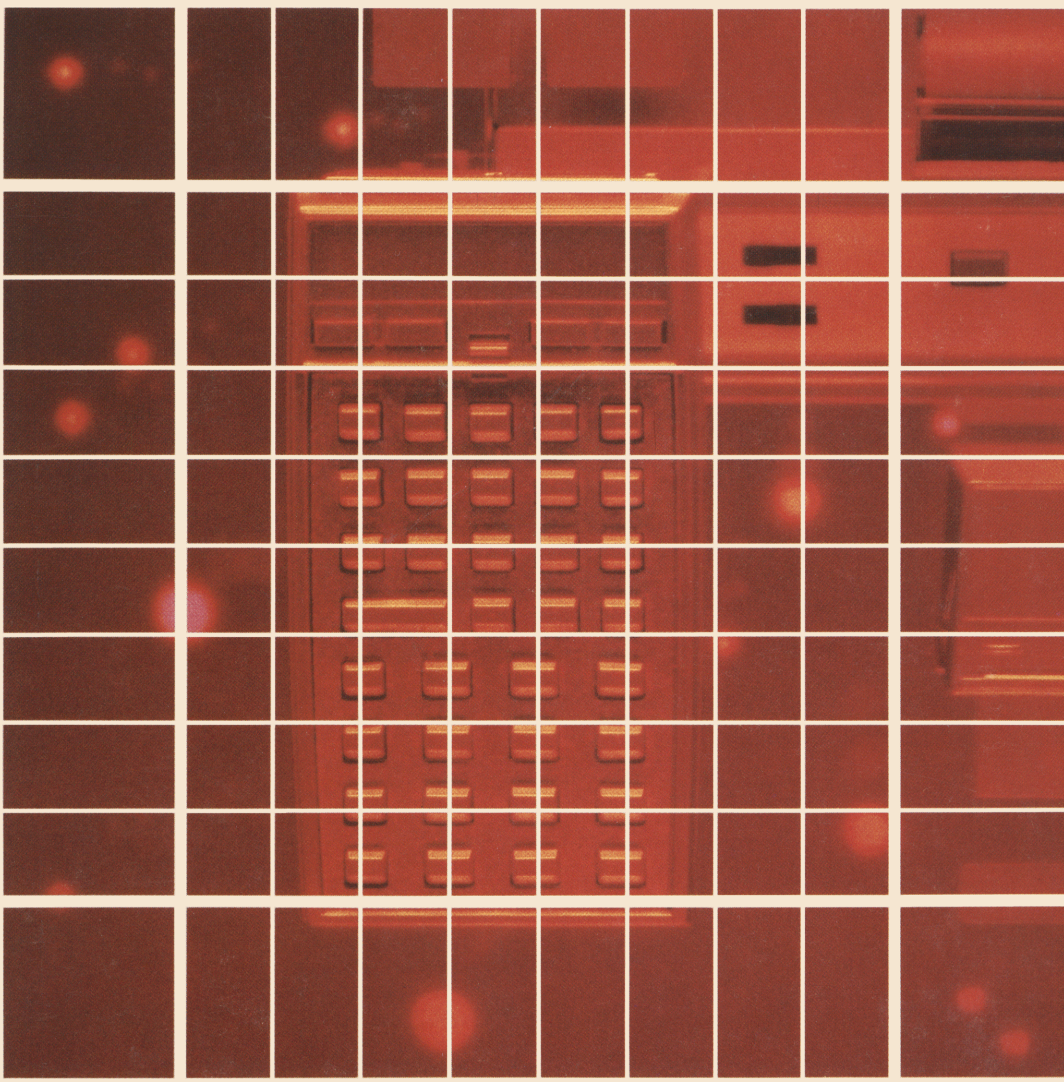
HEWLETT-PACKARD

OWNER'S HANDBOOK AND PROGRAMMING GUIDE

HP-41C



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

Statement of Corporate Objectives
Hewlett-Packard

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

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

HEWLETT PACKARD

The HP-41C Alphanumeric Programmable Scientific Calculator

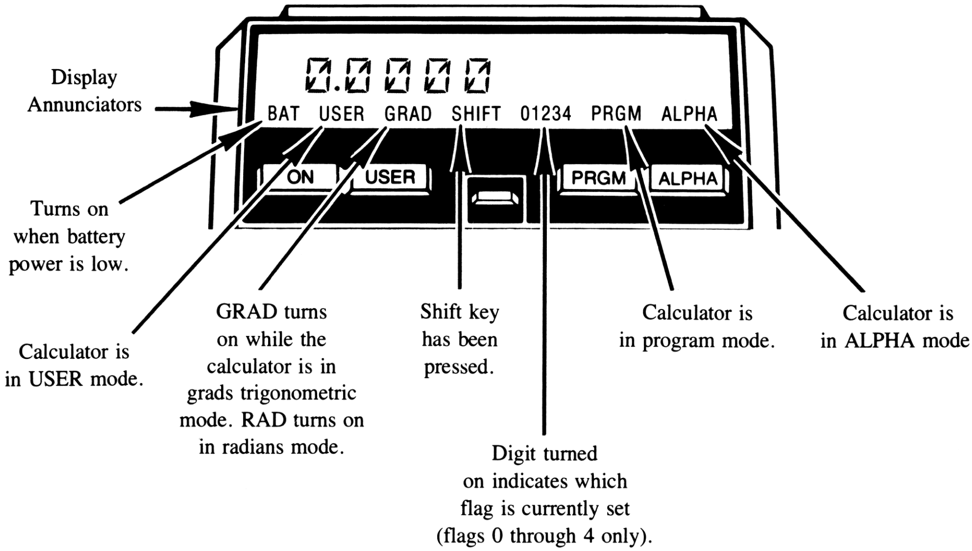
Owner's Handbook and Programming Guide

March 1980

00041-90001 Rev. C 3/80

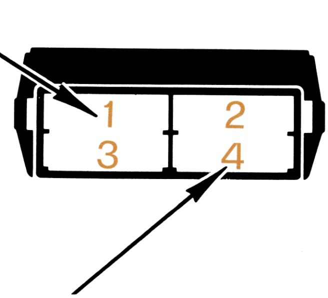
Printed in U.S.A.

Contents



Input/Output Port Locations

Add memory modules beginning with port 1. Do not skip ports when inserting additional memory modules. Each memory module contains 64 registers of memory. Always remove memory modules beginning with the highest numbered