net worth of sample.
x≷y	9.52	Standard deviation (s_y) of age of sample.

```
                CLΣ
          62.00 ENT↑
        1200.00   Σ+
          58.00 ENT↑
        1500.00   Σ+
          62.00 ENT↑
        1450.00   Σ+
          73.00 ENT↑
        1950.00   Σ+
          84.00 ENT↑
        1000.00   Σ+
          68.00 ENT↑
        1750.00   Σ+
                  x̄
                  X≷Y
                  S
                  X≷Y
```

If the six persons used in the sample were actually the *six wealthiest persons,* the data would have to be considered as a population rather than as a sample. The relationship between sample standard deviation (s) and the population standard deviation (σ) is illustrated by the following equation.

$$\sigma = s\sqrt{\frac{n-1}{n}}$$

Since n is automatically accumulated in register $R_{.0}$ when data is accumulated, it is a simple matter to convert the sample standard deviations that have already been calculated to population standard deviations.

If the accumulations are still intact from the previous example in registers $R_{.0}$ through $R_{.5}$, you can calculate the population standard deviations this way:

Press	Display	
f S	347.49	Calculate s_x and s_y.
RCL . 0	6.00	Recall n.
1 −	5.00	Calculate n − 1.
RCL . 0 ÷	0.83	Divide n − 1 by n.
f √x ×	317.21	Population standard deviation σ_x.
PRX	317.21	
x≷y	9.52	Brings s_y to the X-register.
f LAST X	0.91	Recall conversion factor.
×	8.69	Population standard deviation σ_y.
PRX	8.69	

```
                S
             RC.0
1.00            -
             RC.0
                ÷
               √X
317.21         ***
               X≷Y
             LSTX
                X
 8.69          ***
```

Deleting and Correcting Data

If you key in an incorrect value and have not pressed Σ+ , press CL x and key in the correct value.

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

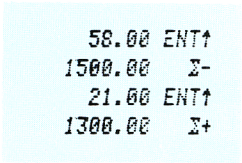
1. Key in the *incorrect* data pair into the X- and Y-registers. (You can use LAST X to return a single incorrect data value to the displayed X-register.)
2. Press f Σ- to delete the incorrect data.
3. Key in the correct values for x and y. (If one value of an x, y data pair is incorrect, both values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

The correct values for mean and standard deviation are now obtainable by pressing f x̄ and f s.

For example, suppose the 58-year old member of the *sample* as given above were to lose his position as one of the wealthiest persons because of a series of ill-advised investments in cocoa futures. To account for the change in data if he were replaced in the sample by

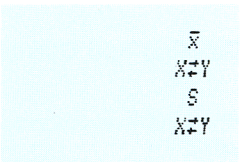
a 21-year old rock musician who is worth 1300 million dollars:

Press	Display	
58 ENTER↑	58.00	
1500	1500.	Data to be replaced.
f Σ-	5.00	Number of entries (n) is now five.
21 ENTER↑	21.00	
1300	1300.	
Σ+	6.00	Number of entries (n) is now six again.



The new data has been calculated into each of the summations present in the storage registers. To see the new mean and standard deviation:

Press	Display	
f x̄	1441.67	The new average (mean) worth.
x⇄y	61.67	The new average (mean) age available in X-register for use.
f s	354.14	The new standard deviation for worth.
x⇄y	21.55	The new standard deviation for age available in X-register for use.

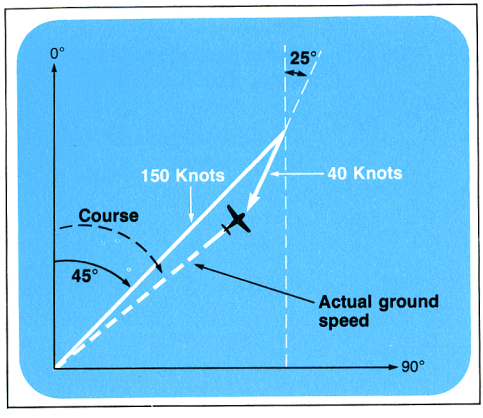


Vector Arithmetic

You can use your HP-19C/HP-29C to add or subtract vectors by combining the polar/rectangular conversion functions (the →P and →R keys) with the summation functions (the Σ+ and Σ- keys).

Example: 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 headwind of 40 knots from a bearing of 25°. What is the actual ground speed and course of the Swordfish?

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



Press	Display	
f CLEAR Σ	*0.00*	Clears summation registers $R_{.0}$ through $R_{.5}$. (Display assumes no results remain from previous examples.)
g DEG	*0.00*	Sets degrees mode.
45 ENTER↑	*45.00*	θ for 1st vector is entered to Y-register.
150	*150.*	r for 1st vector is keyed in.
f →R	*106.07*	Converted to rectangular coordinates.
Σ+	*1.00*	1st vector coordinates accumulated in storage registers $R_{.1}$ and $R_{.3}$.
25 ENTER↑	*25.00*	θ for 2nd vector is entered to Y-register.
40	*40.*	r for 2nd vector is keyed in.
f →R	*36.25*	2nd vector is converted to rectangular coordinates.
f Σ-	*0.00*	2nd vector rectangular coordinates subtracted from those of 1st vector.

```
          CLΣ
          DEG
45.00 ENT↑
150.00    →R
          Σ+
25.00 ENT↑
40.00     →R
          Σ-
```

RCL Σ+	*69.81*	Recalls both $R_{.1}$ and $R_{.3}$.
g →P	*113.24*	Actual ground speed in knots of the Swordfish.
PRX	*113.24*	
x≷y	*51.94*	Course in degrees of the Swordfish.
PRX	*51.94*	

```
                RCΣ+
                →P
      113.24    ***
                X≷Y
       51.94    ***
```

PART TWO
Programming the HP-19C/HP-29C

Section 6

Simple Programming

If you read Meet the HP-19C and HP-29C (pages 13-17), you have already seen that by using the programming capability of your calculator, you can increase the flexibility of the calculator a hundredfold or more, and you save hours of time in long computation. The Continuous Memory permits you to key in programs and save them permanently, even though the HP-19C/HP-29C is turned OFF.

With your HP-19C/HP-29C Scientific Calculator, Hewlett-Packard has provided you with the *HP-19C/HP-29C Applications Book,* containing dozens of programs. You can begin using the programming power of the calculator immediately by keying any of these programs into your HP-19C/HP-29C and then running it as often as you like.

However, in order to get the most from your calculator, you'll want to learn to program yourself, to solve your own problems. This part of the *HP-19C/HP-29C Owner's Handbook and Programming Guide* teaches you step by step to create simple programs that will solve complex problems, then introduces you to the many editing features of the calculator, and finally gives you a glimpse of just how sophisticated your programming can become.

Programming your calculator is an extension of its use as a *manual* problem-solving machine, so if you haven't read Part One, Using Your HP-19C/HP-29C Calculator, you should go back and do so before you begin programming.

After most of the explanations and examples in this part, you will find problems to work using your HP-19C/HP-29C. These problems are not essential to your basic understanding of the calculator, and they can be skipped if you like. But we urge that you work them. They are rarely difficult, and they have been designed to increase your proficiency, both in the actual use of the features on your calculator and in creating programs to solve your *own* problems. If you have trouble with one of the problems, go back and review the explanations in the text, then tackle it again.

So that you can apply your own creative flair to the problems, no solutions are given for them. In programming, any solution that gives the correct outputs is the right one—there is no *one* correct program for any problem. In fact, when you have finished working through this part, and learned all the capabilities of the calculator, you may be able to create programs that will solve many of the problems faster, or in fewer steps, than we have shown in our illustrations.

Now let's start programming!

Set the HP-19C Print Mode Switch MAN TRACE NORM to MAN so you can concentrate on programming.

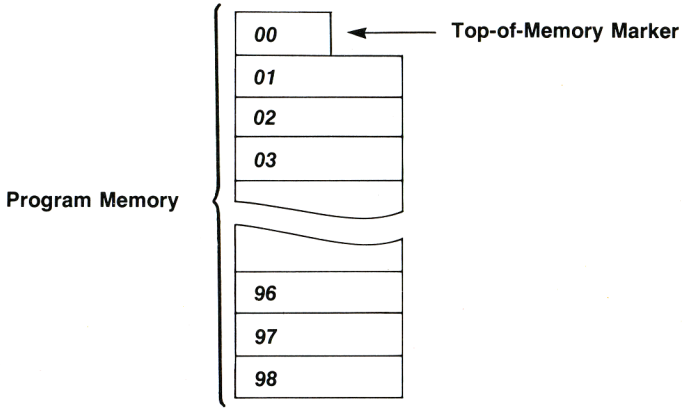
What is a Program?

A *program* is nothing more than a series of calculator keystrokes that you would press to solve a problem manually. The calculator remembers these keystrokes when you key them in, then executes them in order whenever you wish.

Looking at Program Memory

If you worked through Meet the HP-19C and HP-29C (pages 13-17), you learned how to write, load and run a simple program to solve for the area of a sphere. If you haven't disturbed program memory, this program is still in program memory (maintained by Continuous Memory). However, if you have cleared or changed the contents of program memory, return now to page 15 and reenter the program so that you can analyze it further.

As you may remember from this program, a program is nothing more than a series of keystrokes you would press to solve a problem manually. These keystrokes are stored in a part of the calculator known as *program memory*. When you set the calculator to PRGM mode, you can examine the contents of program memory, one step at a time.



First set the calculator to PRGM mode. Then press GTO . 00 (go to step number 00) to return the calculator to the beginning of program memory. The number that you see on the left side of the display indicates the *step number* of program memory to which the calculator is set. You should be set at step 00, indicated by a display of *00*. Now we'll use the SST *(single-step)* key to examine the next step of program memory. SST lets you step through program memeory, one step at a time.

Press	HP-19C Display	HP-29C Display
SST	*01 25 14 00*	*01 15 13 00*

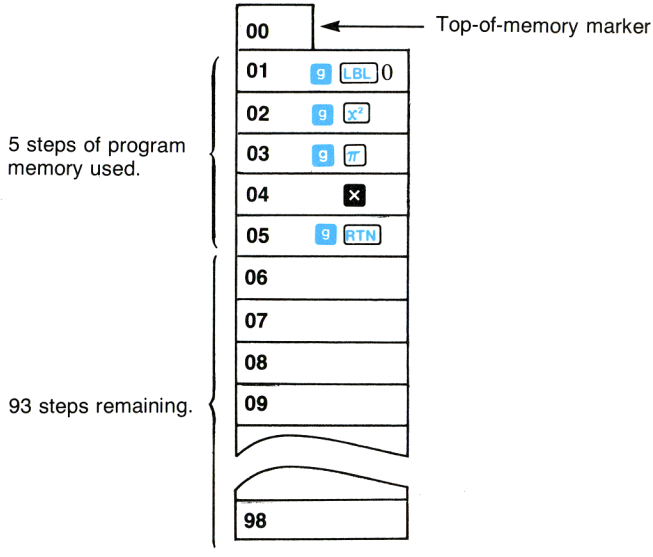
The calculator is now set to step 01 of program memory, as indicated by the number 01 that you see on the left side of the display. The other numbers in the display are two-digit *keycodes* for the keystrokes that have been loaded into that step of program memory (more about keycodes later).

Each step of program memory can "remember" a single operation, whether that operation consists of one, two, three, or four keystrokes. Thus, one step of program memory might contain a single-keystroke operation like CHS, while another step of program memory could contain a two-keystroke operation, like STO 6. Step 01 of program memory currently contains an operation that requires three keystrokes, g LBL 0.

Program Memory

In the HP-19C/HP-29C, keystrokes that make up programs are loaded and stored in a portion of the calculator called *program memory*. Program memory consists of 98 steps and is separate from the stack and storage registers.

When you loaded the program to calculate the area of a sphere into your calculator, the function keys you pressed were "remembered" in program memory after the [LBL] 0 program marker, so the contents of program memory now look like this:



Since you used five steps of program memory for the program with the 0 label, there are still 93 unused steps of program memory for other programs. You can allocate the 93 remaining steps of program memory any way you choose. For example, you could have a 5-step program labeled by 0, a 53-step program labeled 9, and a 5-step program in each of the eight remaining labels 1, 2, 3, 4, 5, 6, 7, and 8. Or you could have fewer programs utilizing only a portion of memory. Or you could use all 98 steps of program memory for one program if you wished. Each label can even be used more than once!

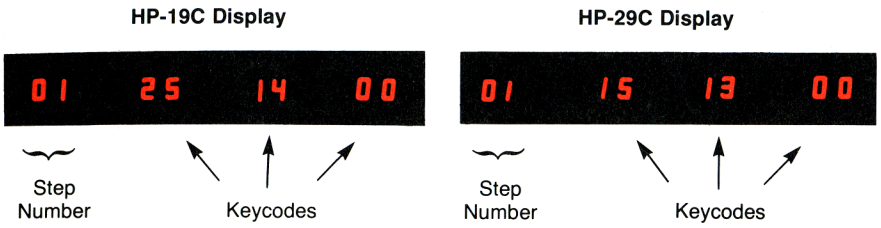
Keycodes

You can tell from the display what keystrokes are loaded into each step of a program by means of the *keycodes* for those keys. Let's look at some keycodes now.

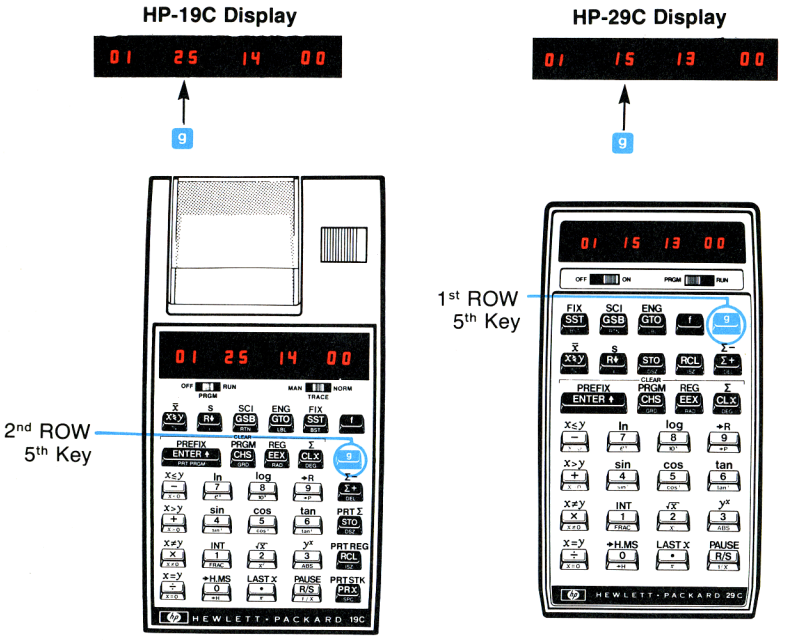
1. First press GTO [•] 00 to ensure that the calculator is set to the top of program memory.
2. HP-19C: Slide the OFF-PRGM-RUN switch OFF PRGM RUN to PRGM.

   HP-29C: Slide the PRGM-RUN switch PRGM RUN to PRGM.

3. Now press the SST *(single-step)* key. This moves the calculator down one step of program memory, and your display should look like the one below. The step number is on the left, and the keycode is on the right of the display.



Each key on the calculator has a keycode. For each keycode, the first digit denotes the *row* of the key on the keyboard and the second digit identifies the number of the key in that row. Always count from the top down, and from left to right. Each key, no matter how large, counts as one.



Now use the SST key again to examine the keycode in the next step of the program:

Press	HP-19C Display	HP-29C Display
SST	02 25 53	02 15 63

For convenience, digit keys are identified by two digit keycodes of 00 through 09, except when prefixed by f or g. When prefixed by f or g the digit keys are identified by the row-column keycode. Since the π key is preceded by the g key, the complete operation contained in step 02 is g π.

Using this handy matrix system, you can easily identify any key by its keycode. Each step of program memory can contain either a single digit or a complete operation, no matter how many keystrokes the operation requires.

Problems

1. What would be the keycodes for the following operations: CHS, g GRD, f →H.MS, STO + 1?

2. What operations are identified by the following keycodes:
   HP-19C: 16 64, 11, 45 01, 45 41 01.
   HP-29C: 14 74, 21, 23 01, 23 51 01.

3. How many steps 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 2 50 STO × 2 RCL 2 f √x ×

4. What keystroke(s) would you load into a program to perform an *x exchange y?* (That is, to exchange the contents of the X-register with those of the Y-register.)

Clearing a Program

The Continuous Memory of your calculator preserves any programs loaded into program memory even though the calculator is turned OFF.

To clear the complete program memory, all 98 steps, press f CLEAR PRGM while in PRGM mode. The calculator is automatically set to step 00.

To clear a selected program from the calculator:

1. HP-19C: Set the OFF-PRGM-RUN switch OFF PRGM RUN to PRGM.
   HP-29C: Set the PRGM-RUN switch PRGM RUN to PRGM.
2. Locate the program you wish to clear by pressing GTO and the label number.
3. Using the SST key, step through the program until you come to the last step of the program. If you count the number of steps in the program, deletion may be easier. (Refer to Program Editing, page 112, for more information).
4. Press g DEL for each step you wish to delete. If you counted the number of steps in the program, you can do that number of deletions. Be careful not to delete beyond the beginning of your program; watch for the label keycode. As you delete each step the program memory automatically moves all subsequent steps up one step and displays the previous step. (Refer to Program Editing, page 112, for more information.)

All steps of program memory formerly used by the cleared program are again available for storing program instructions for any program.

If power has been interrupted to the calculator (that is, battery failure) all instructions in program memory are lost and the calculator resets to step 00.

Creating a Program

In Meet the HP-19C and HP-29C, at the beginning of this handbook, you created, loaded, and ran a program that solved for the surface area of a sphere, given the diameter of that sphere. Now let's create, load, and run another program to show you how to use some of the other features of the calculator.

If you wanted to use the HP-19C/HP-29C to manually calculate the area of a circle using the formula $A = \pi r^2$, you could first key in the radius r, then square it by pressing g x^2. Next you would summon the quantity pi into the display by pressing g π. Finally you would multiply the squared radius and the quantity pi together by pressing ×.

Remember that a program to solve a problem is nothing more than the keystrokes you would press to solve the problem manually. Thus, in order to create a program that will solve for the area of any circle, you use the same keys you pressed to solve the problem manually.

The keys that you used to solve for the area of a circle according to the formula $A = \pi r^2$ are:

g x^2
g π
×

You will load these keystrokes into program memory. In addition, your program will contain two other operations, LBL n and RTN.

The Beginning of a Program

To define the beginning of a program, you should use a g LBL *(label)* instruction followed by one of the program markers 0 through 9. The use of labels permits you to have several different programs or parts of programs loaded into the calculator at any time and to run them in any order you choose.

Ending a Program

To define the end of a program, you should use a g RTN *(return)* instruction. When the calculator is executing a program and encounters a RTN instruction in program memory, it stops (unless executed as part of a subroutine—more about subroutines later). For example, if the calculator were executing a program that had begun with LBL 1, when it encountered g RTN, it would stop. Another instruction that will cause a running program to stop is R/S. When a running program executes a R/S instruction in program memory, it stops just as it does when it executes RTN. Good programming practice, however, dictates that you normally use g RTN rather than R/S to define the end of your program.

The Complete Program

The complete program to solve for the area of any circle given its radius is now:

g LBL 1	Assigns name to and defines beginning of program.
g x^2	Squares the radius.
g π	Summons pi into the display.
×	Multiplies r^2 by π and displays the answer.
g RTN	Defines the end of and stops the program.

Loading a Program

When the calculator is set to PRGM, the functions and operations that are normally executed when you press the keys are not executed. Instead, they are stored in program memory for later execution. *All but seven operations on the keyboard can be loaded into program memory for later execution.* The seven operations that cannot be loaded in as part of a program are:

f CLEAR PREFIX, g PRT PRGM (HP-19C), f CLEAR PRGM, SST, g BST, g DEL, GTO • nn.

All other functions are loaded into the calculator as program instructions to be executed later

To load a complete program into the calculator:

1. HP-19C: Slide the OFF-PRGM-RUN switch OFF PRGM RUN to PRGM.
   HP-29C: Slide the PRGM-RUN switch PRGM RUN to PRGM.
2. Press f CLEAR PRGM to clear program memory of any previous programs and to reset the calculator to the top of program memory.

You can tell that the calculator is at the top of program memory because the digits 00 appear at the left of the display. The digits appearing at the left of the display when the calculator is in PRGM mode indicate the *program memory step number* being shown at any time.

The keys that you must press to key in a program for the area of a circle are:

g LBL 1
g x^2
g π
×
g RTN

Press the first key, g, of the program.

Press	HP-19C	HP-29C
g	00	00

You can see that the display of program memory has not changed. Now press the second and third keys of the program.

Press	HP-19C	HP-29C
LBL	00	00
1	01 25 14 01	01 15 13 01

When the step number (01) of program memory appears on the left of the display, it indicates that a complete operation has been loaded into that step. Nothing is loaded into program memory until a complete operation (whether 1, 2, 3, or 4 keystrokes) has been specified.

Now load the remainder of the program by pressing the keys. Observe the program memory step numbers and keycodes.

Press	HP-19C	HP-29C
g x^2	02 25 53	02 15 63
g π	03 25 63	03 15 73
×	04 51	04 61
g RTN	05 25 13	05 15 12

The program for solving the area of a circle is now loaded into program memory. Notice that nothing could be loaded into the top-of-memory marker, step 00.

Running a Program

To run a program you have only to set the calculator to RUN mode. Then key in any needed data, and press GSB and the number key (0 through 9) that labels your program.

For example, to use the program now in the calculator to solve for circles with radii of 3 inches, 6 meters, and 9 miles:

First, set the calculator to RUN.

Press	Display	
3 GSB 1	28.27	Square inches.
6 GSB 1	113.10	Square meters.
9 GSB 1	254.47	Square miles.

Now let's see how the calculator executed this program.

Searching for a Label

When you switched to RUN mode in the previous example, the calculator was set at step 05, the last step you had filled with an instruction when you were loading the program. When you pressed GSB 1, the calculator began *searching* sequentially downward through program memory, beginning with the current step, for a g LBL 1 instruction. When the calculator searches, it does not execute instructions.

Each of the memory label markers (0 through 9) can be used as many times as you wish. In fact, you can label each of several programs and subroutines with the same number.

In executing the program you recorded, the calculator reached the last step of program memory, step 98, without encountering a g LBL 1 instruction. It then passed step 00 again and continued searching sequentially. Only when the calculator found a g LBL 1 instruction in step 01 did it begin executing instructions. If the label is not found, the calculator will execute no instruction, ***Error*** will be displayed and program memory will be set back to the step at which the search began. Pressing CLx or any key clears the error.

For example: Ensure that the calculator is in RUN mode and key in the following:

Press	Display
GSB 5	*Error*

No g LBL 5 instruction exists in program memory. Now press CLx or any key to clear the error from the display, or you can set the calculator to PRGM mode. The calculator is set to the step at which the search began.

Executing Instructions

When the calculator found the g LBL 1 instruction in step 01, it ceased searching and began *executing* instructions. The calculator executes instructions in exactly the order you keyed them in, performing the g x^2 operation in step 02 first, then g $\sqrt{x}$ as in step 03, etc., until it executes a g RTN instruction or a R/S *(run/stop)* instruction. Since a g RTN instruction is executed in step 05, the calculator stops there and displays the contents of the X-register. (To see the next step number of program memory after the one at which the calculator has stopped, you can briefly switch to PRGM mode.)

If you key in a new value for the radius of a circle in RUN mode and press GSB 1, the calculator repeats this procedure. It searches sequentially downward through program memory until it encounters a g LBL 1 instruction, then sequentially executes the instructions contained in the next steps of program memory until it executes a g RTN or a R/S instruction.

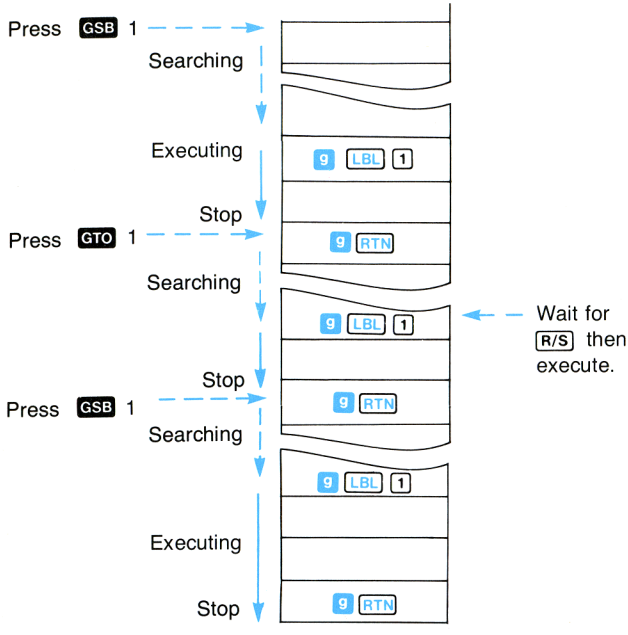
You can see that it is possible to have many different programs or parts of a program loaded in the calculator at any time. You can run any one of these programs by pressing GSB followed by the number key (1 through 9) that corresponds with its label.

It is also possible to have several different programs or routines defined by the same label. For example, suppose you had three programs that were defined by g LBL 1 as illustrated on page 103.

In this example when you press GSB 1, the calculator searches sequentially through program memory from wherever it was located until it encounteres the first g LBL 1 instruction. The calculator then executes instructions until it executes a g RTN or a R/S instruction and stops.

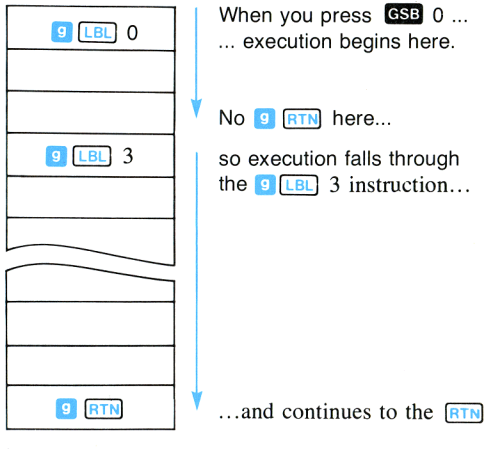
If you had then pressed GTO 1 rather than GSB 1, the calculator would resume searching sequentially from the g RTN or R/S through program memory until it encountered the second g LBL 1, whereupon it would stop. Note that when you search for a label manually from the keyboard using GSB, the calculator will begin execution after it locates the label. But if you search for a label manually from the keyboard using GTO, the calculator will find the label and wait for you to press R/S to begin execution. Refer to Subroutines (page 146) for more information about GTO and GSB.

When you press R/S, the calculator executes all subsequent instructions until it executes a g RTN or a R/S and stops. When you press GSB again, the calculator searches downward to the third g LBL 1 instruction and begins execution there.



Labels and Step 00

The labels (0 through 9) in your programs act as addresses—they tell the calculator where to begin or resume execution. When a label is encountered as part of a program, execution merely "falls through" the label and continues onward. For example, in the program segment shown below, if you press GSB 0, execution would begin at g LBL 0 and continue downward through program memory, on through the g LBL 3 instruction, and continue until the RTN was encountered and execution was stopped.



Execution falls through step 00, too. You can load instructions into steps 01 through 98 of program memory, but you cannot load an instruction into step 00. In fact, step 00 merely acts

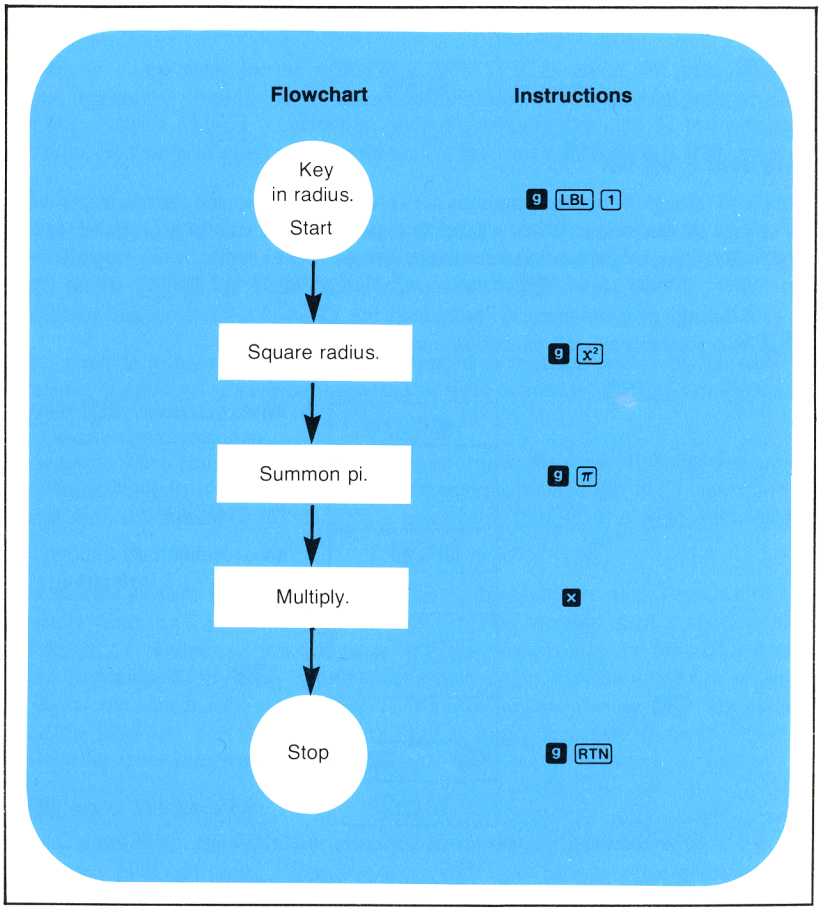
as a kind of label in program memory, a beginning point for the loading of a program. When step 00 is encountered by a running program, execution continues without a halt from step 98 to step 01, just as if step 00 were not there.

Flowcharts

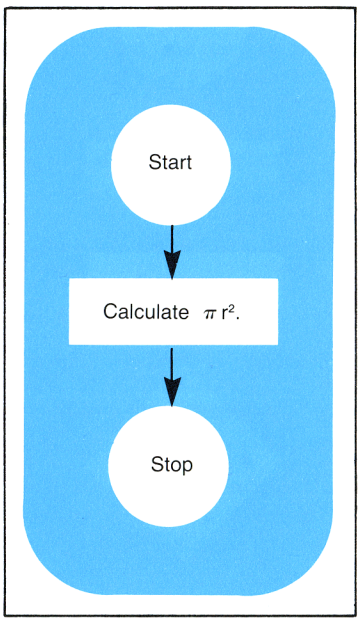
At this point, we digress for a moment from our discussion of the calculator itself to familiarize ourselves with a fundamental and extremely useful tool in programming—the flowchart.

A flowchart is an *outline* of the way a program solves a problem. With 98 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:



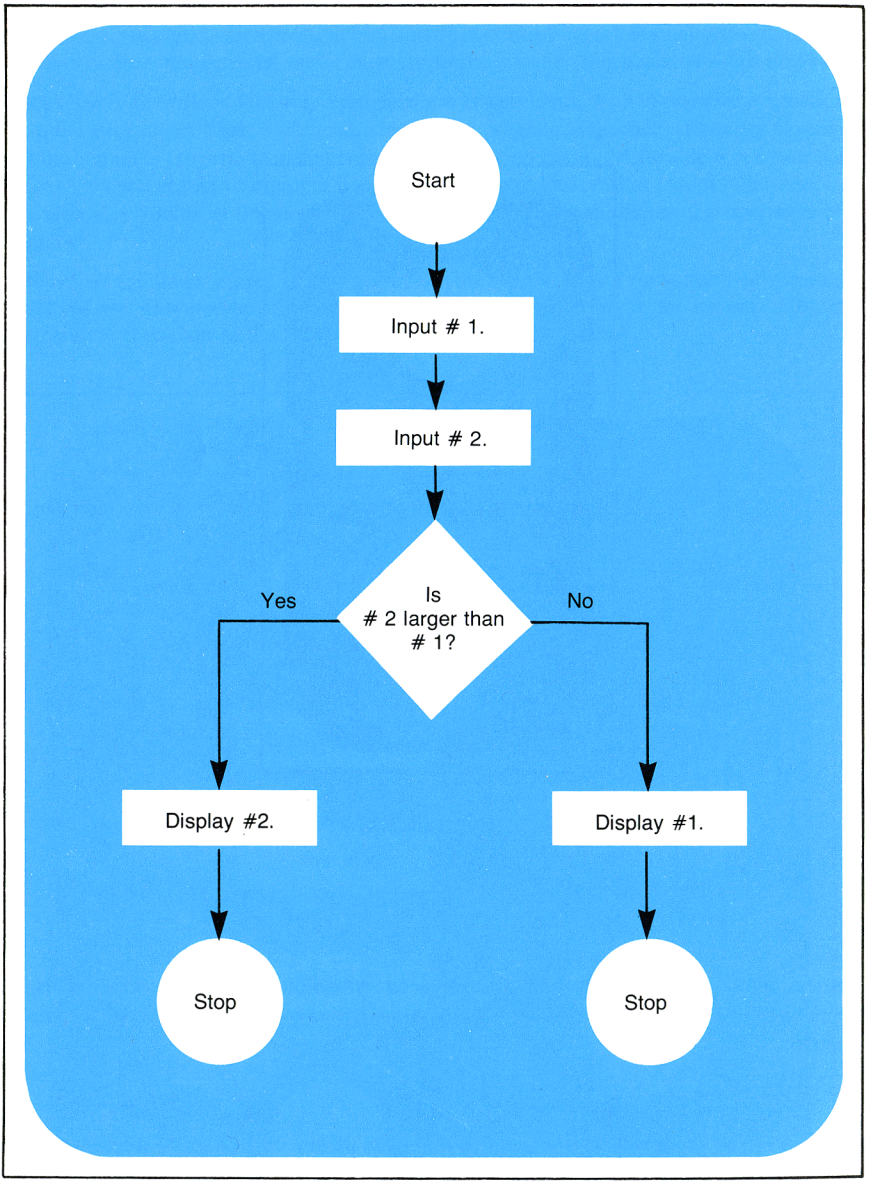
You can see the similarities. At times, a flowchart may duplicate the set of instructions exactly, as shown above. At other times, it may be more useful to have an entire group of instructions represented by a single block in the flowchart. For example, here is another flowchart for the program to calculate 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 that 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 looked like this:



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 calculator.)

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.

Problems

1. You have seen how to write, load, and run a program to calculate the area of a circle from its radius. Now write and load a program that will calculate the radius r of a circle given its area A using the formula $r = \sqrt{A/\pi}$. Be sure to set the calculator to PRGM mode and press f CLEAR PRGM first to clear program memory unless you wish to save the program already in program memory. Define the program with g LBL 2 and g RTN. After you have loaded the program, run it to calculate the radii of circles with areas of 28.27 square inches, 113.10 square meters, and 254.47 square miles. (If you wish, on the HP-19C, include PRX instructions to print the answer.)

   (Answers: ***3.00*** inches, ***6.00*** meters, ***9.00*** miles.)

2. Write and load a program that will convert temperature in Celsius degrees to Fahrenheit, according to the formula F = (1.8 × C) + 32. Define the program with g LBL 3 and g RTN and run it to convert Celsius temperatures of −40°, 0°, and +72°.

   (Answers: ***−40.00*** °F, ***32.00*** °F, ***161.60*** °F.)

3. Immediately after running the program in problem 2, create a program that will convert temperature in degrees Fahrenheit back to Celsius according to the formula C = (F-32)5/9, defining it using g LBL 4 and g RTN, and load it into program memory immediately after the program you loaded in problem 2. Run this new program to convert the temperatures in °F you obtained back to °C.

   If you wrote and loaded the programs as called for in problems 2 and 3, you should now be able to convert any temperature in Celsius to Fahrenheit by pressing GSB 3, and any temperature in Fahrenheit to Celsius by pressing GSB 4. You can see how you can have many different programs loaded into the calculator and select any one of them for running at any time.

The Printer and the Program (HP-19C)

All print functions on the HP-19C (except PRT PRGM) can be recorded as instructions in program memory and later executed as part of a program. In addition, you can use the three modes of the HP-19C printer to aid you in programming.

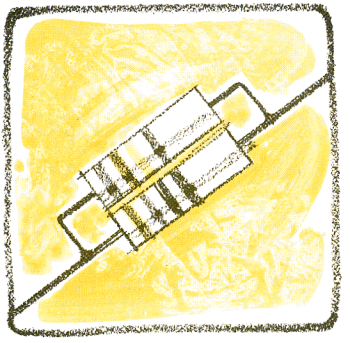
Printer Operation during a Running Program

Like the other keys on your HP-19C, the printer operates in the same natural, normal manner during a running program as it does when you are using the calculator manually. In addition, if you have the Print Mode switch set to TRACE during a running program, the printer will print a mnemonic symbol for each operation as well as printing intermediate and final results calculated during the program.

In most cases, you will probably want the HP-19C to run a program as quickly as possible, and you will not want the running program to have to "slow down" to engage the printer, as it does in the TRACE mode of the Print Mode switch. To ensure the fastest program execution (and greatest battery power conservation), place the Print Mode switch MAN NORM / TRACE in the MAN or NORM position. If you want the HP-19C to print some results during or at the end of a running program, simply place PRX instructions in the program wherever printed results are required.

Using the Printer for Creating Programs

The printer on your HP-19C is a valuable and time-saving tool that you can use not only to record answers and program listings, but also to aid you in creating and editing programs. For example, when creating a program you can use the printer to generate the list of keystrokes that will later form the bulk of your program. With the Print Mode switch MAN NORM / TRACE set to NORM, the printer records a history of the calculations necessary to solve a problem. Then you can simply go back and follow the listing produced by the printer when loading instructions into program memory.



Example: The formula to calculate the total resistance of two parallel resistances in an electrical circuit is:

$$R_t = \frac{R1 \times R2}{R1 + R2}$$

Write a program that will permit you to key in any two parallel resistances and calculate the total resistance.

Solution: Begin by selecting a pair of sample resistances; say, 1500 ohms and 2200 ohms. Then solve for the total resistance using the formula above with the Print Mode switch set to NORM. To solve for parallel resistances of 1500 ohms and 2200 ohms:

Set the OFF-PRGM-RUN switch OFF PRGM / RUN to RUN.

Slide the Print Mode switch MAN NORM / TRACE to NORM.

Press	Display
1500	*1500*
STO 1	*1500.00*
2200	*2200.*
×	*3300000.00*
f LAST X	*2200.00*
RCL 1	*1500.00*
+	*3700.00*
÷	*891.89*
PRX	*891.89*

```
1500.00 STO1
2200.00   ×
        LSTX
        RCL1
          +
          ÷
 691.89   ***
```

As you can see, you have generated a list of keystrokes that solves the problem. Your list should look like this:

```
1500.00 STO1
2200.00   x
        LSTX
        RCL1
          +
          ÷
 691.89  ***
```

Now all you have to do is to add to this list slightly so that it will work for *all* values of R1 and R2. With the Print Mode switch still set to NORM, solve the problem again so that the value for R2 will be stored in register R_1 and the value for R1 will be placed in the LAST X register when the multiplication operation is performed:

```
1500.00 ENT↑
2200.00 STO1
          X≷Y
           x
        LSTX
        RCL1
           +
           ÷
 891.89  ***
```

Enter 1500.
Store 2200 in R_1.
Exchange X and Y.
Multiply 1500 × 2200.
Retreive 1500 from LAST X register.
Recall 2200 from R_1.
Add 2200 and 1500.
Divide 3300000.00 by 3700.
Print value in display.

Now assume that the values for R1 and R2 have been input to the Y- and X-registers, respectively. If the program were defined by g LBL 5, it would look like this:

```
01 *LBL5
02  STO1
03   X≷Y
04    x
05  LSTX
06  RCL1
07    +
08    ÷
09  PRTX
10   RTN
```

Now load the program. The printer is a great aid here, too, as you will see.

Program Load Verification

In PRGM mode, with the Print Mode switch MAN / TRACE / NORM set to TRACE, the printer records the keystrokes you press as you load a program. This gives a printed record of the program *as you key it in,* verifying that you are loading it correctly.

Switch to PRGM and TRACE modes now, and monitor the loading of the program as you press the keystrokes:

First, slide the Print Mode switch MAN / TRACE / NORM to TRACE. Slide the PRGM-RUN switch to PRGM.

Then clear program memory by pressing:

f CLEAR PRGM

Now follow the list of keystrokes to key in the rest of the program.

Press	Display
g LBL 5	01 25 14 05
STO 1	02 45 01
x≷y	03 11
×	04 51
f LAST x	05 16 63
RCL 1	06 55 01
+	07 41
÷	08 61
PRX	09 65
g RTN	10 25 13

```
01 *LBL5
02  STO1
03   X≷Y
04    x
05  LSTX
06  RCL1
07    +
08    ÷
09  PRTX
10   RTN
```

Now switch back to RUN mode and run the program to see that it yields the same answer for your test case:

Slide the OFF-PRGM-RUN switch OFF / PRGM / RUN to RUN.

Slide the Print Mode switch MAN / TRACE / NORM back to NORM (unless you want the printer to trace the operation of the program, too).

Press	Display
1500 ENTER↑	1500.00
2200	2200.
GSB 5	891.89

```
1500.00 ENT↑
2200.00 GSB5
 891.89  ***
```

As you can see, the printer is a great aid in the creation of programs.

Program Listing

Whether the calculator is set to PRGM or RUN, you can list your program.

When you press g PRT PRGM, the printer lists the step number and operation for each step of the program, beginning with the current step and continuing until two R/S instructions or step 98 is encountered.

You can stop the printing of a program at any time by simply pressing any key on the keyboard.

To list the program you currently have in LBL 5:

Press	Display	
GTO 5 g PRT PRGM	*891.89*	Sets calculator to top of program LBL 5.

```
                  GTO5
01 *LBL5 25 14 05
02  STO1     45 01
03   X≷Y        11
04     x        51
05  LSTX     16 63
06  RCL1     55 01
07     +        41
08     ÷        61
09  PRTX        65
10   RTN     25 13
11   R/S        64
```

Note that you positioned the calculator to the beginning of the program LBL 5 using the GTO 5 keys before printing. You can also position the calculator to any *step* (0 through 98) in program memory using the GTO • n n keys.

Printing a Space

If you wish to insert a space between portions of your print-out or between answers, you can use the g SPC *(space)* function. This function advances the paper one space without printing. Digit entry is not terminated by the g SPC function.

For example:

Press	Display	
123	*123.*	
g SPC	*123.*	Paper advances one space.
456	*123456.*	Digit entry was not terminated by printing a space.

From a program the use of g SPC instructions allows you to place as many spaces as you desire in your printed results.

Section 7

Program Editing

Often you may want to alter or add to a program that is loaded in the calculator. On your HP-19C/HP-29C keyboard, you will find several editing functions that permit you to easily change any steps of a loaded program *without* reloading the entire program.

As you may recall, there are six functions plus, on the HP-19C, the PRT PRGM function that cannot be recorded in program memory. All functions and operations can be recorded as instructions in program memory except these seven. These functions are *program editing and manipulation functions,* and they can aid you in altering and correcting your programs.

Nonrecordable Operations

CLEAR PRGM is one keyboard operation that cannot be recorded in program memory. When you press f CLEAR PRGM in PRGM mode, program memory is cleared to R/S instructions and the calculator is reset to the top of memory (step 00) so that the first instruction will be stored in step 01 of program memory. With the calculator set to RUN mode, f CLEAR PRGM merely cancels the f prefix key that you have pressed.

SST *(single step)* is another nonrecordable operation. When you press SST in PRGM mode, the calculator moves to and displays the next step of program memory. When you press SST in RUN mode, the calculator displays the next step of program memory—when you release the SST key, the calculator executes the instruction loaded in that step. SST permits you to single step through a program, executing the program one step at a time or merely viewing each step without execution, as you choose.

BST *(back step)* is a nonrecordable operation that displays the *previous* step of program memory. When you press g BST in PRGM mode, the calculator moves to and displays the previous step of program memory. When you press and release g and then press down BST in RUN mode, the calculator moves to and displays the contents of the previous step of program memory. When you then release BST, the original contents of the X-register are displayed. No instructions are executed.

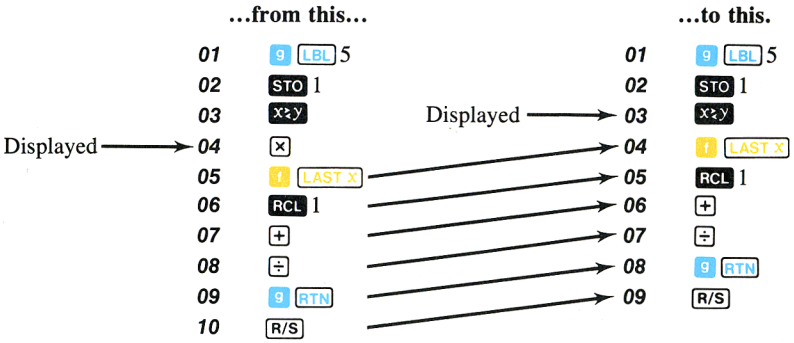
CLEAR PREFIX is a nonrecordable operation used for canceling the f, g, STO, STO •, RCL, RCL •, STO (+ − ÷ ×), STO • (+ − ÷ ×), GTO, GTO •, and GSB keys. If you press f and wish to cancel that prefix key, simply press CLEAR PREFIX. If you press g and wish to cancel that prefix key, simply press f CLEAR PREFIX. All other listed functions can be cleared by pressing f CLEAR PREFIX.

PRT PRGM is an HP-19C printing function used for printing the contents of program memory. When you press g PRT PRGM the printer records the step number, a mnemonic and keycode for each step of the program, beginning with the current step and continuing until two R/S instructions or step 98 is printed.

GTO *(go to)* . n n is another keyboard operation that cannot be loaded as an instruction. (GTO followed by any number, however, *can* be loaded as a program instruction. More about the use of this instruction later.) Whether the calculator is in PRGM or RUN, when you press GTO . followed by a two digit step number, the program memory is set to that step number. No instructions are executed. If the calculator is in RUN mode, you can verify that the calculator is set to the specified step by briefly switching to PRGM mode. The GTO . n n operation is especially useful in PRGM mode because it permits you to jump to any location in program memory for editing of or additions or corrections to your programs.

The DEL *(delete)* key is a nonrecordable operation that you can use to delete instructions from program memory. When the calculator is in PRGM mode and you press g DEL, the instruction at the current step of program memory is erased, and all subsequent instructions in program memory move upward one step. For example, the section of program memory shown below illustrates what would happen when you press g DEL with the calculator set to step 04.

With the calculator set to step 04 when you press g DEL, program memory is changed...



Notice that when a program step is deleted, all keycodes below the deleted step move up one step. The keycodes before the deleted step are moved into the display.

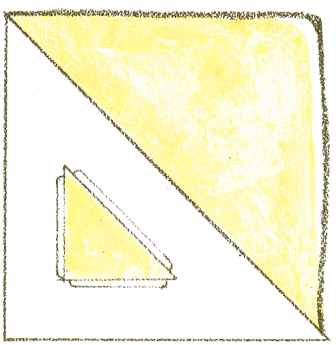
Now let's load a program from the keyboard and use these editing tools to check and modify it.

Pythagorean Theorem Program

The following program computes the hypotenuse of any right triangle, given the other two sides. The formula used is $c = \sqrt{a^2 + b^2}$.

Below are instructions for the program (basically, the same keys you would press to solve for c manually), assuming that values for sides a and b have been input to the X- and Y-registers of the stack.

So that you can concentrate on program displays, set the HP-19C Print Mode switch MAN TRACE NORM to MAN.