port. Since removing memory modules reduces the number of data storage registers, you must be sure that there are enough data storage registers allocated to account for the registers you are removing.



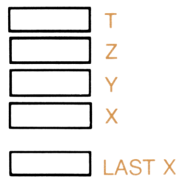
Insert application modules and peripherals beginning with the highest-numbered available port.

To remove:	Execute:
1 memory module	SIZE 064
2 memory modules	SIZE 128
3 memory modules	SIZE 192
4 memory modules	SIZE 256

CAUTION

Always turn the HP-41C off before inserting or removing plug-in modules or peripherals. Failure to do so could damage both the calculator and the module or peripheral.

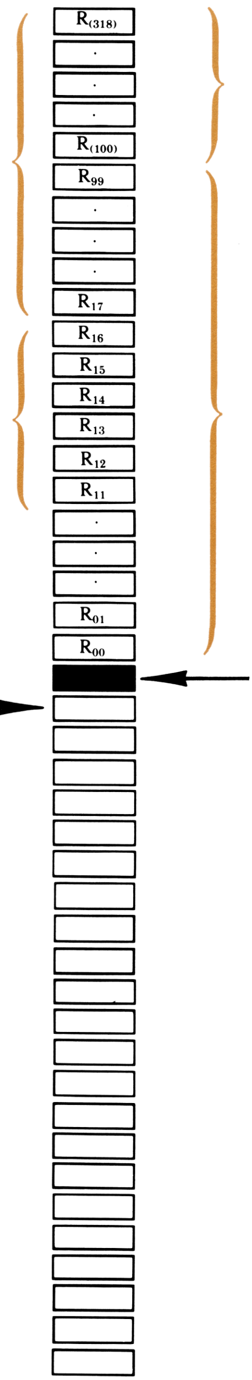
Automatic Memory Stack



ALPHA Register

(24 characters maximum)

Program Memory and Data Storage Registers



Memory modules expand memory here.