To load the program:

First set the calculator to PRGM mode. Then press f CLEAR PRGM to clear program memory of any previous programs and reset the calculator to step 00 of program memory.

Finally, load the program by pressing the keys shown below.

Press	HP-19C	HP-29C
g LBL 9	01 25 14 09	01 15 13 09
g x^2	02 25 53	02 15 63
x⇄y	03 11	03 21
g x^2	04 25 53	04 15 63
+	05 41	05 51
f $\sqrt{x}$	06 16 53	06 14 63
g RTN	07 25 13	07 15 12

With the program loaded into the calculator, you can run the program. For example, calculate the hypotenuse of a right triangle with side *a* of 22 meters and side *b* of 9 meters.

Before you can run the program, you must *initialize* it.

Initializing a Program

Initialization of a program means nothing more than setting up the program (providing inputs, setting display mode, etc.) prior to the actual running of it. Some programs contain initialization routines that set up the data to run the program. In other programs, you may have to initialize manually from the keyboard before running. In the case of the program for calculating the hypotenuse of a triangle, to initialize the program you must place the values for sides *a* and *b* in stack registers X and Y. (Notice that the *order* does not matter in this case.) Thus, to initialize this program:

First, set the calculator to RUN mode.

Press	Display
22 ENTER↑	22.00
9	9.

The program for hypotenuse of a right triangle using the Pythagorean Theorem is now initialized for sides of 22 and 9 meters.

Running the Program

To run the program you have only to press GSB and the number key that selects this program.

Press	Display	
GSB 9	23.77	Length of side *c* in meters.

To compute the hypotenuse of a right triangle with a side a of 73 miles and a side b of 99 miles:

Press	Display	
73 ENTER↑	*73.00*	
99	*99.*	Program initialized for new set of data before running.
GSB 9	*123.00*	Length of side c in miles.

Now let's see how we can use the nonrecordable editing features of the calculator to examine and alter this program.

Resetting to Step 00

As you know, when you press f CLEAR PRGM with the calculator set to PRGM mode, the calculator is reset to step 00 and all instructions in program memory are erased and replaced with R/S instructions. However, you can reset the calculator to step 00 of program memory while *preserving* existing programs in program memory by pressing GTO . 00 in PRGM or RUN mode, or g RTN in RUN mode.

To set the calculator to step 00 with the Pythagorean Theorem program loaded into program memory:

Press	Display	
GTO . 00	*123.00*	Length of side c remains in display from previous running of program.

You could also have pressed g RTN in RUN mode to set the calculator to step 00.

Set the calculator to PRGM mode to verify that the calculator is now set at step 00 of program memory.

Display
00

Single-Step Execution of a Program

With the Program Mode switch set to RUN, you can execute a recorded program one step at a time by pressing the SST *(single-step)* key.

To single-step through the Pythagorean Theorem program using a triangle with side a of 73 miles and side b of 99 miles:

First set the calculator to RUN mode.

Press	Display	
73 ENTER↑	*73.00*	
99	*99.*	Program initialized for this set of data before running.

Now, press SST and hold it down to see the keycode for the next instruction. When you release the SST key, that next instruction is executed.

Press	HP-19C	HP-29C	
SST	*01 25 14 09*	*01 15 13 09*	Keycode for g LBL 9 seen when you hold SST down.
	99.00	*99.00*	g LBL 9 is executed when you release SST.

The first instruction of the program is executed when you press and release SST. (Notice that you didn't have to press GSB 9—when you are executing a program one step at a time, pressing the SST key begins the program from the current step of program memory without the need to press GSB 9.)

Continue executing the program by pressing SST again. When you hold SST down, you see the keycode for the next instruction. When you release SST, that instruction is executed.

Press	HP-19C	HP-29C	
SST	*05 25 53*	*02 15 63*	Keycode for g x^2.
	9801.00	*9801.00*	Executed.

When you press SST a third time in RUN mode, step 03 of program memory is displayed. When you release the SST key, the instruction in that step, $x \rightleftarrows y$, is executed, and the calculator halts.

Press	HP-19C	HP-29C	
SST	*03 11*	*03 21*	Keycode for $x \rightleftarrows y$.
	73.00	*73.00*	Executed.

Continue executing the program by means of the SST key. When you have executed the g RTN instruction in step 07, you have completed executing the program and the answer is displayed, just as if the calculator had executed the program automatically, instead of via the SST key.

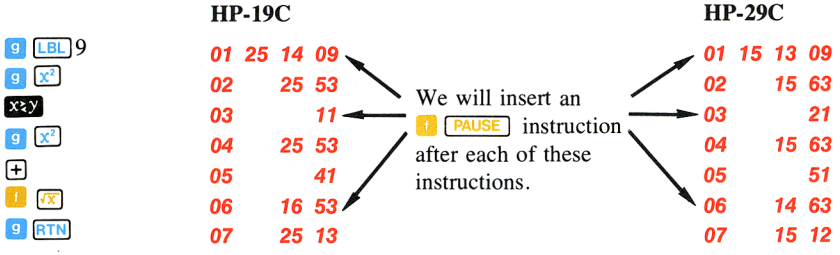
Press	HP-19C	HP-29C
SST	*04 25 53*	*04 15 63*
	5329.00	*5329.00*
SST	*05 41*	*05 51*
	15130.00	*15130.00*
SST	*06 16 53*	*06 14 63*
	123.00	*123.00*
SST	*07 25 13*	*07 15 12*
	123.00	*123.00*

You have seen how the SST key can be used in RUN mode to single-step through a program. Using the SST key in this manner can help you create and correct programs. Now let's see how you can use SST, BST, and GTO . n n in PRGM mode to help you modify a program.

Modifying a Program

Since you have completed execution of the above program, the calculator is set at step 08. You can verify that the calculator is set at this step by setting the calculator to PRGM mode and observing the step number and keycode in the display.

Now let's modify this Pythagorean Theorem program so that the X-register contents will automatically be displayed at certain points in the program. We will do this by inserting the f PAUSE instruction to halt the program and display the contents of the X-register for about 1 second, then resume execution. (More about PAUSE later.)



To begin modification of the loaded program, again reset the calculator to step 00 of program memory without erasing the program:

Ensure that the calculator is set to RUN mode.

Press	Display	
g RTN	*123.00*	Calculator reset to step 00 of program memory.

Single-Step Viewing without Execution

You can use the SST key in PRGM mode to single-step to the desired step of program memory *without* executing the program. When you set the calculator to PRGM mode, you should see that the calculator is reset to step 00 of program memory. When you press SST once, the calculator moves to step 01 and displays the contents of that step of program memory. No instructions are executed.

Set the calculator to PRGM mode.

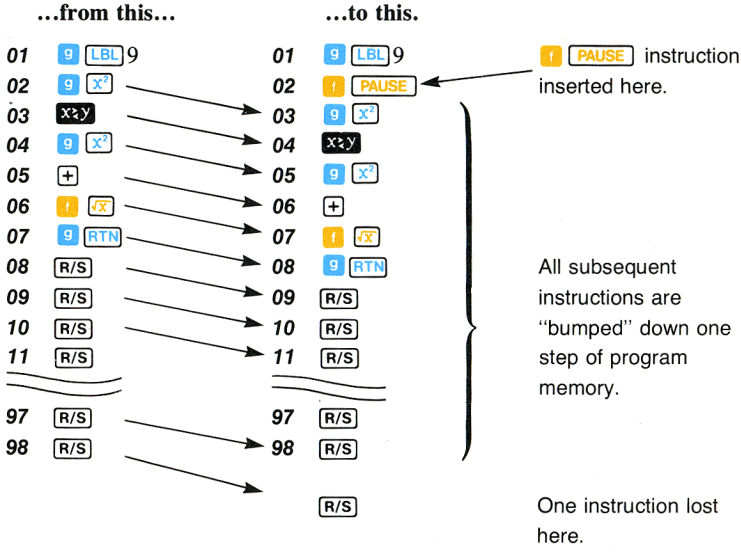
Press	HP-19C	HP-29C	
	00	*00*	Step 00 of program memory.
SST	*01 25 14 09*	*01 15 13 09*	

You can see that the calculator is now set at step 01 of program memory. If you press a *recordable* operation now, it will be loaded in the *next* step, step 02, of program memory, and all subsequent instructions will be "bumped" down one step in program memory.

Thus, to load the f PAUSE instruction so that the calculator will review the contents of the X-register:

Press	HP-19C	HP-29C
f PAUSE	02 16 64	02 14 74

Now let's see what happened in program memory when you loaded that instruction. With the calculator set at step 01, when you pressed f PAUSE program memory was altered...



You can see that when you insert an instruction in a program, all instructions after the one inserted are moved down one step of program memory, and the instruction formerly loaded in step 98 is lost and cannot be recovered. In this case, the last instruction was a R/S instruction and was not used in the program. Note, however, that if you inserted an instruction into program memory when step 98 contained an instruction used in a program, the instruction would be lost from step 98. You should always view the contents of the last few steps of program memory before adding instructions to a program to ensure that no vital instructions will be lost from there.

Going to a Step Number

It is easy to see that if you wanted to single-step from step 00 to some remote step number in program memory, it would take a great deal of time and a number of presses of the SST key. So the calculator gives you another nonrecordable operation, GTO . n n, that permits you to go to *any* step number of program memory.

Whether the calculator is set to PRGM mode or to RUN mode, when you press GTO . n n, the calculator immediately jumps to the program memory step number specified by the two-digit number n n. No instructions are executed. In RUN mode, you can momentarily set the calculator to PRGM mode to view this program information, while if the calculator is already in PRGM mode, the step number and keycode for the instruction contained in that step are displayed. Program searching or execution then will begin with that step of program memory. Loading will begin with the next step of program memory.

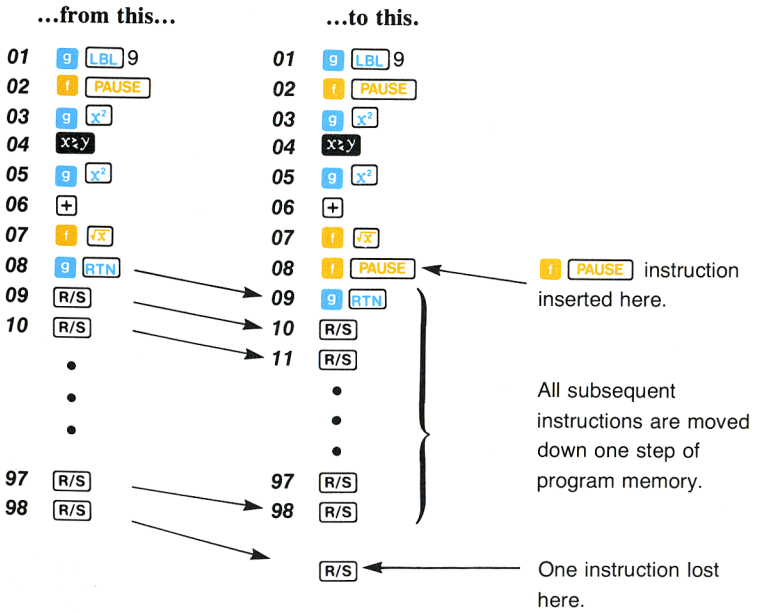
For example, to add an f PAUSE instruction to review the X-register contents after the hypotenuse has been calculated by the instruction in step 07, you can first press GTO *(go to)* followed by a decimal point and the appropriate two digit step number of program memory.

Then press f PAUSE to place that instruction in the *following* step of program memory. Remember that when you add an instruction in this manner, each subsequent instruction is moved down one step in program memory, and the last instruction is lost from step 98. To add the f PAUSE instruction after the g x^2 instruction that is now loaded into step 07:

Press	HP-19C	HP-29C
GTO . 07	*07 16 53*	*07 14 63*
f PAUSE	*08 16 64*	*08 14 74*

As you load the f PAUSE instruction into step 08, the instruction that was *formerly* in step 08 is moved to step 09, and the instructions in subsequent steps are similarly moved down one step. The R/S instruction in step 98 is lost from program memory.

When you added the f PAUSE instruction after step 07, program memory was altered...



Stepping Backwards through a Program

The [BST] *(back step)* key allows you to back step through a loaded program for editing whether the calculator is in RUN or PRGM mode. When you press [g] [BST], the calculator backs up one step in program memory. If the calculator is in RUN mode, the previous step is displayed as long as you hold down the [BST] key. When you release it, the original contents of the X-register are again displayed. In PRGM mode, of course, you can see the step number and keycode of the instruction in the display at all times. No instructions are executed, whether you are in RUN or PRGM mode.

You now have one more [f] [PAUSE] instruction to add to the Pythagorean Theorem program. The [f] [PAUSE] instruction should be added after the [x≷y] instruction, that is now loaded in step 04 of program memory. If you have just completed loading an [f] [PAUSE] instruction in step 08 as described above, the calculator is set at step 08 of program memory. You can use [BST] to back the calculator up to step 04, then insert the [f] [PAUSE] instruction in step 05. To begin:

Ensure that the calculator is set to PRGM mode.

Press	HP-19C	HP-29C	
	08 16 64	08 14 74	Calculator initially set to step 08.
[g] [BST]	07 16 53	07 14 63	Pressing [BST] once moves the calculator back one step in program memory.

When you press [g] [BST], the calculator backs up one step in program memory. No instructions are executed when you use the [BST] key. Continue using the [BST] key to move backward through program memory until the calculator displays step 04.

Press	HP-19C	HP-29C
[g] [BST]	06 41	06 51
[g] [BST]	05 25 53	05 15 63
[g] [BST]	04 11	04 21

Since you wish to insert the [f] [PAUSE] instruction *after* the [x≷y] instruction now loaded in step 04, you move the calculator to step 04 first. As always, when you key in an instruction, it is loaded into the next step after the step being displayed. Thus, if you press [f] [PAUSE] now, that instruction will be loaded into step 05 of program memory, and all subsequent instructions will be moved down, or "bumped," one step.

Press	HP-19C	HP-29C
[f] [PAUSE]	05 16 64	05 14 74

You have now finished modifying the Pythagorean Theorem program so that you can review the contents of the X-register at several points during the running of it. The altered program is shown below:

01	g LBL 9
02	f PAUSE
03	g x^2
04	$x \leftrightarrow y$
05	f PAUSE
06	g x^2
07	+
08	f $\sqrt{x}$
09	f PAUSE
10	g RTN
11	R/S

If you wish, you can use the SST key in PRGM mode to verify that the program in your calculator matches the one shown above. (Refer to Looking at Program Memory, page 95, and The Printer and the Program (HP-19C), page 107.)

Running the Modified Program

To run the Pythagorean Theorem program, you have only to set the calculator to RUN mode, key in the values for sides *a* and *b* and press GSB 9. The calculator will now display the X-register contents, then square side *b*, exchange the contents of the X- and Y-registers, and review the X-register contents again. Finally, the value for the hypotenuse will be calculated, the X-register contents will be reviewed a third time, and the calculated value for the hypotenuse will appear in the X-register when the program stops running.

For example, to compute the hypotenuse of a right triangle with sides *a* and *b* of 22 meters and 9 meters:

Set the calculator to RUN mode.

Press	Display	
22 ENTER↑	*22.00*	
9	*9.*	Program initialized.
GSB 9	*9.00*	After reviewing the X-register contents three times during the running program, the answer in meters is displayed.
	22.00	
	23.77	

Now run the program for a right triangle with sides *a* and *b* of 73 miles and 99 miles.

(Answer: 123 miles)

Deleting Instructions

Often in the modification of a program you may wish to delete an instruction from program memory. To delete the instruction to which the calculator is set, merely press the non-

recordable operation g DEL *(delete)* with the calculator set to PRGM mode. (When the calculator is set to RUN mode, pressing DEL does nothing except cancel a pressed prefix key g.) When you delete an instruction from program memory using the DEL key, all subsequent instructions in program memory are moved *up* one step, and a R/S instruction is loaded into step 98. The calculator moves to the step *before* the deleted step and displays it.

For example, if you wanted to modify the Pythagorean Theorem program that is now loaded into the calculator so that the X-register was only reviewed once, at the end of the program, you would have to delete the f PAUSE instructions that are presently loaded in steps 02 and 05 of program memory. To delete these instructions, you must first set the calculator at these steps using SST, g BST or GTO • n n, then press g DEL. To delete the f PAUSE instruction now loaded in step 02:

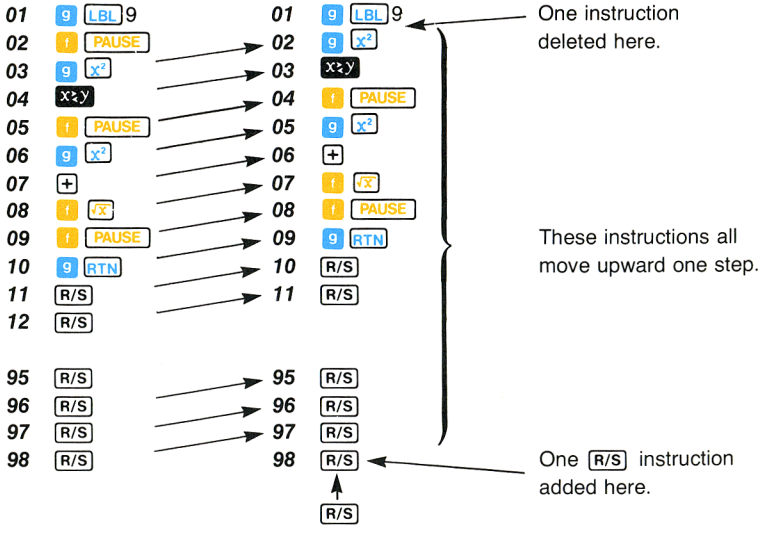
First, set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
GTO •02	*02 16 64*	*02 14 74*	Step 02 is displayed.
g DEL	*01 25 14 09*	*01 15 13 09*	The instruction in step 02 is deleted and the calculator moves to step 01.

You can use the SST key to verify that the f PAUSE instruction has been deleted and subsequent instructions have been moved up one step.

Press	HP-19C	HP-29C	
SST	*02 25 53*	*02 15 63*	The instruction formerly in step 03 was moved up to step 02, and all subsequent instructions were moved up one step when you pressed g DEL.

When you set the calculator to step 02 of program memory and pressed g DEL memory was altered...



To delete the f PAUSE instruction now loaded in step 04 you can use the SST key to single-step down to that step number and then delete the instruction with the g DEL operation.

Press	HP-19C		HP-29C	
SST	03	11	03	21
SST	04	16 64	04	14 74
g DEL	03	11	03	21

The f PAUSE instruction is deleted from step 04 and the calculator displays step 03. Subsequent instructions move up one step of program memory.

If you have modified the program as described above, the calculator should now review the contents of the X-register only once, just before the program stops. The calculated value of the hypotenuse is then displayed.

Set the calculator to RUN mode and run the program for right triangles with:

Sides a and b of 17 and 34 meters. (After reviewing the X-register calculator displays answer for side c, 38.01 meters.)

Sides a and b of 5500 rods and 7395 rods. (After reviewing the X-register calculator displays answer for side c, 9216.07 rods.)

To replace any instruction with another, simply set the calculator to the desired step of program memory, press g DEL to delete the first instruction, then press the keystrokes for the new instruction.

The editing features of the calculator have been designed to provide you with quick and easy access to any part of your program, whether for editing, debugging, or documentation. If a program stops running because of an error or because of an overflow, you can simply set the calculator to PRGM mode to see the step number and keycode of the operation that caused the error or overflow. If you suspect a portion of your program is faulty, you can use the GTO . n n operation from the keyboard to go to the suspect section, then use the SST operation in RUN mode to monitor every change in calculator status as you execute the program one step at a time.

Using the Printer for Editing (HP-19C)

With the HP-19C Print Mode switch set to TRACE, the printer preserves a record of every instruction executed in a running program, as well as all intermediate and final answers. This feature is a valuable aid in debugging and editing programs. To see how the printer reproduces the action of a program, slide the Print Mode switch to TRACE and run the Pythagorean Theorem program to solve for a triangle with sides a and b of 11282 kilometers and 65482.448 kilometers:

Slide the Print Mode switch MAN NORM TRACE to TRACE.

Press	Display	
11282 ENTER↑	*11282.00*	
65482.448	*65482.448*	
GSB 9	*66447.23*	Kilometers.

```
     11282.00 ENT↑
   65482.448 GSB9
01 *LBL9
02    X²
 4287950996.   ***
03   X≠Y
    11282.00   ***
04    X²
 127283524.0   ***
05    +
 4415234520.   ***
06    √X
    66447.23   ***
07   PSE
08   RTN
```

The printer shows every step number, a mnemonic symbol for every instruction executed, and, where calculated, every intermediate and final result.

When a program halts in the middle of execution because of an error or because of an overflow, you can slide the OFF-PRGM-RUN switch to PRGM to see the step number and keycode of the instruction that caused the error or overflow. It may be more helpful, however, to run the program with the Print Mode switch set to TRACE so that you chart the events, step-by-step, that led to the error. With the printer set to the TRACE mode, you can print the operation of the entire program, or, by first addressing the desired beginning step of the program with SST or BST you can print only a portion of the operation of the program if you desire.

You can use the printer in the TRACE mode in conjunction with SST to slow down execution even more. With the Print Mode switch set to TRACE, each time you press the SST key, one instruction is executed and a mnemonic symbol for the instruction and any results are also printed. With this feature, you can examine your programs step-by-step with a fine-toothed comb!

You can also use the printer to verify changes you have made. To print the modified Pythagorean Theorem program now loaded in your calculator:

Press	Display	
GTO 9	*66447.23*	Result from previous problem.
g PRT PRGM	*66447.23*	Prints contents of program 9.

```
                GTO9
01 *LBL9
02    X²
03   X≠Y
04    X²
05    +
06    √X
07   PSE
08   RTN
09   R/S
```

Problems

1. You may have noticed that there is a single keyboard operation, g →P, that calculates the hypotenuse, side c, of a right triangle with sides a and b input to the X- and Y-registers. Replace the x^2, $x \rightleftarrows y$, x^2, +, and $\sqrt{x}$ instructions in the Pythagorean Theorem program with the single g →P instruction as follows:

a. Use the GTO . n n and SST keys to verify that the Pythagorean Theorem program contains the instructions shown below.

```
01  g LBL 9
02  g x²
03  x≷y
04  g x²
05  +
06  f √x
07  f PAUSE
08  g RTN
```

Replace all of these instructions with a g →P instruction.

b. Use the GTO . n n keyboard operation to go to step 06, the last instruction to be deleted in the program.

c. Use the g DEL keyboard operation in PRGM mode to delete the instructions in steps 06, 05, 04, 03, and 02.

Note: When modifying a program, you should always *delete* instructions before you *add* others, to ensure that no vital instructions are "bumped" from the bottom of program memory and lost.

d. Load the g →P instruction into step 02.

e. Verify that the modified program looks like the one below.

```
01  g LBL 9
02  g →P
03  f PAUSE
04  g RTN
```

f. Switch to RUN mode and run the program for a right triangle with sides a and b of 73 feet and 112 feet.

(Answer: 133.69 feet)

2. The following program is used by the manager of a savings and loan company to compute the future amounts of savings accounts according to the formula $FV = PV(1 + i)^n$, where FV is future value or amount, PV is present value, i is the periodic interest rate

expressed as a decimal, and n is the number of periods. With PV entered into the Y-register, n keyed into the X-register, and an annual standard interest rate of 7.5%, the program is:

01	g LBL 6
02	1
03	ENTER↑
04	.
05	0
06	7
07	5
08	+
09	x≷y
10	f y^x
11	×
12	g RTN

a. Load the program into the calculator. On the HP-19C, insert an PRX instruction to print the FV.

b. Run the program to find the future amount of $1,000 invested for 5 years.

(Answer: $1,435.63)

Of $2,300 invested for 4 years.

(Answer: $3,071.58)

c. Alter the program to account for a change of the annual interest rate from 7.5% to 8%.

d. Run the program for the new interest rate to find the future value of $500 invested for 4 years; of $2,000 invested for 10 years.

(Answer: $680.24; $4,317.85)

3. The following program calculates the time it takes for an object to fall to the earth when dropped from a given height. (Friction from the air is not taken into account.) When the program is initialized by keying the height h in meters into the displayed X-register and GSB 0 is pressed, the time t in seconds the object takes to fall to earth is computed according to the formula:

$$t = \sqrt{\frac{2h}{9.8 \text{ meters/second}^2}}$$

a. Clear all previously recorded programs from the calculator and load the program below. On the HP-19C insert PRX instructions to print the height and time.

```
01  g LBL 0
02  2
03  ×
04  9
05  ·
06  8
07  ÷
08  f √x
09  g RTN
```

b. Run the program to compute the time taken by a stone to fall from the top of the Eiffel Tower, 300.51 meters high; from a blimp stationed 1000 meters in the air.

(Answers: 7.83 seconds; 14.29 seconds)

c. Alter the program to compute the time of descent when the height in *feet* is known, according to the formula:

$$t = \sqrt{\frac{2h}{32.1740 \text{ feet/second}^2}}$$

d. Run the altered program to compute the time taken by a stone to fall from the top of the Grand Coulee Dam, 550 feet high; from the 1350-foot height of the World Trade Center buildings in New York City.

(Answers: 5.85 seconds; 9.16 seconds)

Section 8

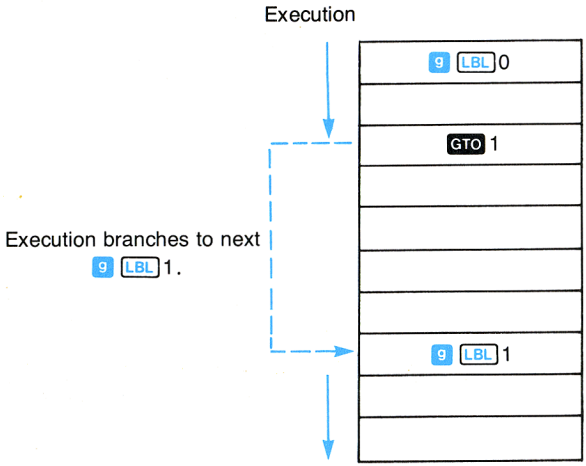
Branching

Unconditional Branching and Looping

You have seen how the nonrecordable operation GTO . n n can be used from the keyboard. to transfer execution to any step number of program memory. You can also use the *go to* instruction as part of a program, but in order for GTO to be *recorded* as an instruction, it must be followed by a label designator (0 through 9). (It can also be followed by the I key—more about this later.)

When the calculator is executing a program and encounters a GTO 1 instruction, for example, it immediately halts execution and begins searching sequentially downward through program memory for that label. When the first g LBL 1 instruction is then encountered, execution begins again.

By using a GTO instruction followed by a label designator in a program, you can transfer execution to any part of the program that you choose.



A GTO instruction used this way is known as an *unconditional branch*. It always *branches* execution from the GTO instruction to the specified label. (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. The calculator 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:

First, set the calculator to PRGM mode.

Press f CLEAR PRGM to clear program memory and reset the calculator to step 00.

Set the HP-19C Print Mode switch MAN TRACE NORM to MAN.

Press	HP-19C	HP-29C	
g LBL 1	01 25 14 01	01 15 13 01	
0	02 00	02 00	
STO 1	03 45 01	03 23 01	
g LBL 4	04 25 14 04	04 15 13 04	
1	05 01	05 01	
STO + 1	06 45 41 01	06 23 51 01	Adds 1 to current number in R_1.
RCL 1	07 55 01	07 24 01	Recalls current number from R_1.
f PAUSE	08 16 64	08 14 74	Displays current number.
f √x	09 16 53	09 14 63	
f PAUSE	10 16 64	10 14 74	Displays square root of current number.
GTO 4	11 14 04	11 13 04	Transfers execution to g LBL 4 again.
g RTN	12 25 13	12 15 12	

To run the program, set the calculator to RUN mode and press GSB 1. The program will begin displaying a table of integers and their square roots and will continue until you press and hold R/S from the keyboard or until the calculator overflows.

How it works: When you press GSB 1, the calculator searches through program memory until it encounters the g LBL 1 instruction that begins the program. It executes that instruction and each subsequent instruction in order until it reaches step 11, the GTO 4 instruction.

The GTO 4 instruction causes the calculator to *search* once again, this time for a g LBL 4 instruction in the program. When it encounters the g LBL 4 instruction loaded in step 04, execution begins again from that g LBL 4. (Notice that the address after a GTO instruction in a program is a *label,* not a step number.)

01 g LBL 1
02 0
03 STO 1
04 g LBL 4
05 1
06 STO + 1
07 RCL 1
08 f PAUSE
09 f √x
10 f PAUSE
11 GTO 4
12 g RTN

Since execution is transferred to the g LBL 4 instruction in step 04 each time the calculator executes the GTO 4 instruction in step 11, 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 the calculator—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 above, or in any part of a program where you wish to transfer execution to another label. When the calculator executes a GTO instruction, it searches sequentially downward through program memory and begins execution again at the first specified label it encounters.

Problems

1. The following program calculates and pauses to display the square of the number 1 each time it is run. Key the program in with the calculator set to PRGM mode, then switch to RUN and run the program a few times to see how it works. Finally, modify the program by inserting an g LBL 7 instruction after the STO 1 instruction in step 03, and a GTO 7 instruction after the second f PAUSE instruction. This should create a loop that will continually display a new number and display its square, then increment the number by 1, display the new number and compute and display *its* square, etc. To load the original program, before modification, set the calculator switch to PRGM mode. Then:

Press	HP-19C	HP-29C
f CLEAR PRGM	00	00
g LBL 4	01 25 14 04	01 15 13 04
0	02 45 01	02 00
STO 1	03 01	03 23 01
1	04 01	04 01
STO + 1	05 45 41 01	05 23 51 01
RCL 1	06 55 01	06 24 01
f PAUSE	07 16 64	07 14 74
g x^2	08 25 53	08 15 63
f PAUSE	09 16 64	04 14 74
g RTN	10 25 13	10 15 12

Note that on the HP-19C you can enter a PRX instruction rather than the f PAUSE to print the table of squares. Set the Print Mode switch MAN TRACE NORM to NORM or MAN.

Run the program to generate a table of squares.

2. Use the flowchart on the following page to create a program that computes and pauses to display (or print on the HP-19C) the future value (FV) of a compound interest savings account in increments of one year according to the formula:

$$FV = PV(1 + i)^n$$

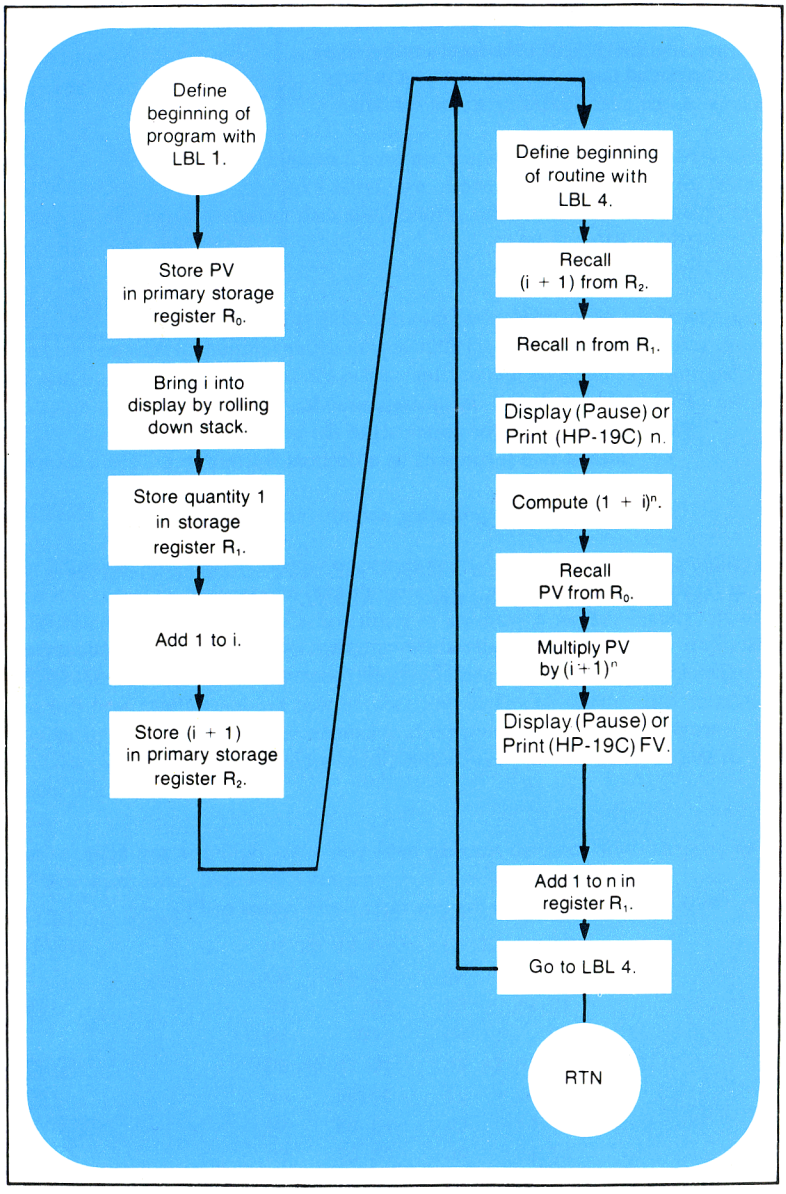
where FV = future value of the savings account.
PV = present value (or principal) of the account.
i = interest rate (expressed as a decimal fraction; e.g., 6% is expressed as 0.06).
n = number of compounding periods (usually, years).

Assume that program execution will begin with *i* entered into the Y-register of the stack and with *PV* keyed into the displayed X-register.

After you have written and loaded the program, run it for an initial interest rate *i* of 6% (keyed in as .06) and an initial deposit (or present value, *PV*) of $1000.

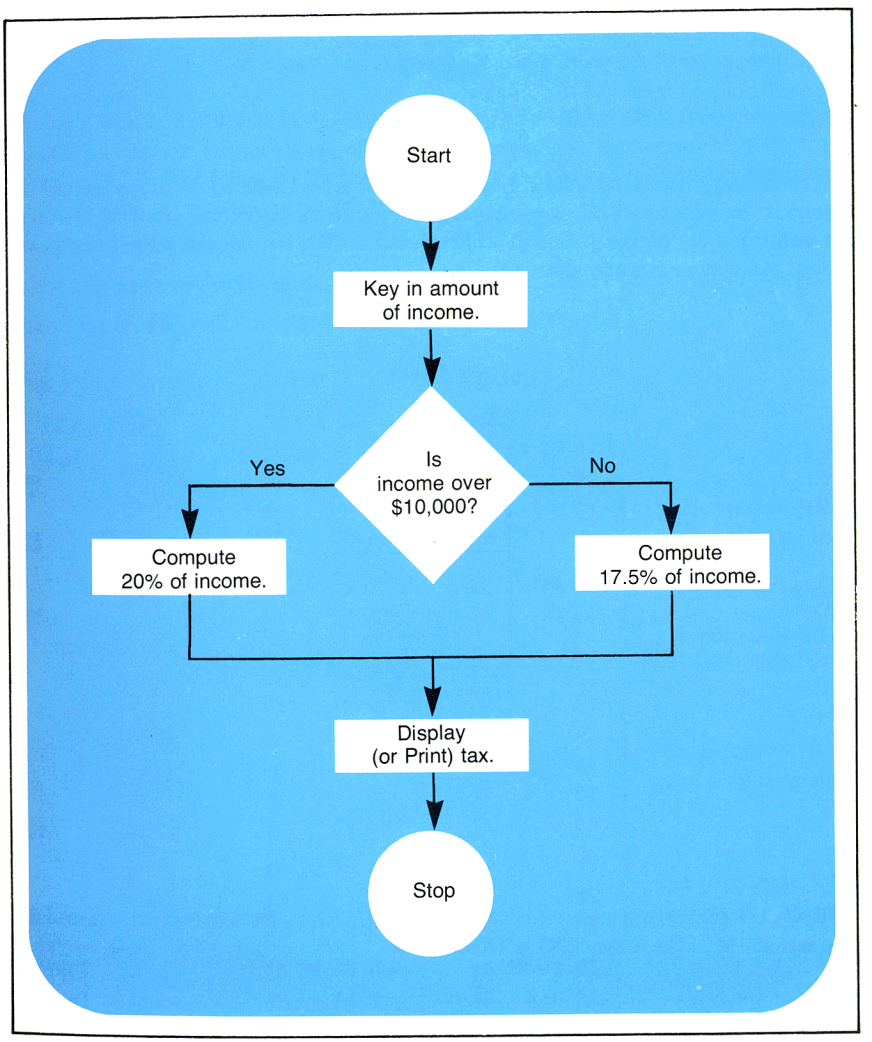
(Answer: 1st year, $1060; 2nd year, $1123.60; 3rd year, $1191.02; etc.)

The program will continue running until you press R/S (or any key), or until the calculator overflows. You can see how your savings would grow from year to year. Try the program for different interest rates *i* and values of *PV*.


Define beginning of program with LBL 1.
Store PV in primary storage register R_0.
Bring i into display by rolling down stack.
Store quantity 1 in storage register R_1.
Add 1 to i.
Store (i + 1) in primary storage register R_2.
Define beginning of routine with LBL 4.
Recall (i + 1) from R_2.
Recall n from R_1.
Display (Pause) or Print (HP-19C) n.
Compute $(1 + i)^n$.
Recall PV from R_0.
Multiply PV by $(i+1)^n$
Display (Pause) or Print (HP-19C) FV.
Add 1 to n in register R_1.
Go to LBL 4.
RTN

Conditionals and Conditional Branches

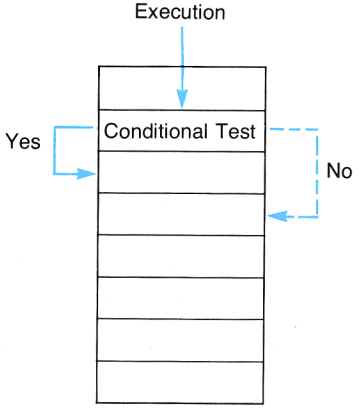
Often there are times when you want a program to make a decision. For example, suppose an accountant wishes to write a program that will calculate and display the amount of tax to be paid by a number of persons. For those with incomes of $10,000 per year or under, the amount of tax is 17.5%. For those with incomes of over $10,000, the tax is 20%. A flowchart for the program might look like this:



The *conditional* operations on your HP-19C/HP-29C keyboard are useful as program instructions to allow your calculator to make decisions like the one shown above. The eight conditionals that are available on your calculator are:

- 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 unequal to 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.
- f [x>y] tests to see if the value in the X-register is greater than the value in the Y-register.
- 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 unequal to zero.
- g [x<0] tests to see if the value in the X-register is less than zero.
- g [x>0] tests to see if the value in the X-register is greater than zero.

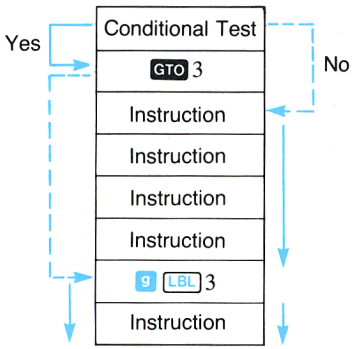
Each conditional essentially asks a question when it is encountered as an instruction in a program. If the answer is YES, program execution continues sequentially downward with the next instruction in program memory. If the answer is NO, the calculator branches *around* the next instruction. For example:



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 step immediately following the conditional test can contain any instruction. The most commonly used instruction, of course, will be a GTO instruction. This will branch program

execution to another section of program memory if the conditional test is true.



Now let's look at that accountant's problem again. For persons with incomes of more than $10,000 he wants to compute a tax of 20%. For persons with incomes of $10,000 or less, the tax is 17.5%. The following program will test the amount in the X-register and compute and display the correct percentage of tax.

To key in the program:

Set the calculator to PRGM mode.

Ensure that the HP-19C Print Mode switch is set to MAN.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 1	01 25 14 01	01 15 13 01	
EEX	02 23	02 33	Amount of $10,000 placed in Y-register.
4	03 04	03 04	
x≷y	04 11	04 21	
f x>y	05 16 41	05 14 51	If amount of income is greater than $10,000, go to portion of program defined by label 2.
GTO 2	06 14 02	06 13 02	
1	07 01	07 01	Tax percentage for this portion of program is 17.5.
7	08 07	08 07	
.	09 63	09 73	
5	10 05	10 05	
GTO 3	11 14 03	11 13 03	
g LBL 2	12 25 14 02	12 15 13 02	Tax percentage for this portion of program is 20.
2	13 02	13 02	
0	14 00	14 00	
g LBL 3	15 25 14 03	15 15 13 03	
g %	16 25 11	16 15 21	
g RTN	17 25 13	17 15 12	

To run the program to compute taxes on incomes of $15,000 and $7,500:
Set the calculator to RUN mode.

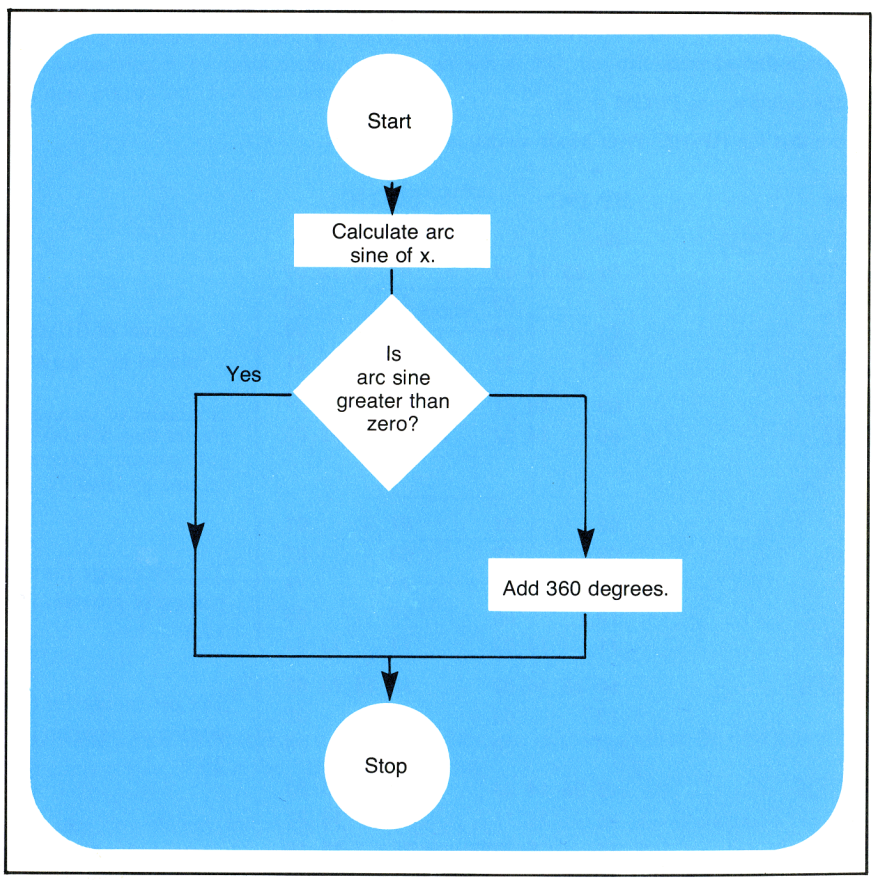
Press	Display	
15000 GSB 1	*3000.00*	Dollars of tax.
7500 GSB 1	*1312.50*	Dollars of tax.

Another place where you often want a program to make a decision is within a loop. The loops that you have seen have to this point 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, forever (or, more practically, until the calculator overflows or you halt the running program by pressing R/S or any other key).

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 a specified number of iterations or when a certain value has been reached within the loop.

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. The program below contains a loop that displays consecutive integers and their common logarithms. You can specify the *lowest* integer by storing a number in storage register R_0, but the program will continue until you press [R/S] or any other key from the keyboard, or until the calculator's capacity for display is exceeded.

01	[g] [LBL] 1
02	[f] [FIX] 9
03	[RCL] 0
04	[f] [INT]
05	[f] [PAUSE]
06	[f] [log]
07	[f] [PAUSE]
08	1
09	[STO] [+] 0
10	[GTO] 1
11	[g] [RTN]

Using the additional instructions [RCL] 8, [x≥y], [GTO] 2 and [LBL] 2, you should be able to modify this program to halt execution when a certain number is reached. As you add these instructions, assume that the value for the upper limit has been manually stored in primary storage register R_8.

When the program is running and the value in register R_0 becomes greater than the limit you store in register R_8, program execution should be transferred out of the loop to the [RTN] instruction to halt the running program.

Modify the program, key it into the calculator, and intialize the calculator by storing a lower limit of 1 in register R_0 and an upper limit of 5 in register R_8. Then run the program. Your displays should look like the ones below. Try other upper and lower limits. (The lower limit must always be greater than zero, and the upper limit should be greater than the lower limit.)

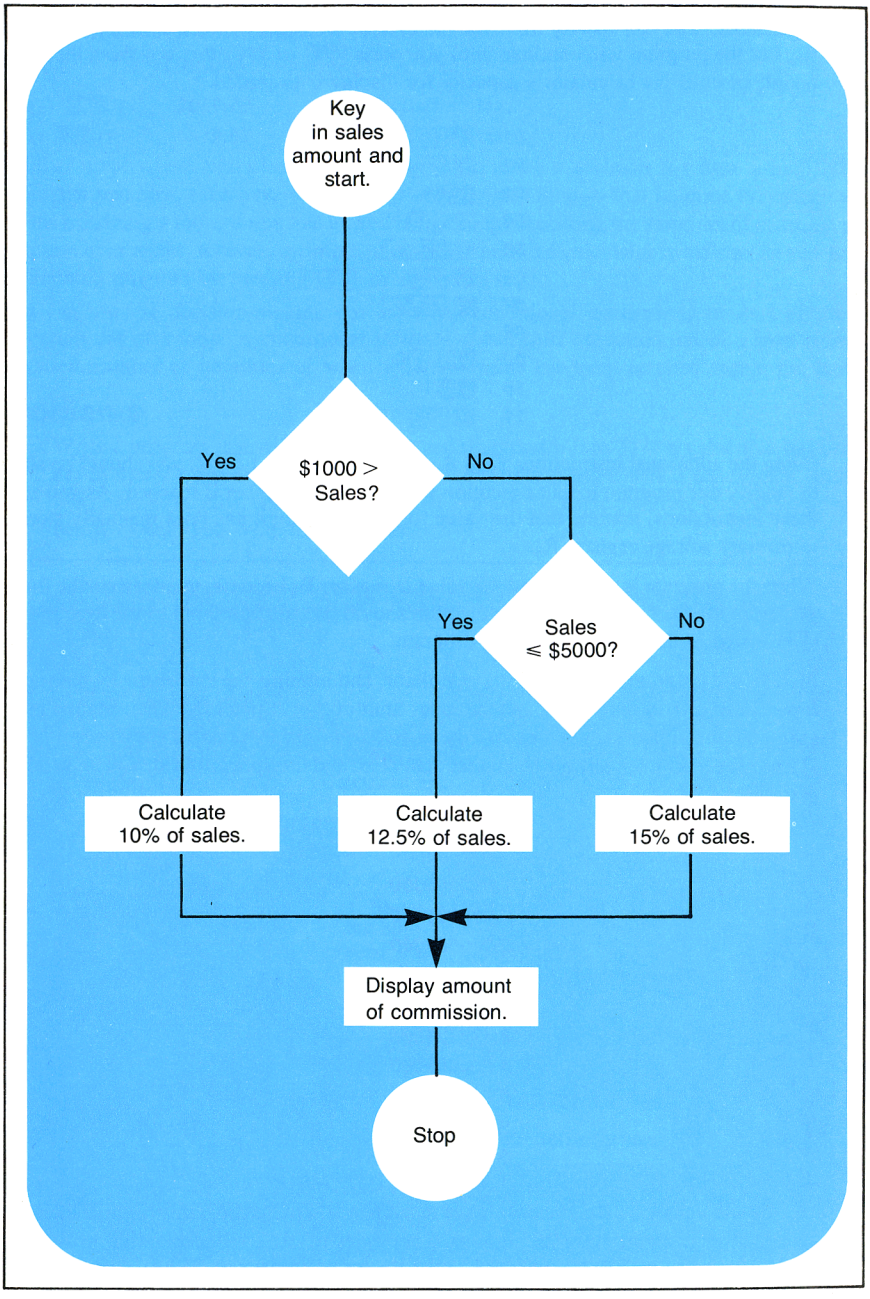
Display

1.000000000
0.000000000
2.000000000
0.301029996
3.000000000
0.477121255
4.000000000
0.602059991
5.000000000
0.698970004

3. Use the flowchart on the opposite page to help you write a program that will allow a salesman to compute his commissions at the rates of 10% for sales of up to $1000, 12.5% for sales of $1000 to $5000, and 15% for sales of over $5000. The program should display the amount of commission when it stops.