Extended data storage registers (addressed indirectly).

Statistics registers. Location definable using [ΣREG] .

Primary data storage registers. Initial allocation is R_{00} through R_{16} to data storage registers. Can be changed with the [SIZE] function.

Movable memory partition. Initial allocation of memory is 17 registers to data storage and the remainder (46) to program memory. This allocation can be changed (the memory partition moved up or down) using [SIZE] . [SIZE] specifies the number of registers allocated to data storage registers--the remainder are automatically allocated to program memory. There are seven bytes in each register of program memory.

Top of Program Memory.

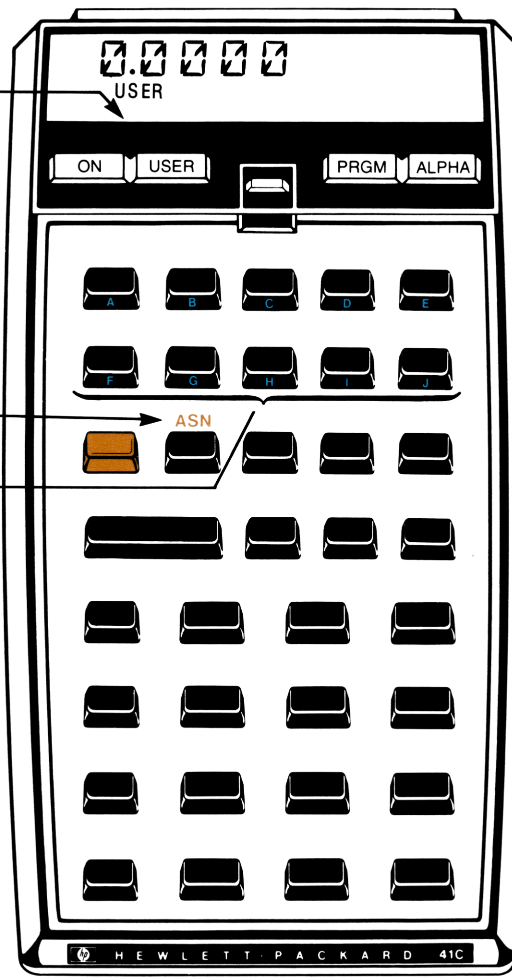
Program memory lines.

Each program should begin with an ALPHA label and end with an END instruction. Press [shift] [GTO] [•] [•] before you begin keying in a program.

Maximum of 15 ALPHA characters in any one program line.

Function key assignments of standard functions (not programs that you write) are stored below unused program memory.