Load the program and run it for sales amounts of $500, $1000, $1500, $5000, and $6000.

(Answers: $50.00, $125.00, $187.50, $625.00, $900.00)


Key in sales amount and start.
$1000 > Sales?
Yes
No
Sales ≤ $5000?
Yes
No
Calculate 10% of sales.
Calculate 12.5% of sales.
Calculate 15% of sales.
Display amount of commission.
Stop

Section 9

Program Interruptions

In your programs, there may often be occasions when you want a program to halt during execution so that you can key in data, or to pause so that you can quickly view results before the program automatically resumes running. Besides illustrating the two functions, R/S and PAUSE, that are used for program interruptions, this section also shows you how the keyboard can be used to halt program execution, and how an error will halt a running program.

Using R/S

The R/S *(run/stop)* function can be used either as an instruction in a program or 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 location in program memory.

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

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: The following program lets you key in a percentage discount and calculates the cumulative cost of various quantities of differently priced items from which the discount has been subtracted. A R/S instruction is inserted in the program to allow you to key in data.

To key in the program:

Set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 5	01 25 14 05	01 15 13 05	
f CLEAR REG	02 16 23	02 14 33	
STO 0	03 45 00	03 23 00	Store discount percentage in R_0.
g LBL 9	04 25 14 09	04 15 13 09	
R/S	05 64	05 74	Stop to key in quantity and price.
×	06 51	06 61	
RCL 0	07 55 00	07 24 00	

g %	08		25	11	08		15	21	
−	09			31	09			41	
STO + 1	10	45	41	01	10	23	51	01	Add to running total in R_1.
RCL 1	11		55	01	11		24	01	Recall running total for display.
GTO 9	12		14	09	12		13	09	
g RTN	13		25	13	13		15	12	

Now run the program to calculate the cumulative total of the following purchases at a discount of 15%.

Quantity	Price of Each
5	$ 7.35
7	$12.99
14	$14.95

Then run the program to calculate the cumulative total of the following purchases at a discount of 25%:

Quantity	Price of Each
7	$ 4.99
12	$ 1.88
37	$ 8.50

In order to calculate the cumulative total for each percentage of discount, merely key in the percentage value and press GSB 5. When the calculator stops executing, key in the quantity of an item and press ENTER↑. Then key in the price of that item and press R/S to resume program execution from that point.

To run the program:

Set the calculator to RUN mode.

Press	Display	
15	15.	Key in discount %.
GSB 5	15.00	
5 ENTER↑	5.00	
7.35 R/S	31.24	Running total.
7 ENTER↑	7.00	
12.99 R/S	108.53	Running total.
14 ENTER↑	14.00	
14.95 R/S	286.43	Cost for all items at 15% discount.
25	25.	Percentage of discount.
GSB 5	25.00	
7 ENTER↑	7.00	
4.99 R/S	26.20	Running total.
12 ENTER↑	12.00	
1.88 R/S	43.12	Running total.
37 ENTER↑	37.00	
8.50 R/S	278.99	Cost for all items at 25% discount.

If you have a number of halts for data entries like the ones shown here, it may be helpful to "identify" each step by recording a familiar number into the program immediately before each [R/S] instruction. When the calculator then stops execution because of the [R/S] instruction, you can look at the displayed X-register to see the "identification number" for the required data input at that point. For example, if your program contained stops for data inputs, it might be helpful to have the numbers 1 through 8 appear so that you would know which input was required each time. (Don't forget that the "identification number" will be pushed up into the Y-register of the stack when you key in a new number.)

Pausing to View Output

You now know two instructions that will slow or halt a running program for data output—[PRX] (HP-19C) and [R/S]. [PRX] can print (on the HP-19C) the value contained in the X-register at any point in the program, while [R/S] stops a running program and allows you to view results in the display. For HP-19C printer operation, refer to The Printer and the Program (page 107).

Another instruction that can be used to slow a running program to view output is the [PAUSE] instruction. An [f] [PAUSE] instruction executed in a program momentarily interrupts program execution. The length of the pause is about one second, although more [PAUSE] instructions in subsequent steps of program memory can be used to lengthen viewing time, if desired.

You can use [f] [PAUSE] in a program to monitor the operation of the program without printing every result.

Example: Named after a 13th-century mathematician, the Fibonacci series is a series of numbers that expresses many relationships found in mathematics, architecture, and nature. (For example, in many plants, the proliferation of branches follows a series of Fibonacci numbers.) The series is of the form 0, 1, 1, 2, 3, 5, 8, 13..., where each element is the sum of the two preceding elements.

The following program contains a loop that generates the next Fibonacci element, pauses to display it, then generates still another Fibonacci element and pauses to display *that,* etc. The loop is an infinite one, and the program will run, continually displaying the next Fibonacci element, until you stop the program by pressing [R/S] (or any key) from the keyboard, or until the calculator overflows.

To key in the program:

Set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 2	01 25 14 02	01 15 13 02	
0	02 00	02 00	
STO 0	03 45 00	03 23 00	
1	04 01	04 01	
STO 1	05 45 01	05 23 01	
f PAUSE	06 16 64	06 14 74	
g LBL 0	07 25 14 00	07 15 13 00	
RCL 0	08 55 00	08 24 00	
RCL 1	09 55 01	09 24 01	
+	10 41	10 51	
f PAUSE	11 16 64	11 14 74	
STO 0	12 45 00	12 23 00	Infinite loop.
RCL 0	13 55 00	13 24 00	
RCL 1	14 55 01	14 24 01	
+	15 41	15 51	
f PAUSE	16 16 64	16 14 74	
STO 1	17 45 01	17 23 01	
GTO 0	18 14 00	18 13 00	
g RTN	19 25 13	19 15 12	

Now switch to RUN mode and run the program. Press and hold R/S (or any key) to stop the program after you have seen how quickly the Fibonacci series increases. To run the program:

Set the calculator to RUN mode.

Press	Display
GSB 2	1.00
	1.00
	2.00
	3.00
	5.00
	8.00
	13.00
	21.00
	34.00
	55.00
	89.00
R/S	144.00

Keyboard Stops

As you know, pressing any key from the keyboard during a running program halts that program. The program may halt after any step—if you set the calculator to PRGM mode after a program is halted, you will see the step number and keycode of the next step to be executed.

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. For example, you can resume execution of the Fibonacci series program now:

Press	Display	
[R/S]	*233.00*	The program resumes execution.
	377.00	
[R/S]	*610.00*	The program halts again.

Error Stops

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

Setting the calculator to PRGM mode clears the error, as does pressing any key from the keyboard. (The key function is not executed.) You can then resume program execution, if you wish, by pressing [R/S] from the keyboard in RUN mode.

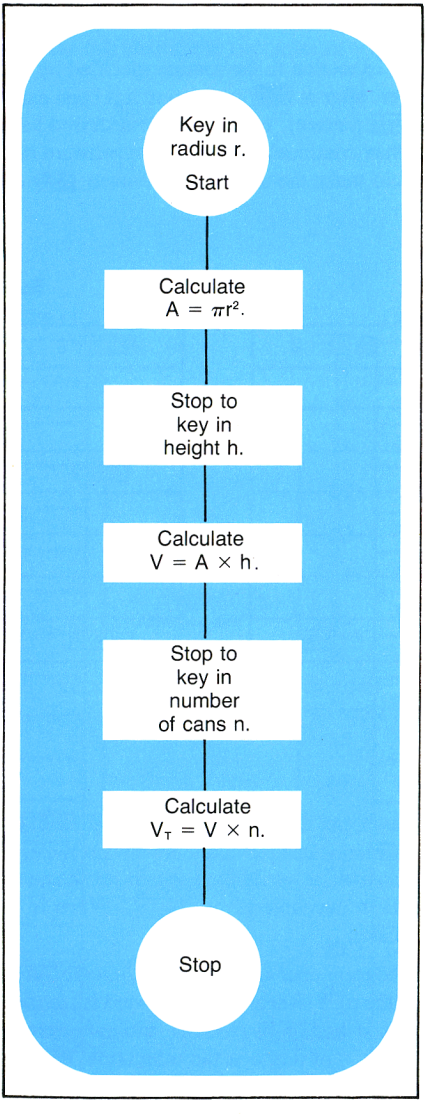
Problems

1. For several different sizes of cans, the foreman at a canning company knows the radius r of the base of the can, the height h of the can, and n, the number of cans of that size. Write a program that will permit the foreman to key in the value for the radius and stop to display the area A of the can. The foreman then keys in the value for the height h, presses [R/S], and the program calculates the volume V according to the formula $V = A \times h$, stopping to display the value for V. Finally the foreman keys in the number of cans n, presses [R/S] again, and the total volume, V_T, is calculated and displayed.

Use the following flowchart to help you write and load the program. Then run the program for 20,000 cans with heights of 25 centimeters and radii of 10 centimeters; for 7500 cans with heights of 8 centimeters and base radii of 4.5 centimeters.

(Answers: $A = 314.16$ cm^2, $V = 7853.98$ cm^3, $V_T = 157079632.7$ cm^3)

(Answers: $A = 63.62$ cm^2, $V = 508.94$ cm^3, $V_T = 3817035.07$ cm^3)

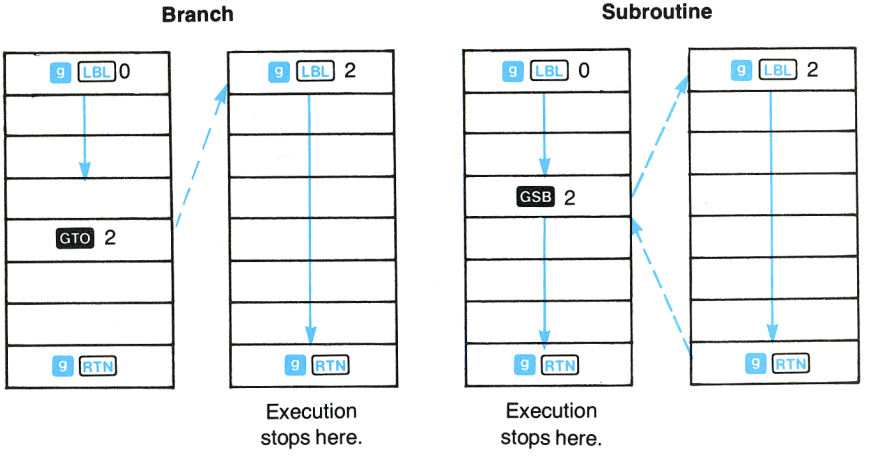


Section 10

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 label address (0 through 9). You can also select a subroutine with the i function—more about that later.

A GSB instruction transfers execution to the routine specified by the label address, just like a GTO instruction. However, after a GSB instruction has been executed, when the running program then executes a g RTN *(return)*, execution is 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 illustration of a branch, on the left, if you pressed GSB 0 from the keyboard, the program would execute instructions sequentially downward through program memory. If it encountered a GTO 2 instruction, it would then search for the next g LBL 2 and continue execution from there, until in encountered a g RTN. When it executed the g RTN instruction, execution would stop.

However, if the running program encounters a GSB 2 *(go to subroutine 2)* instruction, as shown in the illustration on the right, it searches downward for the next g LBL 2 and resumes execution. When it encounters a g RTN *(return)*, program execution is once again transferred, this time back to the point of origin of the subroutine, and execution *resumes* with the next instruction after the GSB 2.

As you can see, the only difference between a subroutine and a normal branch is the transfer of execution *after* the g RTN. After the GTO, the next g RTN halts a running program; after a GSB, the next g RTN returns execution back to the main program, where it continues until another g RTN (or a R/S) is encountered. The same routine may be executed by GTO and GSB any number of times in a program.

Example: A quadratic equation is of the form $ax^2 + bx + c = 0$. Its two roots may be found by the formulas $r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ and $r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$. Notice the similarity between the solutions for r_1 and r_2. The program below permits you to key in the values for a, b, and c in storage registers R_1, R_2 and R_3; the resultant roots r_1 and r_2 are available by pressing GSB 5 and GSB 6.

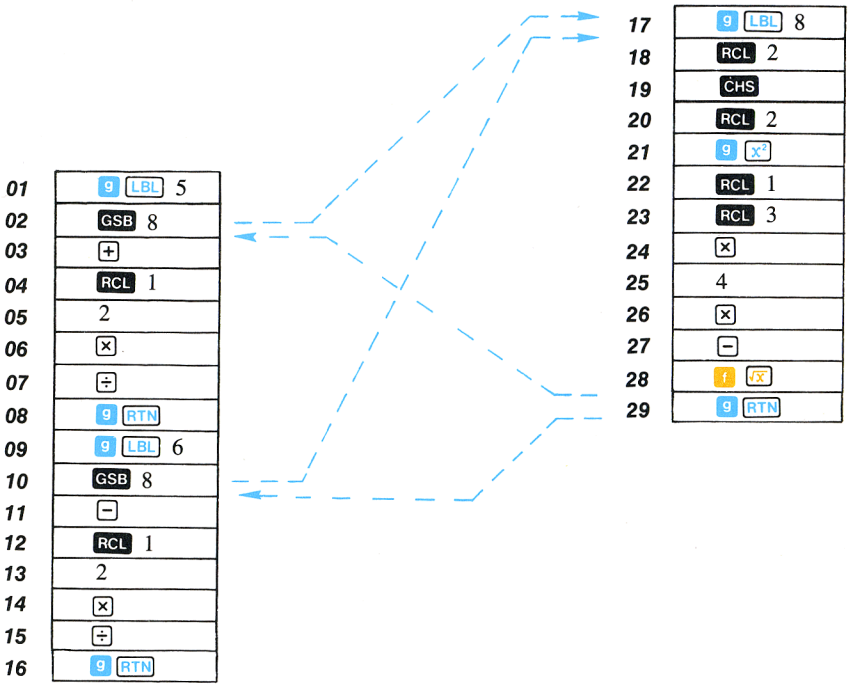
Here is a complete program for calculating the two roots of a quadratic equation:

Input a: STO 1
Input b: STO 2
Input c: STO 3

	Calculate *r*1		Calculate *r*2
01	g LBL 5	19	g LBL 6
02	RCL 2	20	RCL 2
03	CHS	21	CHS
04	RCL 2	22	RCL 2
05	g x^2	23	g x^2
06	RCL 1	24	RCL 1
07	RCL 3	25	RCL 3
08	×	26	×
09	4	27	4
10	×	28	×
11	−	29	−
12	f $\sqrt{x}$	30	f $\sqrt{x}$
13	+	31	−
14	RCL 1	32	RCL 1
15	2	33	2
16	×	34	×
17	÷	35	÷
18	g RTN	36	g RTN

These sections of program memory are identical.

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:



With the modified program, when you press GSB 5, execution begins with the g LBL 5 instruction in step 01. When the GSB 8 instruction in step 02 is encountered, execution transfers to g LBL 8 in step 17 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 step 29 is encountered, execution transfers back to the main routine and continues with the + instruction in step 03. Thus the root r_1 is computed and displayed, and the routine stops with the RTN in step 08.

When you press GSB 6, execution begins with g LBL 6, transfers out to execute the g LBL 8 subroutine, and returns. This time $\sqrt{b^2 - 4ac}$ is *subtracted* from $-b$, and root r_2 is computed. By using a subroutine, seven steps of program memory are saved!

To key in the program and the subroutine:

Set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 5	01 25 14 05	01 15 13 05	Calculates $\frac{-b + \sqrt{b^2 - 4ac}}{2a} = r_1$.
GSB 8	02 13 08	02 12 08	
+	03 41	03 51	
RCL 1	04 55 01	04 24 01	
2	05 02	05 02	
×	06 51	06 61	
÷	07 61	07 71	
g RTN	08	08 15 12	
g LBL 6	09 25 14 06	09 15 13 06	Calculates $\frac{-b - \sqrt{b^2 - 4ac}}{2a} = r_2$.
GSB 8	10 13 08	10 12 08	
−	11 31	11 41	
RCL 1	12 55 01	12 24 01	
2	13 02	13 02	
×	14 51	14 61	
÷	15 61	15 71	
g RTN	16	16 15 12	
g LBL 8	17 25 14 08	17 15 13 08	Subroutine places $-b$ in Y-register and $\sqrt{b^2 - 4ac}$ in X-register, ready for addition or subtraction.
RCL 2	18 55 02	18 24 02	
CHS	19 22	19 32	
RCL 2	20 55 02	20 24 02	
g x^2	21 25 33	21 15 63	
RCL 1	22 55 01	22 24 01	
RCL 3	23 55 03	23 24 03	
×	24 51	24 61	
4	25 04	25 04	
×	26 51	26 61	
−	27 31	27 41	
f $\sqrt{x}$	28 16 53	28 14 63	
g RTN	29	29 15 12	

To initialize the program, you key in a and press STO 1, key in b and press STO 2, and key in c and press STO 3. Then, to find root r_1, press GSB 5. To find root r_2, press GSB 6.

Run the program now to find the roots of the equation $x^2 + x - 6 = 0$; of $3x^2 + 2x - 1 = 0$.

To run the program:

Set the calculator to RUN mode.

Press	Display	
1 STO 1	*1.00*	
1 STO 2	*1.00*	
6 CHS STO 3	*-6.00*	
GSB 5	*2.00*	Calculates the first root, r_1.
GSB 6	*-3.00*	Calculates the second root, r_2.
3 STO 1	*3.00*	
2 STO 2	*2.00*	
1 CHS STO 3	*-1.00*	
GSB 5	*0.33*	Calculates r_1.
GSB 6	*-1.00*	Calculates r_2.

If the quantity $b^2 - 4ac$ is a negative number, the calculator will display ***Error*** and the running program will stop.

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 1 to 6) and then pausing to display the value of the second die (another integer from 1 to 6). Finally the values of the two dice are added together to give the total value.

The "heart" of the program is a random number generator (actually a pseudo random number generator) that is executed first as a subroutine and then as part of the main program. When you key in a first number, called a "seed," and press GSB 1, the digit for the first die is generated and displayed using the g LBL 2 routine as a subroutine. Then the digit for the second die is generated using the same routine as part of the main program. The program then uses the generated number as a new seed for successive "throws" of the dice.

To key in the program:

Set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 1	01 25 14 01	01 15 13 01	
STO 0	02 45 00	02 23 00	
g LBL 0	03 25 14 00	03 15 13 00	
0	04 00	04 00	
STO 1	05 45 01	05 23 01	
GSB 2	06 13 02	06 12 02	g LBL 2 executed first as a subroutine.
g LBL 2	07 25 14 02	07 15 13 02	
RCL 0	08 55 00	08 24 00	
9	09 09	09 09	
9	10 09	10 09	
7	11 07	11 07	
×	12 51	12 61	
g FRAC	13 25 52	13 15 62	
STO 0	14 45 00	14 23 00	
6	15 06	15 06	
×	16 51	16 61	
1	17 01	17 01	g LBL 2 then executed as a routine.
+	18 41	18 51	
f INT	19 16 52	19 14 62	
f FIX 0	20 16 13 00	20 14 11 00	
f PAUSE	21 16 64	21 14 74	
STO + 1	22 45 41 01	22 23 51 01	
RCL 1	23 55 01	23 24 01	
g RTN	24 25 13	24 15 12	
GTO 0	25 14 00	25 13 00	
g RTN	26 25 13	26 15 12	

Now set the calculator to RUN mode and "roll" the dice. To roll the dice, key in the initial decimal "seed" (that is, $0 < n < 1$). Then press GSB 1. The calculator will display first the number rolled by the first die, then the number rolled by the second, and finally, when the program stops, you can see the total number rolled by the dice. To make another roll, press R/S . The program uses the last number as a new seed for the roll.

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 R/S) until the dice again total your point (you win) or you roll a 7 or 11 (you lose). To run the program:

Press	Display	
.2315478	*0.2315478*	The seed.
GSB 1	*10.*	Your point is 10.
R/S	*8.*	You missed your point.
R/S	*5.*	Missed it again.
R/S	*7.*	Woops! You lose.

Now try it again using the last number as the new seed.

Press	Display	
R/S	*8.*	Your point is 8.
R/S	*8.*	Congratulations! You win.

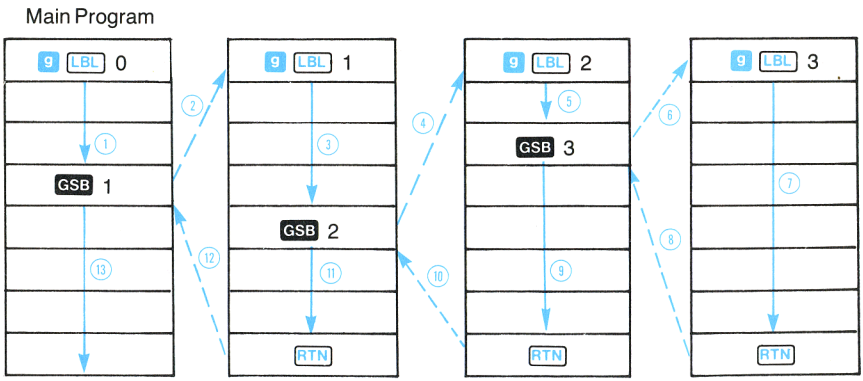
Before you continue, reset the display to two decimal places.

Press	Display
f FIX 2	*8.00*

Subroutine Limits

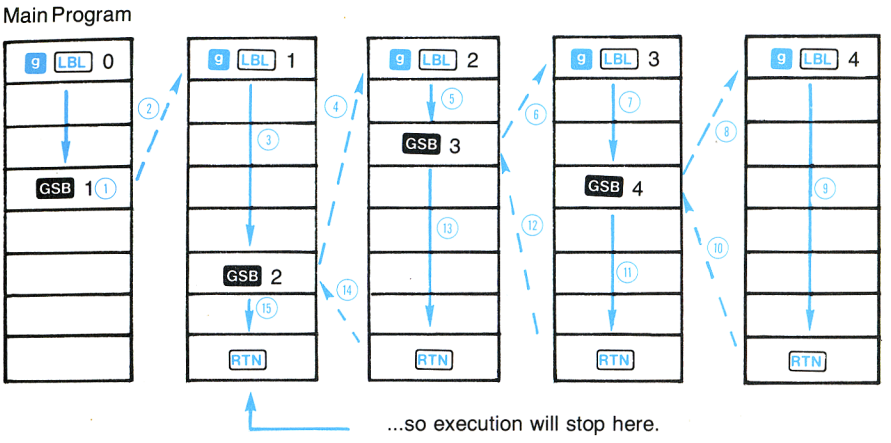
A subroutine can call up another subroutine, and that subroutine can call up yet another. Subroutine branching is limited only by the number of *returns* that can be held pending by the calculator. Three subroutine returns can be held pending at any one time in the HP-19C/ HP-29C. The diagram below should 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...