USER Mode Keyboard



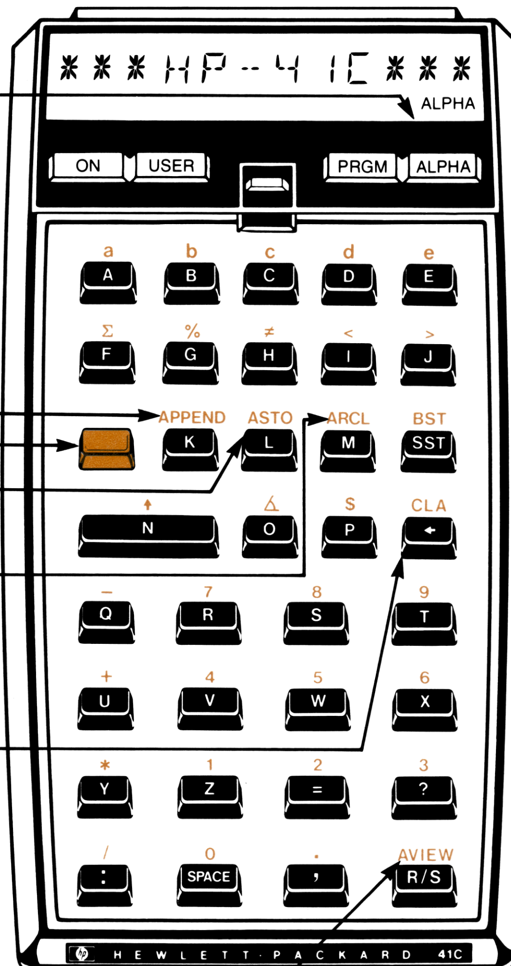
USER mode key toggles the HP-41C into and out of USER mode. In USER mode, the functions on reassigned keys become active. If a key is not reassigned, the normal mode function (shown on the face of or above the key) remains active.

ASN (assign) is used to reassign the keyboard for USER mode operation. The only key locations that cannot be reassigned are , ON , USER , PRGM , and ALPHA . All other key locations, including shifted key locations, can be reassigned. Any function in any of the catalogs can be reassigned. This includes programs with ALPHA labels that you store into program memory. To assign a function or program to a key:

1. Press ASN .
2. Press ALPHA and the program or function name to be assigned to a key. Press ALPHA again.
3. Press the key, or and the key which is being reassigned.
4. The key becomes active with the reassigned function while the calculator is in USER mode.

To reassign a key back to its normal mode function, press ASN ALPHA ALPHA and the key (or and the key).

If a key in the top two rows (and shifted top row) is not reassigned, special local label searching is performed in USER mode. When the key is pressed the calculator searches for the local labels that correspond to the ALPHA character on the key (LBL A through LBL J and LBL a through LBL e). If the label is found in the current program, execution begins there. If the label is not found in the current program, the normal mode function of the key is executed. In USER, normal and PRGM modes, the top row of keys corresponds to the numbers 01 through 05 and the second row corresponds to the numbers 06 through 10 (from left to right). The number/key correspondence is effective only when a function is executed or keyed in that expects a one or two-digit parameter or address. For example, pressing STO LN is the same as pressing STO 05. Only the right-most digit is used when a single-digit parameter is expected.



ALPHA mode key places the calculator into ALPHA mode. Effective in normal, user or program modes.

APPEND is used for adding to or editing ALPHA strings. If ALPHA entry has been terminated, APPEND puts back the prompt so you can add to the ALPHA register.

Shift key pressed before another key specifies a shifted ALPHA character or function (shown here above the keys). To get the character shown here on the face of the key, simply press that key.

ASTO stores the left-most six characters in the ALPHA register into the indicated data storage register.

ARCL recalls the contents of the indicated register and adds it to whatever is already in the ALPHA register.

CLA clears the display and ALPHA register.

The ← key deletes the right-most character from the ALPHA register. (Clears ALPHA register and display if ALPHA entry has been terminated.)

AVIEW displays the contents of the ALPHA register without disturbing the ALPHA or X-registers.

Pressing a key executes the function shown on the face of that key.

pressed before another key executes the function shown above that key.

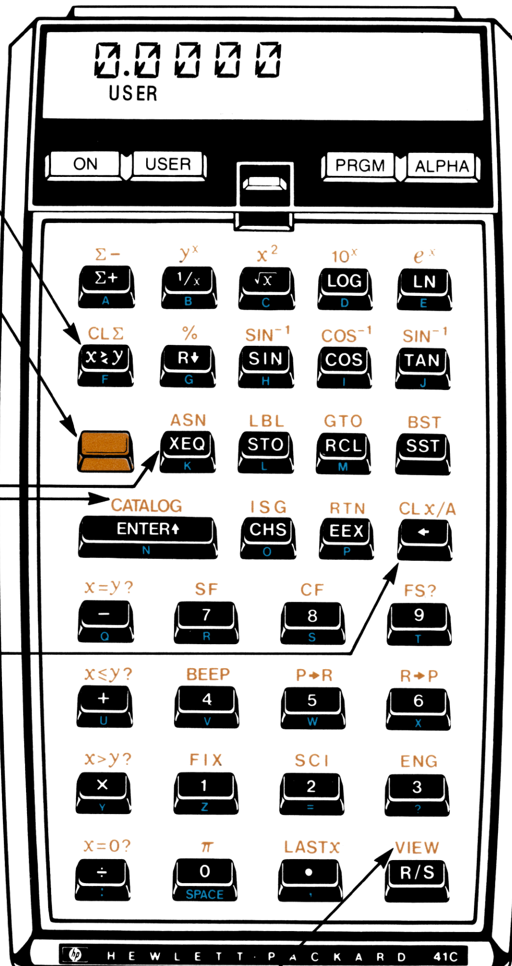
XEQ (execute) is used to execute functions and programs from the display. To execute and standard function, or a program that is stored in program memory:

1. Press XEQ.
2. Press ALPHA and the program or function name. Press ALPHA.

The named function or program will be executed. If the function expects parameters, then the calculator prompts for them.

CATALOG lists the contents of the function and program catalogs. Catalog 1 is the user catalog and it contains the global (ALPHA) labels and END instructions of programs stored in program memory. The catalog is positioned in memory to the displayed program labels during the listing of catalog 1. Catalog 2 lists all functions contained in currently plugged-in application modules and peripherals. Catalog 3 lists all standard HP-41C functions. Press any key (other than ON or R/S) to slow the listing down. Press R/S and any other key to terminate a running listing. Press R/S to stop the listing (use SST and BST to step through the catalog manually).

Deletes the right-most digit during digit or parameter entry. If digit entry has been terminated, clears entire displayed X-register. Also clears messages from the display. Press CLX to clear the displayed X-register. If ← is held down while HP-41C is turned on, does master clear.



Views the contents of any register without disturbing the stack. To clear viewed data and return to the contents of the X-register, press ←.

Introducing the HP-41C

The Philosophy of the HP-41C System

The HP-41C represents a totally new concept in the design of Hewlett-Packard calculators. In fact, because of the advanced capabilities of the HP-41C, it can even be called a personal computing system. The HP-41C is the first Hewlett-Packard handheld calculator offering an exciting array of alphanumeric capabilities.

With so many different kinds of calculator users and applications in the world, we at Hewlett-Packard decided we could provide a significant contribution by designing and building you a *quality* calculator with expandable and flexible capability. The alphanumeric HP-41C is just that calculator.

You can increase the storage capacity of the basic calculator by five times. And, if you wish, you can even specify which functions are active on the keyboard, and how they are positioned. As expansions of the calculator system we are making available a number of peripherals to provide you with a true computing system—one that can even interface with other devices.

The HP-41C has a great number of functions, and at first you will not need to learn how every function and feature works; just be aware that they are there. A part of the design philosophy of the HP-41C was to provide a healthy number of functions and let you choose what you need. As your programming and calculating needs expand and become more sophisticated, you will use more and more of the functions provided. If you need a function that is not on the basic HP-41C, chances are that you can write a program that will fill that special need. Those special programs, along with all programs you write can be assigned by name to the keyboard for execution just like any of the standard functions—at the press of a single key! We are also making available a continuing supply of special application modules that plug into the HP-41C. They are designed to give you answers you need to solve special application problems.

Aside from the advanced computer-like capabilities of the HP-41C, possibly the most attractive feature of the machine is its ability to solve problems *easily*. Experience or knowledge of complicated programming languages is not required. Yet some of the most sophisticated computer experts appreciate the advanced programming and operating features of the HP-41C.