Naturally, the calculator can execute the RTN instruction as a stop any number of times. Also, if you press GSB 0 through GSB 9 from the *keyboard*, all *pending* RTN instructions are forgotten by the calculator.

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.

Problems

1. Look closely at the program for finding roots r_1 and r_2 of a quadratic equation (page 149). Can you see other instructions that could be replaced by a subroutine? (Hint: look at steps 04 through 08 and steps 12 through 16.) Modify the program by using another subroutine and run it to find the roots of $x^2 + x - 6 = 0$; of $3x^2 + 2x - 1 = 0$.

   (Answers: 2, −3; 0.33, −1)

   Did you save any more steps of program memory?

2. The surface area of a sphere can be calculated according to the equation $A = 4\pi r^2$, where r is the radius. The formula for finding the volume of a sphere is $V = \frac{4\pi r^3}{3}$. This may also be expressed as $V = \frac{r \times A}{3}$.

   Create and load a program to calculate the area A of a sphere given its radius r. Define the program with g LBL 0 and RTN and include an initialization routine to store the value of the radius. Then create and load a second program to calculate the volume V of a sphere, using the equation $V = \frac{r \times A}{3}$. Define this second program with g LBL 2 and RTN, and include the instruction GSB 1 to use a portion of program 1 as a subroutine calculating area.

Run the two programs to find the area and volume of the planet earth, a sphere with a radius of about 3963 miles. Of the earth's moon, a sphere with a radius of about 1080 miles.

Answers: Earth area = 197359487.5 square miles
Earth volume = 2.6071188×10^{11} cubic miles
Moon area = 14657414.69 square miles
Moon volume = 5276669290 cubic miles

3. Create, load, and run a program that will display all permutations of any three integers that you have stored in registers R_1, R_2, and R_3. For example, all permutations of the integers 1, 2, and 3 might be displayed as:

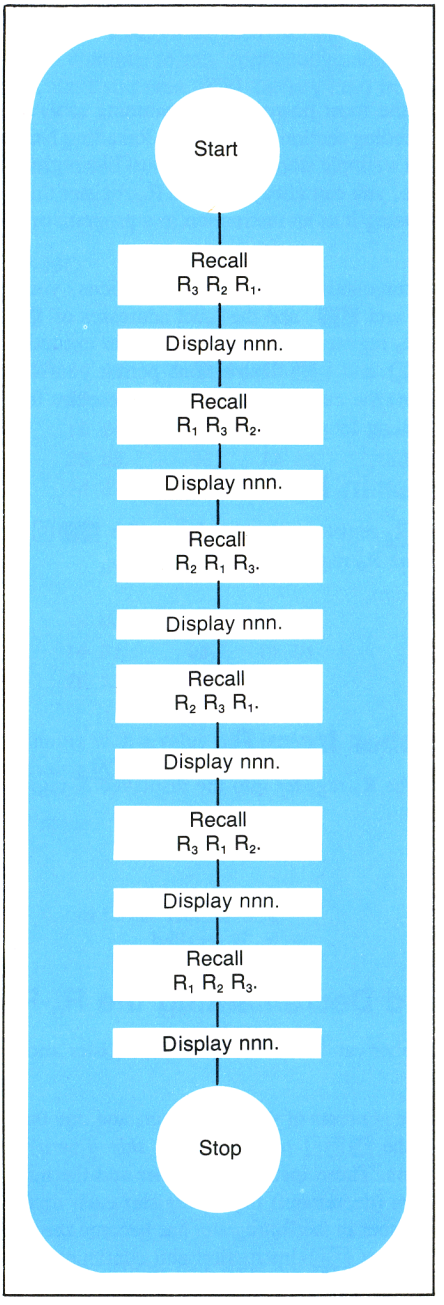
123
132
213
231
312
321

The following subroutine will cause the digits you recall from R_1, R_2, and R_3 to be displayed as a permutation in the order you have recalled them. Use the subroutine and the flowchart on the following page to help you create and load the program.

g LBL 5
1
0
0
×
x≷y
1
0
×
+
+
f PAUSE
g RTN

This subroutine pauses to display numbers recalled into the Z-, Y-, and X-registers of the stack as *nnn*.

The program should recall the contents of storage registers R_1, R_2, and R_3 into the Z-, Y-, and X-registers of the stack and then use the "display *nnn*" subroutine to show them in the order that they are recalled.


Start
Recall
R_3 R_2 R_1.
Display nnn.
Recall
R_1 R_3 R_2.
Display nnn.
Recall
R_2 R_1 R_3.
Display nnn.
Recall
R_2 R_3 R_1.
Display nnn.
Recall
R_3 R_1 R_2.
Display nnn.
Recall
R_1 R_2 R_3.
Display nnn.
Stop

Section 11

Controlling the R_0-Register

The R_0-register is one of the most powerful programming tools available to you on your HP-19C/HP-29C. In a preceding section, Storing and Recalling Numbers, you learned about the use of the R_0-register as a simple storage register, just like registers R_1 through R_9 and $R_{.0}$ through $R_{.5}$. And of course, you can always use the R_0-register this way, as another storage register, whether you are using it as an instruction in a program or operating manually from the keyboard.

Using the R_0-register in conjunction with other instructions, you can specify the storage register addresses of STO and RCL, and the label addresses of GTO and GSB. By storing a negative number in the R_0-register, you can even transfer execution to any step number of program memory. The ISZ and DSZ instructions permit you to increment (add 1 to) or decrement (subtract 1 from) the current value in R_0. These are features that you will find extremely useful in controlling loops.

Storing a Number in R_0

To store a number in the R_0-register, you simply use the STO 0 operation. For example, to store the number 7 in the R_0-register:

Press	Display
7 STO 0	*7.00*

Recalling a Number from R_0

To recall a number from the R_0-register into the displayed X-register, simply use RCL 0:

Press	Display	
CL x	*0.00*	
RCL 0	*7.00*	The number stored in R_0 is recalled.

Incrementing and Decrementing the R_0-Register

You have seen how a number can be stored in the R_0-register and then changed by storing another number there.

Another way of altering the contents of the R_0-register, and one that is most useful during a program, is by means of the g ISZ *(increment R_0, skip if zero)* and g DSZ *(decrement R_0, skip if zero)* instructions. These instructions either add the number 1 to (increment) or subtract the number 1 from (decrement) the R_0-register each time they are executed. In a running program, if the number in the R_0-register has become zero, program execution *skips* the next step after the ISZ or DSZ instruction and continues execution (just like a false conditional instruction).

The g ISZ and g DSZ instructions always increment or decrement first; *then* the test for zero is made. For test purposes, numbers between but not including –1 and +1 are the same as zero.

Example: Here is a program that illustrates how g ISZ works. It contains a loop that pauses to display the current value in the R_0-register, then uses the g ISZ instruction to increment that value. The program will continue to run, continually adding one to and displaying the contents of the R_0-register, until you press R/S (or any key) from the keyboard.

To key in the program:

Set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 1	01 25 14 01	01 15 13 01	
RCL 0	02 55 00	02 24 00	Recalls R_0-register contents.
f PAUSE	03 16 64	03 14 74	Pauses to display contents.
g ISZ	04 25 55	04 15 24	Adds 1 to R_0-register.
GTO 1	05 14 01	05 13 01	If contents of R_0-register are not zero, execution transfers back to g LBL 1.
1	06 01	06 01	If contents of R_0-register are zero, 1 is placed in R_0-register.
STO 0	07 45 00	07 23 00	
GTO 1	08 14 01	08 13 01	
g RTN	09 25 13	09 15 12	

Now run the program beginning with a value of 0 in the R_0-register. Stop the program after five iterations or so by pressing R/S .

Set the calculator to RUN mode.

Press	Display	
0 STO 0	0.00	Zero stored in R_0-register.
GSB 1	0.00	
	1.00	
	2.00	
	3.00	
	4.00	
R/S	5.00	

Although the ISZ and DSZ instructions increment and decrement the R_0-register by 1, the value of the R_0-register need not be a whole number.

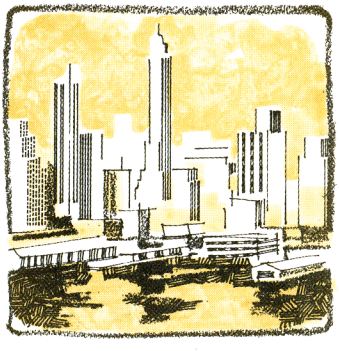
For example:

Press	Display
5.28 CHS	-5.28
STO 0	-5.28
GSB 1	-5.28
	-4.28
	-3.28
	-2.28
	-1.28
R/S	1.00

In practice, you will find that you will usually use ISZ and DSZ with numbers that are integers, since these instructions are most useful as counters—that is, to control the number of iterations of a loop—and to select storage registers, or subroutines. (More about using the R_0-register as a selection register later.)

The DSZ *(decrement R_0, skip if zero)* instruction operates in the same manner as the increment instruction, except that it subtracts, rather than adds, one each time it is used. When a running program executes a g DSZ instruction, for example, it subtracts 1 from the contents of the R_0-register, then tests to see if the R_0-register is 0. (A number between +1 and −1 tests as zero.) If the number in the R_0-register is greater than zero, execution continues with the next step of program memory. If the number in the R_0-register is *zero,* the calculator skips one step of program memory before resuming execution.

Example: The island of Manhattan was sold in the year 1624 for $24.00. The program on the next page shows how the amount would have grown each year if the original amount had been placed in a bank account drawing 5% interest compounded annually. The number of years for which you want to see the amount is stored in the R_0-register, then the DSZ instruction is used to keep track of the number of iterations through the loop.



To key in the program:

Set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 0	01 25 14 00	01 15 13 00	
STO 0	02 45 00	02 23 00	
1	03 01	03 01	
6	04 06	04 06	
2	05 02	05 02	
4	06 04	06 04	Initialization routine.
STO 1	07 45 01	07 23 01	
2	08 02	08 02	
4	09 04	09 04	
STO 2	10 45 02	10 23 02	
g RTN	11 25 13	11 15 12	
g LBL 1	12 25 14 01	12 15 13 01	
RCL 2	13 55 02	13 24 02	
5	14 05	14 05	
g %	15 25 11	15 15 21	Counting loop, controlled by R_0-register and DSZ.
STO + 2	16 45 41 02	16 23 51 02	
1	17 01	17 01	
STO + 1	18 45 41 01	18 23 51 01	
g DSZ	19 25 45	19 15 23	
GTO 1	20 14 01	20 13 01	
RCL 1	21 55 01	21 24 01	◄ When value in R_0 becomes
f FIX 0	22 16 13 00	22 14 11 00	zero, execution skips to here,
f PAUSE	23 16 64	23 14 74	and year and amount are
RCL 2	24 55 02	24 24 02	displayed.
f FIX 2	25 16 13 02	25 14 11 02	
f PAUSE	26 16 64	26 14 74	
g RTN	27 25 13	27 15 12	

To run the program, key in the number of years for which you want to see the amount. Press GSB 0 to store the number of years in the R_0-register and otherwise initialize the program. Then press GSB 1 to run the program.

For example, to run the program to find the amount of the account after 5 years; after 15 years:

Set the calculator to RUN mode.

Press	Display	
5 GSB 0		Program initialized.
GSB 1		

After five years, in 1629, the account would have been worth $30.63.

15 GSB 0
GSB 1

Program initialized.

After 15 years, in 1639, the account would have been worth $49.89.

How it works: When you key in the number of years and initialize the program by pressing GSB 0, the number of years is stored in the R_0-register by the STO 0 instructions. The year (1624) is stored in storage register R_1, and the amount ($24.00) is stored in storage register R_2.

When you then press GSB 1, calculation begins. Each time through the loop, 5% of the amount is computed and added to the amount in R_2, and one (1) year is added to the year in R_1. The DSZ instruction subtracts one from the R_0-register; if the value in R_0 is not then zero, execution is transferred back to g LBL 1 and the loop is executed again.

The loop continues to be executed until the value in the R_0-register becomes zero. Then execution skips to the RCL 1 instruction in program memory step 21. Execution continues sequentially downward from step 21, recalling the current year from R_1 and formatting and displaying it, then recalling the current amount from R_2 and formatting and displaying that following the year.

To see what the amount in the account would be in 1977, you can key in the number of years from 1624 to 1977 (the number is 353) and initialize and run the program. (This will take 4-5 minutes to run, plenty of time to go get a cup of coffee.)

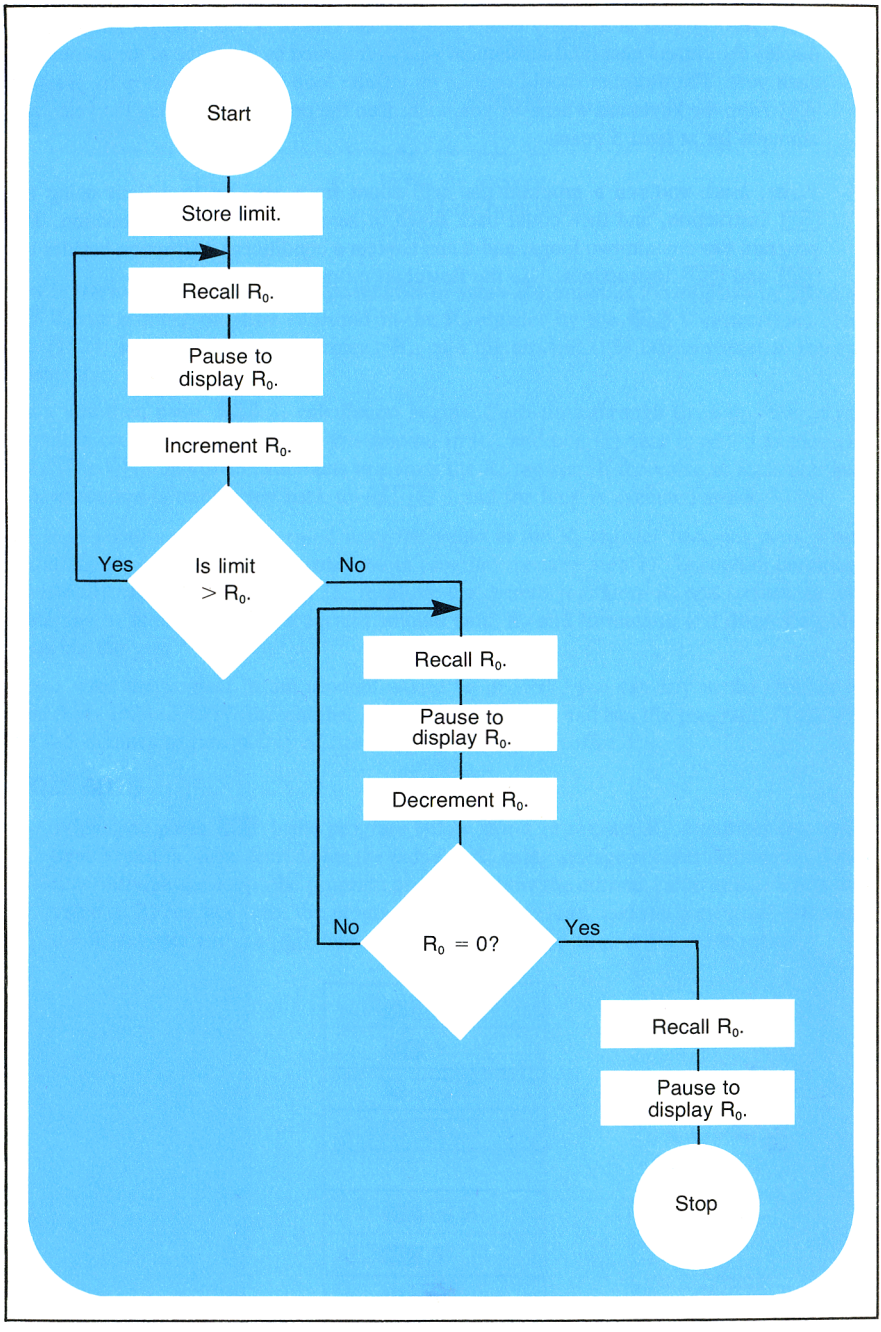
Problems

1. When you press GSB 1 the program below stores in register R_9 a number that you have keyed in, then decrements the value in R_9 using storage register arithmetic. Each time through the loop, the program pauses to show the current value in R_9. When the value in R_9 reaches zero, the program stops. Write, load, and run a program that uses the R_0-register and g DSZ instead of R_9 and g X≠0 to give the same results.

g LBL 1
STO 9
g LBL 2
f PAUSE
1
STO − 9
RCL 9
g X≠0
GTO 2
g RTN

2. Write and load a program using [ISZ] to illustrate how an initial deposit of $1000 would grow year-by-year at a yearly compound interest rate of 5.5%. The program should display the current year (and subsequent years), followed by the value of the account for each year. The program should contain an infinite loop that you can stop by pressing [R/S] from the keyboard whenever you wish. Run the program to display the years and amounts for at least 5 years.

3. Write, load, and run a program that will count from zero *up* to a limit using the [ISZ] instruction, and then count back down to zero using the [DSZ] instruction. The program can contain two loops, and it can contain a conditional instruction besides the [ISZ] and [DSZ] instructions. Use the flowchart on page 162 to help you.


Start
Store limit.
Recall R_0.
Pause to display R_0.
Increment R_0.
Is limit $> R_0$.
Yes
No
Recall R_0.
Pause to display R_0.
Decrement R_0.
$R_0 = 0$?
No
Yes
Recall R_0.
Pause to display R_0.
Stop

Section 12

Using the R_0-Register for Indirect Control

You have seen how the value in the R_0-register can be altered using the STO, ISZ and DSZ operations. But the value contained in the R_0-register can also be used to *control* other operations. The i *(indirect)* function combined with certain other functions allows you to control those functions using the current number in the R_0-register. i uses the number stored in the R_0-register as an *address*.

The indirect operations that can be controlled by the R_0-register are:

STO i, when the number in the R_0-register is 0 through 29, stores the value that is in the display in the primary or indirect storage register addressed by the integer portion of the absolute value of the current number in the R_0-register.

RCL i, when the number in the R_0-register is 0 through 29, recalls the contents of the primary or indirect storage register addressed by the current number in the R_0-register.

STO + i, STO − i, STO × i, and STO ÷ i, when the number in the R_0-register is 0 through 29, perform storage register arithmetic upon the contents of the primary or indirect storage register addressed by the current number in the R_0-register.

GTO i, when the number in the R_0-register is 0 or a positive 1 through 9, transfers execution of a running program sequentially downward through program memory to the next label specified by the current number in the R_0-register.

GTO i, when the number in the R_0-register is a negative number between −1 and −99, transfers execution of a running program *back* in program memory the number of steps specified by the current negative number in the R_0-register.

GSB i, when the number in the R_0-register is 0 through 9, transfers execution of a running program to the subroutine specified by the current number in the R_0-register. Like a normal subroutine, when a RTN is then encountered, execution transfers and continues with the step following the GSB i.

GSB i, when the number in the R_0-register is a negative number between −1 and −99, transfers execution of a running program *back* in program memory the number of steps specified by the current negative number in the R_0-register. Execution from that point is like a normal subroutine, so if a RTN instruction is then encountered, execution is transferred once again, this time to the next instruction after the GSB i.

Note that you can use the i key with the above functions with or without using the g prefix key. That is, pressing STO i is the same as pressing STO g i.

If the number in the R_0-register is outside the specified limits when the calculator attempts to execute one of these operations, the display will show ***Error.*** When using i, the calculator uses for an address only the integer portion of the number currently stored in the R_0-register. Thus, 25.99998785 stored in the R_0-register retains its full value there, but when used as address i, it is read as 25 by the calculator.

In all cases using the [i] *(indirect)* function, the calculator looks at only the integer portion of the current number stored in the R_0-register.

You can already see that using the R_0-register and [i] in conjunction with these other functions gives you a tremendous amount of computing power and exceptional programming control. Now let's have a closer look at these operations.

Indirect Store and Recall

You can use the number in the R_0-register to address the 30 storage registers that are in your calculator. When you press STO [i], the value that is in the display is stored in the storage register addressed by the number in the R_0-register. RCL [i] addresses the storage registers in a like manner, as do the storage register arithmetic operations STO [+] [i], STO [−] [i], STO [×] [i], and STO [÷] [i]. (If you have forgotten the normal operation of the storage registers, or of storage register arithmetic, go back and review section 4, Storing and Recalling Numbers, in this handbook.)

When using STO [i], RCL [i], or any of the storage register arithmetic operations utilizing the [i] function, the R_0-register can contain numbers positive or negative from 0 through 29. The numbers 0 through 15 address primary storage registers R_0 through R_9, $R._0$ through $R._5$, while numbers from 16 through 29 will address indirect storage registers $R_{(16)}$ through R_{29}. Notice that with the number 0 in the R_0-register, [i] addresses the R_0 register itself!

The following diagram should illustrate these addresses more clearly. Notice that the indirect registers (R_{16} through R_{29}) can *only* be used indirectly. That is, you must use the R_0-register in conjunction with the [i] key to use these registers.

Primary Registers

	[i] Address
R_0	0
R_1	1
R_2	2
R_3	3
R_4	4
R_5	5
R_6	6
R_7	7
R_8	8
R_9	9
$R._0$	10
$R._1$	11
$R._2$	12
$R._3$	13
$R._4$	14
$R._5$	15

Indirect Registers

	[i] Address
$R_{(16)}$	16
$R_{(17)}$	17
$R_{(18)}$	18
$R_{(19)}$	19
$R_{(20)}$	20
$R_{(21)}$	21
$R_{(22)}$	22
$R_{(23)}$	23
$R_{(24)}$	24
$R_{(25)}$	25
$R_{(26)}$	26
$R_{(27)}$	27
$R_{(28)}$	28
$R_{(29)}$	29

By using the calculator manually, you can easily see how STO (i) and RCL (i) are used in conjunction with the R_0-register to address the different storage registers:

Set the calculator to RUN mode.

Press	Display	
CL x f FIX 2		
f CLEAR REG		Clears all storage registers to zero.
5 STO 0		Stores the number 5 in the R_0-register.
1.23 STO (i)		Stores the number 1.23 in the storage register addressed by the number in R_0—that is, storage register R_5.
24 STO 0		This number stored in the R_0-register.
85083 STO (i)		This number stored in the indirect storage register ($R_{(24)}$) addressed by the current number (24) in R_0.
12 STO 0		Stores the number 12 in the R_0-register.
77 EEX 43 STO (i)		Stores the number 7.7×10^{44} in the storage register addressed by the number in R_0—that is, in storage register $R_{.2}$.

To recall numbers that are stored in any primary register, you can use the RCL *(recall)* key followed by the number key of the register address. To recall numbers that are stored in the indirect storage registers, the address of the desired indirect register must be stored in the R_0-register. The contents can then be recalled by simply pressing RCL (i). Note that numbers that are stored in *any* of the registers (primary or indirect) can be recalled using the indirect address of the desired register and the (i) key.

For example:

Press	Display	
12 STO 0		Store the indirect address (12) of the desired register ($R_{.2}$) in the R_0-register.

RCL (i)	Recall the number in the primary storage register with the indirect address of 12.
CL x	
RCL (•) 2	Directly recall the number in $R_{.2}$.

By changing the number in the R_0-register, you change the address specified by STO (i) or RCL (i). For example:

Press	Display	
24 STO 0		Store the indirect address of the desired register.
RCL (i)		Recall the number in indirect register $R_{(24)}$.
5 STO 0		
RCL (i)		Recall the number in primary register R_5.

Storage register arithmetic is performed upon the contents of the register addressed by R_0 by using STO (+) (i), STO (−) (i), STO (×) (i), and STO (÷) (i). Again, you can access any storage register, primary or indirect using the R_0-register for addressing. For example:

Press	Display	
1 STO (+) (i)		One added to number in storage register (R_5) currently addressed by the 0-register.
RCL (i)		
2 STO (×) (i)		
RCL (i)		
CL x		
RCL 5		

Naturally, the most effective use of the R_0-register as an address for STO and RCL is in a program.