Obviously the HP-41C is part of an extremely capable personal computing system. At the same time, the HP-41C is a very friendly calculator, so take some time to work carefully through this handbook. You will be surprised at how easily and quickly you will learn to take advantage of the power of your new HP-41C.

Sample Problems

The HP-41C display contains seven "annunciators" or key words that tell you the status of the calculator.

BAT USER GRAD SHIFT 0 1 2 3 4 PRGM ALPHA

Press the [ON] key now and check to see if the USER annunciator in the display is on. If the USER annunciator is displayed, press the [USER] key (located just below the display) to turn the USER annunciator off.

If the BAT (battery) annunciator is displayed, or your HP-41C does not have the batteries installed, refer to Batteries, page 240.

To get the feel of your new calculator try a few simple calculations. Press [FIX] 4 now so your display will match the displays in the following problems.

To solve:	Keystrokes	Display
5+6=11	5 [ENTER↑] 6 [+]	11.0000
8÷2=4	8 [ENTER↑] 2 [÷]	4.0000
7−4=3	7 [ENTER↑] 4 [−]	3.0000
9×8=72	9 [ENTER↑] 8 [×]	72.0000
19.85^2	19.85 [x^2]	394.0225

Now let's look at a sample problem to see how the HP-41C is used to solve the problem manually and then automatically using a program.

Most conventional home water heaters are cylindrical in shape, and you can easily calculate the heat loss from such a water heater. The formula $q=hAT$ can be used, where

- q is the heat loss from the water heater (Btu per hour),
- h is the heat-transfer coefficient,
- A is the total surface area of the cylinder, and
- T is the temperature difference between the cylinder surface and the surrounding air.

For our example let's assume you have a 52-gallon cylindrical water heater and you wish to determine how much energy is being lost because of poor insulation. In initial measurements, you found an average temperature difference between the heater surface and surrounding air of 15 degrees Fahrenheit. The surface area of the tank is 30 square feet and the heat transfer coefficient is approximately 0.47.

To calculate the heat loss of the water heater, merely press the following keys in order.

Keystrokes	Display	
15 ENTER↑	*15.0000*	Temperature difference.
30	*30_*	Area of water heater (sq. ft.).
×	*450.0000*	Intermediate answer.
.47	*.47_*	Coefficient of heat transfer.
×	*211.5000*	Heat loss in Btu per hour.

Programming the Sample Problem

The water heater in the example loses about 212 Btu every hour at the 15-degree temperature difference. Suppose you decide to calculate the heat loss of the water at *many* temperature differences. You could calculate the heat loss *manually* for each temperature difference. Or an easier and faster method is to write a *program* that will calculate the heat loss for any temperature difference.

Now let's write, load and run a program to do just that!

Writing the Program. You have already written it! The program is the same series of keystrokes you executed to solve the problem manually. One additional instruction, a *label,* is used to define the beginning of the program.

Loading the Program. To load the instructions of the program into the HP-41C:

Press the following keys in order. The display shows symbols or names representing each instruction entered. The calculator records (remembers) the instructions as you enter them.

Keystrokes	
PRGM	Places the HP-41C into PRGM (program) mode. The annunciator will show in the display to let you know that the HP-41C is now in PRGM mode.
■ GTO • •	This prepares the calculator for the program.
■ LBL ALPHA HEAT ALPHA	Defines the beginning of the program and names it HEAT.
30 × .47 ×	The same instructions you executed to solve the problem manually.

Running the Program. Press the following keys to run the "HEAT" program. Find the heat loss of the water heater at temperature differences of 22 and 65 degrees Fahrenheit.