Example: The following program uses a loop to place the number representing its address in storage registers R_0 through R_9, $R_{.0}$ through $R_{.5}$, and $R_{(16)}$ through $R_{(29)}$. During each iteration through the loop, program execution pauses to show the current value of R_0. When R_0 reaches zero, execution is finally transferred out of the loop by the g DSZ instruction and the program stops.

To key in the program:

Set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 1	01 25 14 01	01 15 13 01	
f CLEAR REG	02 16 23	02 14 33	
2	03 02	03 02	Program initialized.
9	04 09	04 09	
STO 0	05 45 00	05 23 00	
g LBL 2	06 25 14 02	06 15 13 02	Current value in R_0 stored in
RCL 0	07 55 00	07 24 00	storage register addressed by
STO (i)	08 45 12	08 23 22	(i).
f PAUSE	09 16 64	09 14 74	Pause to display current value of R_0.
g DSZ	10 25 45	10 15 23	Subtract one from value in R_0-register.
GTO 2	11 14 02	11 13 02	If $R_0 \neq 0$, execute loop again.
g RTN	12 25 13	12 15 12	

When the program is run, it begins by clearing the storage registers and placing 29 in the R_0-register. Then execution begins, recalling the current value in the R_0-register and storing that number in the corresponding address—for example, when the R_0-register contains the number 17, that number is recalled and stored in the indirect storage register ($R_{(17)}$) that is addressed by the number 17. Each time through the loop, the number in the R_0-register is decremented, and the result is used both as data and as an address by the STO (i) instruction. When the number in the R_0-register reaches zero, execution transfers out of the loop and the program stops.

To run the program:

Set the calculator to RUN mode.

Press	Display
GSB 1	*29.00*
	28.00
	etc.
	1.00

Notice that the contents of the R_0-register have been decremented to zero.

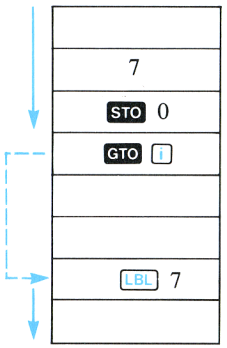
Press	Display
RCL (i)	*0.00*

Indirect Control of Branches and Subroutines

Like addressing of storage registers using STO (i) and RCL (i), you can address routines, subroutines, even entire programs, with the R_0-register.

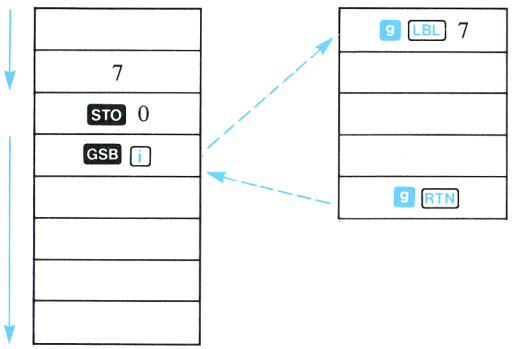
To address a routine using the R_0-register, use the instruction GTO (i). When a running program encounters a GTO (i) instruction, execution is transferred sequentially downward to

the LBL (0 through 9) that is addressed by the number in the R_0-register. Thus, with the number 7 stored in R_0, when the instruction GTO i is encountered, execution is transferred downward in program memory to the next LBL 7 instruction before resuming.



Naturally, you can also press GTO i from the keyboard to begin execution from the specified LBL.

Subroutines can also be addressed and utilized with the R_0-register. When GSB i is executed in a running program (or pressed from the keyboard), execution transfers to the specified LBL and executes the subroutine. When a RTN is then encountered, execution transfers back to the next instruction after the GSB i and resumes. For example, with the number 7 stored in the R_0-register, GSB i causes execution of the subroutine defined by LBL 7 and RTN.



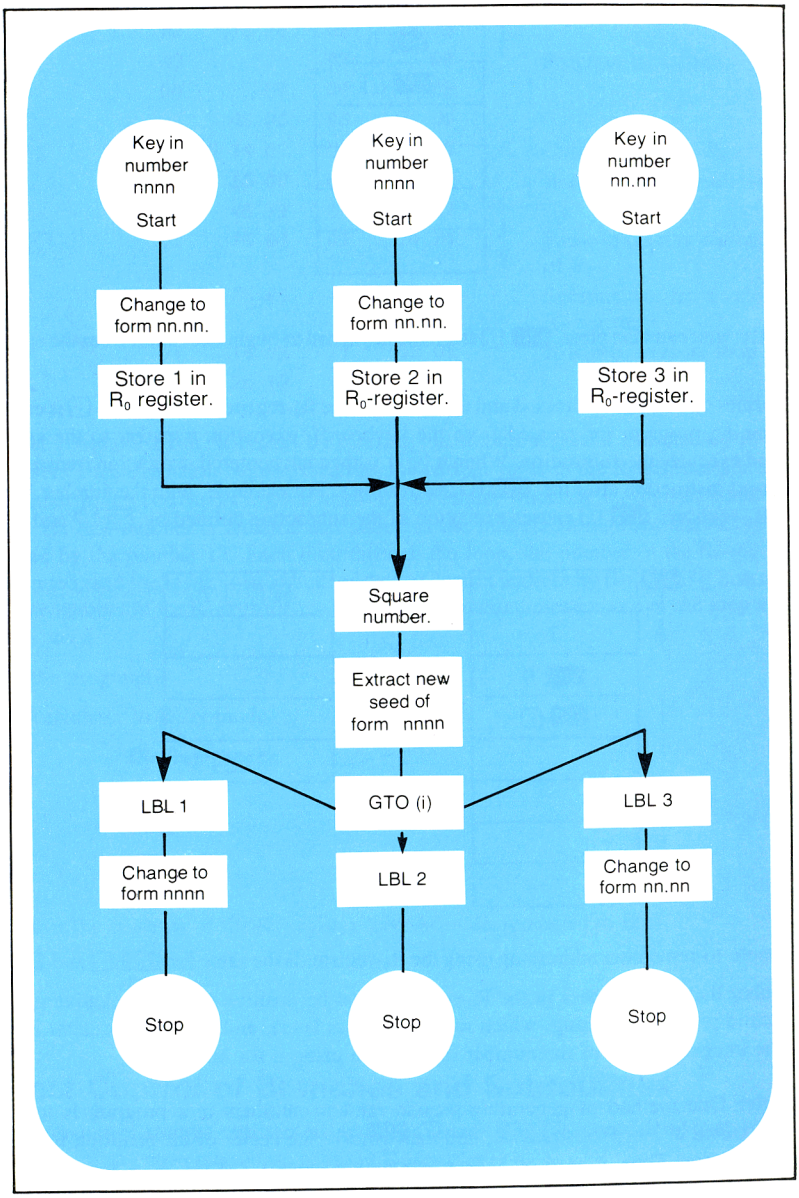
The simple-to-remember addressing using the R_0-register is the same for GTO i and GSB i.

Remember that the numbers in the R_0-register must be positive or zero (negative numbers cause rapid reverse branching, which we will discuss later), and that the calculator looks at only the integer portion of the number in R_0 when using it for an address.

Example: One method of generating pseudo random numbers in a program is to take a number (called a "seed"), square it, and then remove the center of the resulting square and square *that,* etc. Thus, a seed of 5182 when squared yields 26853124. A random number generator could then extract the four center digits, 8531, and square that value. Continuing for several iterations through a loop would generate several random numbers.

The following program uses the GTO [i] instruction to permit you to key in a four-digit seed in any of three forms: *nnnn*, *.nnnn*, or *nn.nn*. The seed is squared and the square truncated by the main part of the program, and the resulting four-digit random number is displayed in the form of the original seed: *nnnn*, *.nnnn*, or *nn.nn*.

A flowchart for the program might look like this:



The use of the GTO [i] instruction lets you select the operations that are performed upon the number after the main portion of the program.

By storing 1, 2, or 3 in the R_0-register depending upon the format of the seed, the program selects the form of the result after it is generated by the main portion of the program. Although the program shown here stops after each result, it would be a simple matter to create a loop that would iterate several times, increasing the apparent randomness of the result each time.

To key in the complete program:

Set the calculator to PRGM mode.

Press	HP-19C	HP-29C	
f CLEAR PRGM	00	00	
g LBL 4	01 25 14 04	01 15 13 04	
EEX	02 23	02 33	Changes *nnnn* to *nn.nn*.
2	03 02	03 02	
÷	04 61	04 71	
1	05 01	05 01	Places 1 in X-register for storage in R_0.
GTO 7	06 14 07	06 13 07	
g LBL 5	07 25 14 05	07 15 13 05	
EEX	08 23	08 33	Changes *.nnnn* to *nn.nn*.
2	09 02	09 02	
×	10 51	10 61	
2	11 02	11 02	Places 2 in X-register for storage in R_0.
GTO 7	12 14 07	12 13 07	
g LBL 6	13 25 14 06	13 15 13 06	
3	14 03	14 03	Places 3 in X-register for storage in R_0.
g LBL 7	15 25 14 07	15 15 13 07	
STO 0	16 45 00	16 23 00	Stores address of later operation in R_0.
x≷y	17 11	17 21	Brings *nn.nn* to X-register.
g x²	18 25 53	18 15 63	Squares *nn.nn*.
EEX	19 23	19 33	
2	20 02	20 02	Truncates two final digits of square.
×	21 51	21 61	
f INT	22 16 52	22 14 62	
EEX	23 23	23 33	
4	24 04	24 04	Truncates two leading digits of square.
÷	25 61	25 71	
g FRAC	26 25 52	26 15 62	
GTO [i]	27 14 12	27 13 22	Transfers execution to appropriate operational routine.

g LBL 1	28 25 14 01	28 15 13 01		
EEX	29 23	29 33		
4	30 04	30 04	Result appears as *nnnn*.	
×	31 51	31 61		
f FIX 0	32 16 13 00	32 14 11 00		
g RTN	33 25 13	33 15 12		
g LBL 2	34 25 14 02	34 15 13 02		
f FIX 4	35 16 13 04	35 14 11 04	Result appears as *.nnnn*.	
g RTN	36 25 13	36 15 12		
g LBL 3	37 25 14 03	37 15 13 03		
EEX	38 23	38 33		
2	39 02	39 02	Result appears as *nn.nn*.	
×	40 51	40 61		
f FIX 2	41 16 13 02	41 14 11 02		
g RTN	42 25 13	42 15 12		

We could also have stored the digits for 100 (that is, EEX 2) and recalled them for use in steps 02-03, 08-09, 19-20, and 38-39, but we have used this more straightforward program to illustrate the use of the GTO i instruction.

When you key in a four-digit seed number in one of the three formats shown, an address (1, 2, or 3) is placed in the R_0-register. This address is used by the GTO i instruction in step 27 to transfer program execution to the proper routine so that the new random number is seen in the same form as the original seed.

Now run the program for seeds of 5182, .5182 and 51.82. To run the program:

Set the calculator to RUN mode.

Press	Display	
5182 GSB 4	*8531.*	Random number generated in the proper form.
.5182 GSB 5	*0.8531*	
51.82 GSB 6	*85.31*	

The program generates a random number of the same form as the seed you keyed in. To use the random number as a new seed (simulating the operation of an actual random number generator, in which a loop would be used to decrease the apparent predictability of each succeeding number), continue pressing GSB and the appropriate label key:

Press	Display
GSB 6	*77.79*
GSB 6	*51.28*
GSB 6	*29.63*

With a few slight modifications of the program, you could have used a GSB i instruction instead of the GTO i instruction.

Rapid Reverse Branching

Using GTO [i] and GSB [i], with a negative number stored in R_0, you can actually branch to any *step number* of program memory.

As you know, when a GTO or GSB instruction is executed, the calculator does not execute further instructions until it has searched downward through program memory and located the next *label* addressed by GTO or GSB. When GTO [i] or GSB [i] is executed in a running program, with 0 or a positive 1 through 9 stored in the R_0-register, the running program searches downward through program memory until it locates the next LBL addressed by the number in R_0. Then execution resumes.

With a *negative* number stored in the R_0-register, however, execution is actually transferred *backward* in program memory when GTO [i] or GSB [i] is executed. The calculator does not search for a label, but instead transfers execution *backward* the number of steps specified by the negative number in the R_0-register. (This is advantageous because the search is often much faster than searching for a label, and because you can thus transfer execution even though all labels in the calculator have been used for other purposes.)

For example, in the section of program memory shown below, −11 is stored in the R_0-register. Then, when step 87, GTO [i] is executed, the running program jumps backward 11 steps through program memory to step 76 (that is, step 87 − 11 = 76) and execution resumes again with step 76 of program memory.

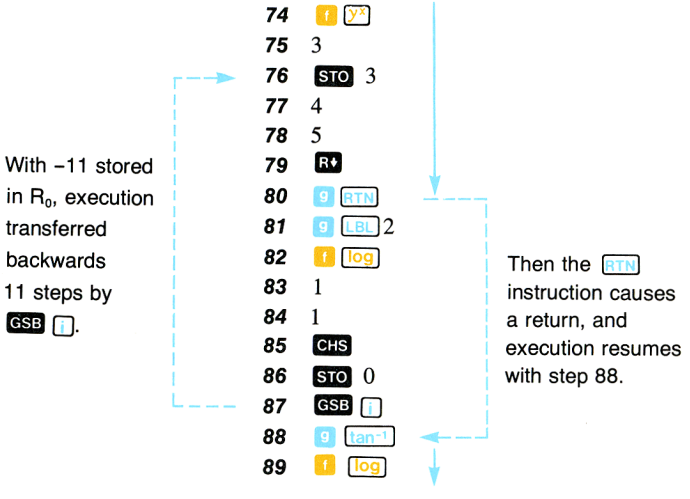
With −11 stored in R_0, execution transferred backwards 11 steps by GTO [i].

74	f y^x
75	3
76	STO 3
77	4
78	5
79	R↓
80	g RTN
81	g LBL 2
82	f log
83	1
84	1
85	CHS
86	STO 0
87	GTO [i]
88	g $\tan^{-1}$

When GTO [i] has been performed in a running program, execution then continues until the next RTN or R/S instruction is encountered, whereupon the running program stops. Thus, if you pressed GSB 2 with the instructions shown above loaded into the calculator, the instructions in steps 81 through 87 would be executed in order. Then the program would jump backward and execute step 76 next, continuing with 77, 78, etc., until the RTN instruction was encountered in step 80. The running program would then stop.

With a negative number stored in the R_0-register, GSB [i] also transfers execution backward the number of steps specified by the number in R_0. However, subsequent instructions are then executed as a *subroutine,* so when the next RTN instruction is encountered, execution transfers back to the instruction following the GSB [i] instruction (just like a normal subroutine would be executed.)

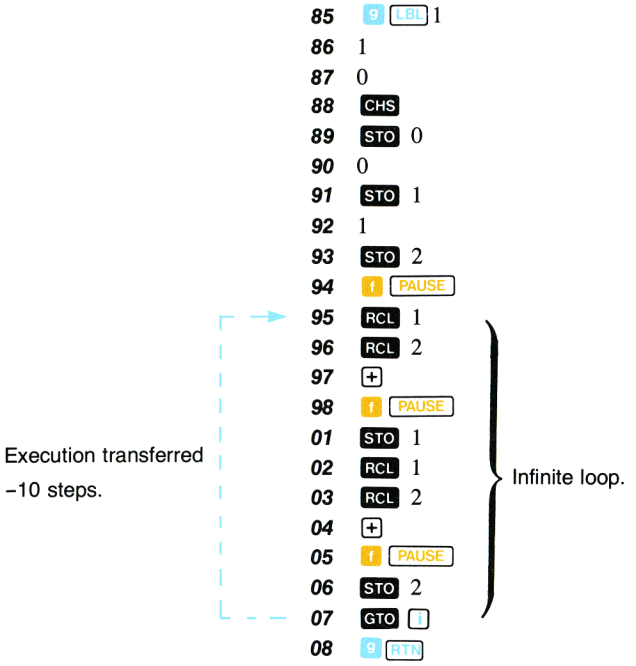
The section of program memory below shows how GSB [i] operates. If you press GSB 2, −11 will be stored in the R_0-register. When GSB [i] is then executed a running program jumps back 11 steps from step 87 and resumes execution with step 76. When the RTN *(return)* instruction in step 80 is encountered, execution returns and continues with step 88.



Rapid reverse branching using GTO [i] and GSB [i] are extremely useful instructions as part of your programs. Rapid reverse branching permits you to transfer execution to *any* step number of program memory. With a negative number stored in the R_0-register, the resulting step number can always be found by combining the negative number in R_0 with the step number of the GTO [i] or GSB [i] instruction.

Execution can even be transferred backward past step 00. To find the resulting step number of program memory, find the sum of the negative number in the R_0-register and the step number containing the GTO [i] or GSB [i] instruction, then add 98. Thus, if the R_0-register contained −11 and a GTO [i] instruction were encountered in step 07, execution would be transferred to step 94 of program memory (7 − 11 + 98 = 94).

Example: The program on page 175 contains an infinite loop that generates and displays a Fibonacci series (refer to page 142 for an explanation of a Fibonacci series.) Although you normally would not set up a single routine that began in step 85 and continued through step 08, the routine illustrates how the GTO [i] instruction coupled with a negative number in the R_0-register can transfer program execution back in program memory, even past step 00.



When the program is run, steps 86 through 89 store −10 in the R_0-register. Thereafter, execution of the GTO i instruction in step 07 causes the running program to jump back 10 steps and resume execution with step 95 (that is, 07 − 10 + 98 = 95). Thus, an infinite loop is set up that generates and displays the Fibonacci series until you stop the program by pressing R/S (or any key) from the keyboard.

To load the complete program, you must first load the instructions in steps 01 through 08, then go to step 84 and load the instructions into steps 85 through 98. To load the program into the calculator:

Set the calculator to PRGM mode.

Press	HP-19C		HP-29C	
f CLEAR PRGM	00		00	
STO 1	01	45 01	01	23 01
RCL 1	02	55 01	02	24 01
RCL 2	03	55 02	03	24 02
+	04	41	04	51
f PAUSE	05	16 64	05	14 74
STO 2	06	45 02	06	23 02
GTO i	07	14 12	07	13 22
g RTN	08	25 13	08	15 12

Now go to step 84 and continue loading instructions, beginning with the g LBL 1 contained in step 85:

Press	HP-19C	HP-29C	
GTO • 84	84 64	84 74	Sets calculator to step 84.
g LBL 1	85 23 14 01	85 15 13 01	
1	86 01	86 01	
0	87 00	87 00	
CHS	88 22	88 32	
STO 0	89 45 00	89 23 00	
0	90 00	90 00	
STO 1	91 45 01	91 23 01	
1	92 01	92 01	
STO 2	93 45 02	93 23 02	
f PAUSE	94 16 64	94 14 74	
RCL 1	95 55 01	95 24 01	
RCL 2	96 55 02	96 24 02	
+	97 41	97 51	
f PAUSE	98 16 64	98 14 74	

Now switch to RUN mode and run the program. Press R/S (or any key) to stop the program after you have seen how quickly the Fibonacci series increases. To run the program:

Set the calculator to RUN mode.

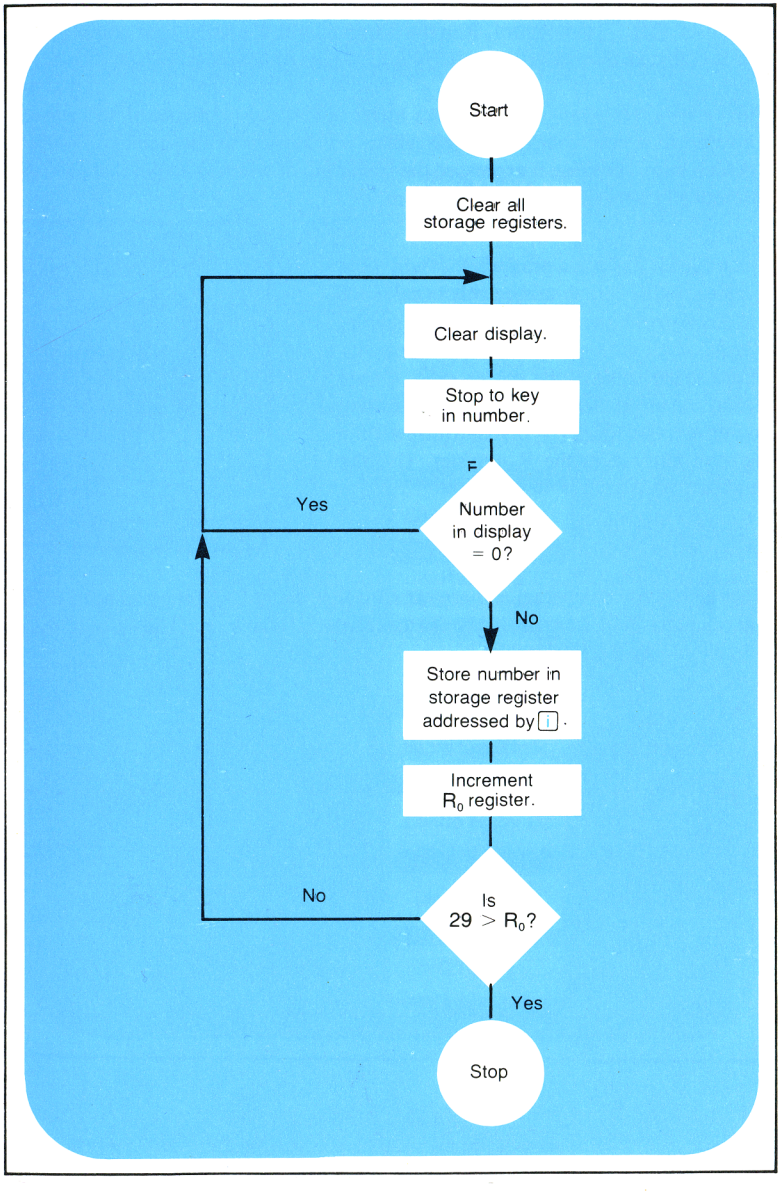
Press	Display
GSB 1	1.00
	1.00
	2.00
	3.00
	5.00
	8.00
	13.00
	21.00
	34.00
	55.00
	89.00
	144.00
	233.00
	377.00
R/S	610.00

Each element in the Fibonacci series is the sum of the previous two elements in the series.

Rapid reverse branching can be specified with numbers from –1 through –99 in the R_0-register. If you attempt to execute GTO i or GSB i when the magnitude of the integer portion of the negative number in R_0 is greater than 99, the calculator displays

Problems

1. a. Create and load a program using [ISZ] and [STO] [i] that permits you to key in a series of values during successive stops. The values should be stored in storage registers R_1 through R_9, $R_{.0}$ through $R_{.5}$, and $R_{(16)}$ through $R_{(29)}$ in the order you key them in. Use the following flowchart to help you.



b. Now create and load a program immediately after the first one that will recall and display the contents of each storage register in reverse order (that is, display $R_{(29)}$ first, then $R_{(28)}$, etc.). The program should stop running after it has displayed the contents of R_1.

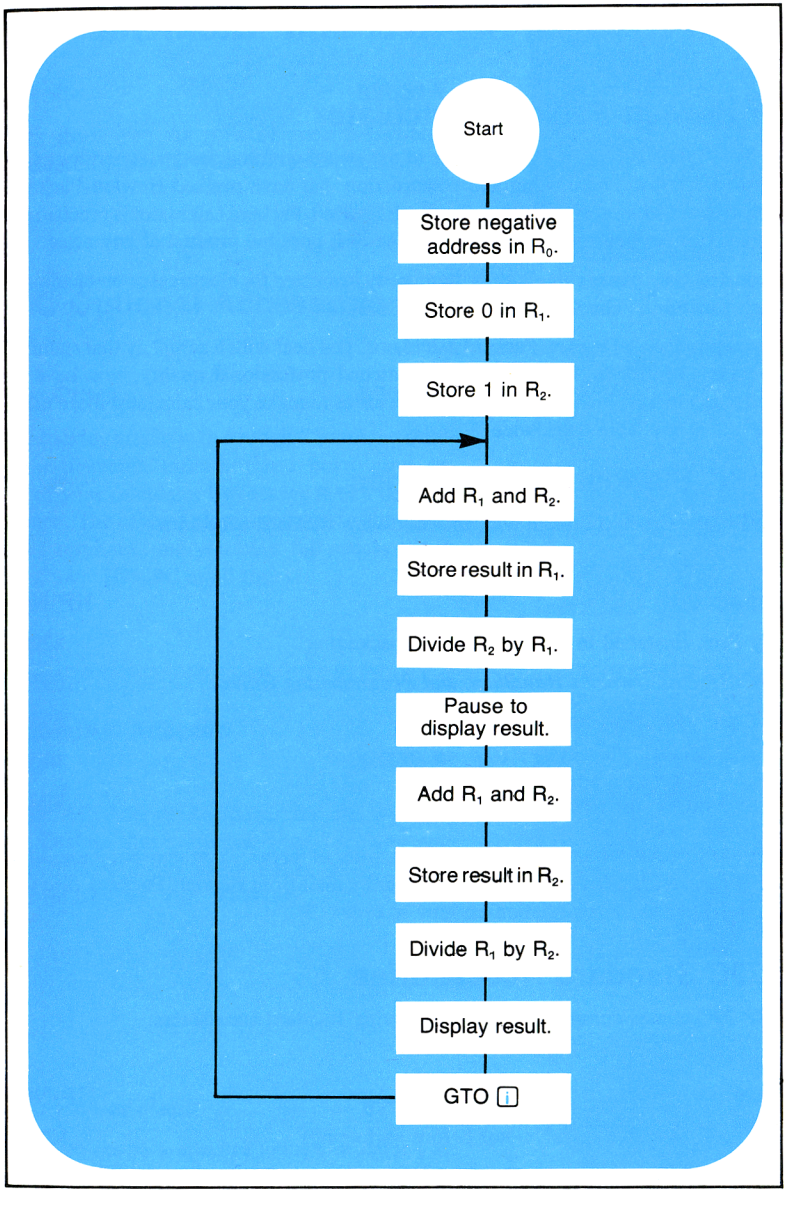
Run the program you loaded for problem 1a, keying in a series of 29 different values. Then run the program you loaded for 1b. All 29 values should be shown, but the last one you keyed in should be the first displayed, etc.,

2. Modify the Random Number Generator program on pages 169-172 to use GSB i instead of GTO i for control. Run the program with the same seed numbers to ensure that it still runs correctly.

3. One curious fact about the Fibonacci series is that the quotients of successive terms converge to a common value. This value was known to the ancient Greeks as the "golden ratio" because it expressed the ideal ratio of width to length that gave the most aesthetically appealing building or room.

Create, load, and run a program that will yield this ideal ratio. You should be able to calculate and display each successive ratio (for example, 2/3, 3/5, 5/8, 8/13, etc.,) until the series converges to the value of the golden ratio. Create a loop by using the rapid reverse branching power of the GTO i instruction with a negative number in the R_0-register. Use the flowchart on page 179 to help you.

When you run the program and are satisfied that the golden ratio has been calculated, you can press R/S from the keyboard to stop the infinite loop. (The value of the golden ratio should be 0.618033989.)


Start
Store negative address in R_0.
Store 0 in R_1.
Store 1 in R_2.
Add R_1 and R_2.
Store result in R_1.
Divide R_2 by R_1.
Pause to display result.
Add R_1 and R_2.
Store result in R_2.
Divide R_1 by R_2.
Display result.
GTO i

Appendix A

Accessories, Service, and Maintenance

Your Hewlett-Packard Calculator

Your HP-19C/HP-29C 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, and each function is checked for proper operation.

When you purchase a Hewlett-Packard calculator, you deal with a company that stands behind its products. Besides an instrument of unmatched professional quality, you have at your disposal many extras, including a host of accessories to make your calculator more usable and service that is available worldwide.

HP-19C Standard Accessories

Your HP-19C comes complete with the following standard accessories:

Accessory	HP Number
Battery Pack (installed in calculator before packaging)	82052A
HP-19C/HP-29C Owner's Handbook and Programming Guide	5955-2110
HP-19C/HP-29C Applications Book	5955-2111
AC Adapter/Recharger (90-127 Vac, 50-60 Hz)	82059A
Carrying Case	82064A

Your HP-19C also comes standard with two rolls of paper. You can purchase additional standard accessories from your nearest dealer or by mail from Hewlett-Packard. See Optional Accessories below for information on how to order.

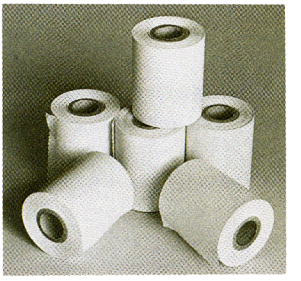
HP-29C Standard Accessories

You HP-29C comes complete with the following standard accessories:

Accessory	HP-Number
Battery Pack (installed in calculator before packaging)	82019A
HP-19C/HP-29C Owner's Handbook and Programming Guide	5955-2110
HP-19C/HP-29C Application Book	5955-2111
AC Adapter/Recharger (90-127 Vac, 50-60 Hz)	82041A
Carrying Case	82027A

You can purchase additional standard accessories from your nearest dealer or by mail from Hewlett-Packard. See Optional Accessories below for information on how to order.

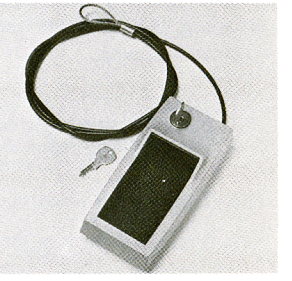
HP-19C Optional Accessories



Paper Rolls 82051A

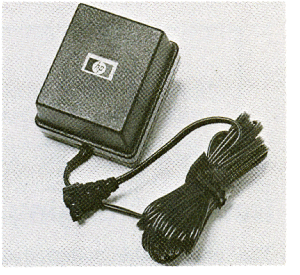
Each pack gives you six rolls of special Hewlett-Packard thermal paper for your HP-19C printer.

HP-29C Optional Accessories



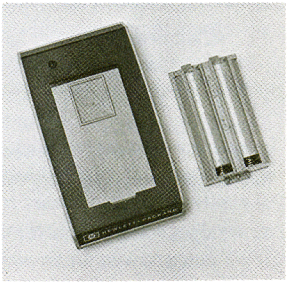
Security Cradle 82029A

A durable locking cradle with a tough 6-foot long steel cable that prevents unauthorized borrowing or pilferage of your calculator by locking it to a desk or work surface. The cable is plastic-covered to eliminate scarring of furniture, and you have full access to all features of your HP-29C at all times.



Switchable AC Adapter/ Recharger 82026 A

Switchable AC Adapter/Recharger for use with the HP-29C. Enables you to operate your calculator and recharger battery packs using either 90-127 Vac, 50-60 Hz or 200-254 Vac, 50-60 Hz.



Reserve Power Pack 82028A

The reserve power pack attached to the calculator's ac adapter/recharger to keep an extra battery pack freshly charged and ready for use. Comes complete with extra battery pack.

To order additional standard or optional accessories for your HP-19C/HP-29C see your nearest dealer or fill out an Accessory Order Form and return it with check or money order to:

HEWLETT-PACKARD
Corvallis Division
P.O. Box 999
Corvallis, Oregon 97330

If you are outside the U.S., please contact the Hewlett-Packard Sales Office nearest you.

Availability of all accessories, standard or optional, is subject to change without notice.

AC Line Operation

Your calculator contains a rechargeable battery pack that is made up 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. Even though you are using the ac adapter/recharger, the batteries must remain in the calculator whenever the calculator is used.

> **Note:** Attempting to operate the calculator from the ac line with the battery pack removed may result in wrong or improper displays.

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

1. You need not turn the calculator OFF.
2. Insert the female ac adapter/recharger plug into the rear connector of the calculator.
3. Insert the power plug into a live ac power 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 being charged when you are operating the calculator from the ac adapter/recharger. With the batteries in the calculator and the recharger connected, the batteries will charge with the calculator off or on. Normal charging times from fully discharged battery pack to full charge are (times depend on ac line voltage value):

Calculator off: 6 – 12 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:** It is normal for the ac adapter/recharger to be warm to the touch when it is plugged into an ac outlet.

Battery Operation

To operate the calculator from battery power alone, simply disconnect the female 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.

The HP-19C printer is the most power-consuming part of the calculator, and you can maximize battery operating time by leaving the calculator in MAN printing mode when printing is not necessary.

Using Continuous Memory

When you turn off your calculator, the following information is retained:

- All programs that are loaded into the calculator.
- Contents of the 16 primary storage registers.
- Display status (FIX, SCI, or ENG, and number of displayed digits).
- Contents of the displayed X-register.

Regardless of where you stopped in a program, the calculator returns to step 00 (top of program memory) when you turn it on again.

Numbers in the T-, Z-, and Y-registers of the stack, LAST X, and trigonometric mode status (DEG, RAD, or GRAD) are not saved when you turn the calculator off; however you can use the primary storage registers to retain data in the calculator.

Continuous Memory requires that the *batteries be kept in the calculator*. If the low power indicator appears in the display, turn your calculator off immediately, and connect it to an ac outlet or insert a new battery pack. If you allow the battery to discharge completely, the information in Continuous Memory will be lost.

If you drop or traumatize your calculator, or if power to the Continuous Memory is interrupted whether the calculator is off or on, the contents of program memory and the data storage registers may be lost. If this occurs, when the calculator is then turned on, the display will show **Error.** To restore the display, ensure that the battery is charged, or connect the ac adapter/recharger, and press any key.

Battery Pack Replacement

If it becomes necessary to replace the battery pack, use only another Hewlett-Packard battery pack like the one shipped with your calculator. Continuous Memory requires that batteries be replaced as quickly as possible. Normally you have a minimum of 5 seconds to change the batteries. Leaving batteries out of the calculator for extended periods will result in loss of information in Continuous Memory.

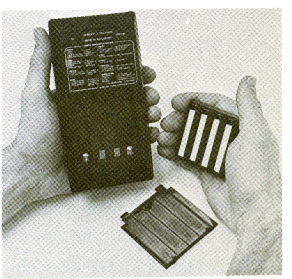
CAUTION

Use of any batteries other than the Hewlett-Packard battery pack may result in damage to your calculator.

HP-19C Battery Pack Replacement

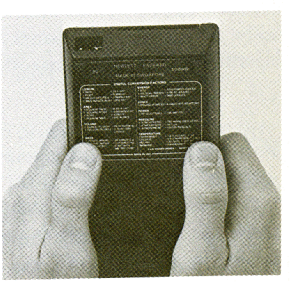
To replace the battery pack, use the following procedure:

1. Turn the HP-19C power switch to OFF, and disconnect the recharger from the calculator.
2. Place your thumb in the semicircular slot on the battery compartment door, and press down. The door will spring open.



3. Remove the battery pack.
4. Drop in a new pack.

5. Slant the leading edge of the door into the lower edge of the doorway. Place your thumbs on the two pads on the upper edge of the door, and press firmly. The latch will snap into place.

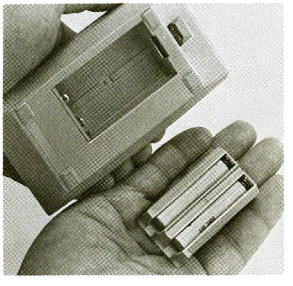


HP-29C Battery Pack Replacement

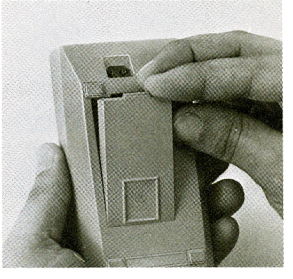
To replace the battery pack, use the following procedure:

1. Set the calculator ON-OFF switch to OFF and disconnect the battery charger/ac adapter from the calculator.
2. Press down on the thumbset at the rear of the calculator and slide the battery pack in the direction of the arrow.

3. When the key on the battery pack becomes visible pull that end of the pack up and permit the battery pack to fall into the palm of your hand.
4. Insert the new battery pack in the direction of the arrow. Slant the leading edge of the pack into the edge of the doorway.



5. Snap the battery pack into place by pressing it gently.



If you use your HP-29C extensively in field work or during travel, you may want to order the optional Reserve Power Pack, consisting of a battery charging attachment and a spare battery pack. The Reserve Power Pack enables you to charge one battery pack while using the other in the calculator.

Battery Care

When not being used, the batteries in your calculator have a self-discharge rate of approximately 1% of available charge per day. After 30 days, a battery pack could have only 50 to 75% 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. 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. The battery pack is warranted for one year, and if the warranty is in effect, 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 or use the Accessory Order Form provided with your calculator to order a replacement.

WARNING

Do not attempt to incinerate or multilate 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.

To maximize the life you get from battery pack, keep HP-19C printing to a minimum and display only the fewest number of digits necessary during portable operation.

Your HP-19C Printer

The printing device in your HP-19C is a thermal printer that uses a moving print head to print upon a special heat-sensitive paper. When the print head is energized, it heats the paper beneath it. The heat causes a chemical change in the paper, which then changes color. The printer in your HP-19C prints answers quickly and quietly, and has been expressly designed to give you a permanent record of your computations in a portable scientific calculator.

Paper for Your HP-19C

Because the printer in your HP-19C is a thermal printer, it requires special heat-sensitive paper. You should use only the Hewlett-Packard thermal paper available in 22-foot rolls from your nearest HP distributor or sales office, or by mail from:

HEWLETT-PACKARD
Corvallis Division
P.O. Box 999
Corvallis, Oregon 97330

Because of the special heat-sensitive requirements of the paper, standard adding machine paper will *not* work in the HP-19C. Also, since different types of thermal paper vary in their sensitivities, the use of thermal paper other than that available from Hewlett-Packard may result in poor print quality or even in damage to your calculator.

CAUTION

Use only Hewlett-Packard paper in your HP-19C.

The heat-sensitive paper used in your HP-19C should be stored in a cool, dark place. Discoloration of paper may occur if it is exposed to direct sunlight for long periods of time, if storage temperatures rise above 50°C (122°F), or if the paper is exposed to excessive humidity or to acetone, ammonia, or other organic compounds. (Exposure to gasoline or oil fumes will not harm your HP-19C paper supply.)

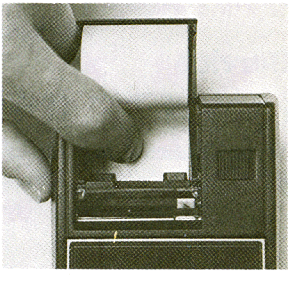
Printed tapes from your HP-19C will last 30 days or more without fading under fluorescent light, but to ensure the permanence of your records, you should store printed tapes at room temperature in a dark place away from direct sunlight, heat, or fumes from organic compounds. (For added permanence, you can copy tapes with a suitable office copier.)

Replacing the Paper

To replace the paper roll in your HP-19C, proceed as follows:

1. Push the switch next to the paper well to the right. The paper cover will spring open.

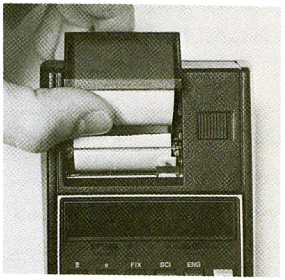
2. Remove the empty core from the paper well.
3. Before inserting the new roll of paper, discard the first 2/3 turn to ensure that no glue, tape, or other foreign matter is on the paper. Make sure that the leading edge of the paper is straight, not crooked or jagged. **Do not fold the paper as the double thickness of the edge may obstruct the paper feed.**



4. Temporarily place the paper roll in the paper roll cover. With your finger, push the leading edge of paper into the slot near the bottom of the paper well. Continue pushing until the paper passes the top edge of the plastic tear bar.

5. Turn the calculator on, and press g SPC to assure that the paper is advancing properly.

6. Drop the roll of paper into the paper well and close the cover.



If the paper is feeding properly through the printer mechanism but no printing appears on the tape, the paper roll is probably inserted backwards. The paper is chemically treated and will print on only one side.

Printer Maintenance

The printer in your HP-19C, like the rest of the calculator, is crafted for engineering excellence and is designed to give trouble-free operation with a minimum of maintenance. All moving parts in the printer mechanism contain self-lubricating compound, and no lubrication, cleaning, or servicing of the mechanism is ever required. You may want to occasionally remove the clear plastic tear bar and clean it with mild soap and water solution. (Do not use acetone or alcohol to clean the tear bar.)

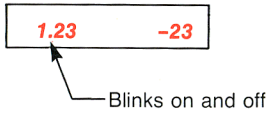
You should *never* attempt to insert a tool, such as a screwdriver, pencil, or other hard object, into the printer or its mechanism. If the paper tape should become jammed and fail to feed properly, clear it by grasping the tape and pulling it forward through the printer mechanism. (You can remove the plastic tear bar for accessibility.)

> **Note:** Printer operation may be affected if the printer is in close proximity to a strong magnetic field. Normal operation can be restored by removing the calculator from the vicinity of the magnetic field. No permanent damage will result.

Service

Low Power

When you are operating from battery power in RUN mode, the decimal point blinks on and off to warn you that you have a limited operating time left. On the HP-19C, switching the Print Mode switch to MAN (and curtailing printing operations) may result in an extension of operating time.



In PRGM mode, a blinking decimal point will appear between the step number and the keycode.

You must then either connect the ac adapter/recharger to the calculator as described under AC Line Operation, or you must 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 in RUN mode, 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 from 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

Warranty

Full One-Year Warranty

The HP-19C/HP-29C and its accessories are warranted against defects in materials and workmanship for one (1) year from the date of delivery. During the warranty period, Hewlett-Packard will repair or, at its option, replace at no charge components that prove to be defective, provided the calculator or accessory is returned, shipping prepaid, to Hewlett-Packard's Customer Service Facility. (Refer to Shipping Instructions).

This warranty does not apply if the calculator or accessory has been damaged by accident or misuse, or as a result of service or modification by other than an authorized Hewlett-Packard Customer Service Facility. No other expressed warranty is given by Hewlett-Packard. **HEWLETT-PACKARD SHALL NOT 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.

Out-of-Warranty

After the one-year warranty period, calculators will be repaired for a moderate charge. All repair work performed beyond the warranty period is warranted for a 90-day period.

Warranty Transfer

If you sell your calculator or give it as a gift, the warranty is transferable and remains in effect for the new owner until the original one-year expiration date. It is not necessary for the owner to notify Hewlett-Packard of the transfer.

Warranty Information Toll-Free Number

800-648-4711 In Nevada call collect 702-323-2704.

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.

Repair Policy

Repair Time

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

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 Customer Service Facility. 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 Customer Service Facility. Send the unit to:

Hewlett-Packard
Customer Service Facility
1000 N.E. Circle Blvd.
Corvallis, OR 97330

After warranty repairs are completed, the Customer Service Facility 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).

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 Customer Service Facility.

Appendix B

Improper Operations

If you attempt a calculation containing an improper operation—say, division by zero—the calculator display will show ***Error.*** The calculator will display ***Error*** when the calculator is first turned on if power to Continuous Memory has been interrupted. In addition, if the HP-19C Print Mode switch is set to NORM or TRACE, the word ERROR will be printed. The following are improper operations:

÷	where $x = 0$.
y^x	where $y = 0$ and $x \leq 0$, where $y < 0$ and 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 $\lvert x \rvert$ is > 1.
$\cos^{-1}$	where $\lvert x \rvert$ is > 1.
STO ÷	where $x = 0$.
$\bar{x}$	where $n = 0$.
s	where $n \leq 1$.

STO + n, STO − n, STO × n, STO ÷ n, STO + • n, STO − • n, STO × • n, STO ÷ • n, where magnitude of number in storage register n would then be larger than $9.999999999 \times 10^{99}$.

STO i where ABS (INT R_0) > 29.
RCL i where ABS (INT R_0) > 29.

STO + i, STO − i, STO × i, STO ÷ i, where ABS (INT R_0) > 29, or where magnitude of number in storage register addressed by R_0 would be larger than $9.999999999 \times 10^{99}$.

GTO i, GSB i, where INT $R_0 < -99$ or INT $R_0 > 9$ or $0 \leq$ INT $R_0 \leq 9$ and there is no such label.

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, like PRx (HP-19C) and GTO 3 , 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 PRx 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 3 instruction followed by a digit entry sequence, the stack will lift.

The table below lists all legal operations on the HP-19C/HP-29C. 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.

Printed Symbol	Keystrokes	Enabling, Disabling, or Neutral	Saves x in LAST X
ABS	g ABS	E	Yes
	g BST	N	
CHS	CHS *	E	
CLRG	f CLEAR REG	N	
CLΣ	f CLEAR Σ	N	
CLX	CL x	D	
COS	f COS	E	Yes
COS⁻¹	g COS⁻¹	E	Yes
DEG	g DEG	N	
	g DEL	N	
÷	÷	E	Yes
DSZ	g DSZ	N	
	EEX *		
ENG1 ENG9	f ENG 0 through 9	N	
ENT↑	ENTER↑	D	
e^x	g e^x	E	Yes
FRC	g FRAC	E	Yes
FIX0 FIX9	f FIX 0 through 9	N	
GRAD	g GRD	N	
GSB0 GSB9	GSB 0 through 9		
GTO0 GTO9	GTO 0 through 9		
→HMS	f →H.MS	E	Yes
→H	g →H	E	Yes
	g i	E	
INT	f INT	E	Yes
ISZ	g ISZ	N	
LSTX	f LAST X	E	
*LBL0 *LBL9	g LBL 0 through 9	N	
LN	f ln	E	Yes
LOG	f log	E	Yes
−	−	E	Yes
→P	g →P	E	Yes
%	%	E	Yes
Pi	g π	E	
+	+	E	Yes
PSE	f PAUSE	N	
PREG	f PRT REG	N	
PRTΣ	f PRT Σ	N	
	g SPC	N	
PRST	f PRT STK	N	
	PRX	N	
→R	f →R	E	Yes
R↓	R↓	E	
RAD	g RAD	N	
RCL0 RCL9	RCL 0 through 9	E	
RC.0 RC.5	RCL • 0 through 5	E	

Printed Symbol	Keystrokes	Enabling, Disabling, or Neutral	Saves x in LAST X
R/S	R/S	N	
RTN	g RTN **	N	
S	f S	E	Yes
SCI0 SCI9	f SCI 0 through 9	N	
Σ+	Σ+	D	Yes
Σ-	f Σ-	D	Yes
SIN	f sin	E	Yes
SIN⁻¹	g sin⁻¹	E	Yes
√X	f √x	E	Yes
	SST	N	
ST÷0 ST÷9	STO ÷ 0 through 9, ⊡0 through ⊡5	E	
S÷.0 S÷.5			
ST-0 ST-9	STO − 0 through 9, ⊡0 through ⊡5	E	
S-.0 S-.5			
ST+0 ST+9	STO + 0 through 9, ⊡0 through ⊡5	E	
S+.0 S+.5			
ST×0 ST×9	STO × 0 through 9, ⊡0 through ⊡5	E	
Sx.0 Sx.5			
STO0 STO9	STO 0 through 9	E	
ST.0 ST.5	STO ⊡ 0 through 5	E	
STOi	STO i	E	
TAN	f tan	E	Yes
TAN⁻¹	g tan⁻¹	E	Yes
x	×	E	Yes
X=0?	g x=0	N	
X≠0?	g x≠0	N	
X<0?	g x<0	N	
X>0?	g x>0	N	
X=Y?	f x=y	N	
X≠Y?	f x≠y	N	
X≤Y?	f x≤y	N	
X>Y?	f x>y	N	
x̄	f x̄	E	Yes
X²	g x²	E	Yes
1/X	g 1/x	E	Yes
X≷Y	x≷y	E	
Yˣ	f yˣ	E	Yes
10ˣ	g 10ˣ	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.

**The end of a program acts exactly like a RTN operation.

HP-19C/HP-29C Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U-Z

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 (statue)†	=	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 chosen. * Exact values.

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 collect 702/323-2704.)

5955-2110

Printed In U.S.A.