Keystrokes	Display	
[PRGM]	*211.5000*	Takes calculator out of PRGM mode—turns PRGM annunciator off. Result remains from previous example.
22	*22_*	The first temperature difference.
[XEQ] (execute)	*XEQ __*	Prompts *Execute what?* with *XEQ __*.
[ALPHA] HEAT [ALPHA]	*310.2000*	Press the letter keys to spell the program name. The program is executed and the heat loss (Btu per hour) is displayed.
65	*65_*	The second temperature difference.
[XEQ]	*XEQ __*	Execute what?
[ALPHA] HEAT [ALPHA]	*916.5000*	Btu per hour.
[■] [CLX]	*0.0000*	Clear the display.

You can save even more time and keystrokes by assigning the program to a key on the keyboard! Programs that you assign to keys are treated just like any other functions when you place the HP-41C into a special "USER" mode. Then you can execute your program with the press of a single key—without entering the program name each time! Let's assign the HEAT program to the [Σ+] key now.

Keystrokes	Display	
[■] [ASN]	*ASN_*	The HP-41C prompts *Assign what?*
[ALPHA] HEAT [ALPHA]	*ASN HEAT_*	The program name. The HP-41C is now prompting you for the key location.
[Σ+]	*0.0000*	HEAT is now assigned to the [Σ+] location.

Now run HEAT for temperature differences of 38°F, 27°F and 45°F. To run HEAT, you now press the [USER] key, located just below the display, to place the HP-41C into USER mode. Notice that the HP-41C lets you know that it is in USER mode by turning on the USER annunciator in the display.

Keystrokes	Display	
USER	0.0000	Puts the calculator in USER mode and turns on the USER annunciator.
38 HEAT (Σ+)	535.8000	Since HEAT is assigned to the Σ+ key in USER mode, you can run HEAT quickly and simply as a keyboard function.

Try holding the HEAT (Σ+) key down briefly. Notice how the HP-41C reminds you that HEAT is assigned to that key (in USER mode) by showing the name HEAT in the display while you press and hold the key. (Holding the key down longer than about a half second nullifies the function.)

Keystrokes	Display	
27 HEAT (Σ+)	ᵀHEAT 380.7000	Press and hold the key for a moment to see the program name. When you release the key, the function is executed, giving the answer in Btu per hour.
45 HEAT	634.5000	Btu per hour.
▬ CLX	0.0000	Clears the display.
USER	0.0000	Takes the HP-41C out of USER mode.

Programming the HP-41C is *that* easy! The exciting capabilities of the HP-41C together with the ease of programming and execution make the HP-41C possibly the most usable, capable handheld calculator system you can own.

The HP-41C Configuration

Continuous Memory. The HP-41C maintains *all* information in the calculator in Continuous Memory—one of the newest, most advanced memory systems available in a scientific calculator. All data, programs and functions—everything in the calculator—are maintained by Continuous Memory while the calculator is turned off. You can turn the HP-41C off, then back on and continue working where you left off. In fact, to conserve battery power, the HP-41C turns itself off after 10 minutes of inactivity.

Alphabetic/Numeric Capability. The HP-41C is one of the first handheld scientific calculators to offer both alphabetic and numeric character capability. Alphabetic characters allow you to name and label programs and functions, prompt for data with meaningful words or sentences, display exact error messages, label variables and constants—even display messages!

The Catalogs of Functions. The HP-41C has three separate catalogs of functions. You can list the programs *you* have written; more than 130 resident HP-41C functions; all functions contained in plug-in modules (more about modules in a minute). There is never any doubt as to what is resident in the calculator—all you have to do is list the catalogs.

Key Reassignments. Nearly *any* function in the HP-41C (functions you have written, standard HP-41C functions, application module functions) can be assigned or reassigned to most key or shifted key locations on the keyboard. This allows you to ''personalize'' your calculator, positioning functions on the keyboard where *you* want them.

HP-41C Extensions. The basic HP-41C comes with 63 data storage registers or 63 registers of program memory (that's 200-400 lines)—and you can define the combination of data storage registers and registers of program memory that you desire (for example, the HP-41C begins with a combination of 17 data storage registers and 46 registers of program memory). But you are not limited to the basic machine capacity! You have the option to increase the capacity and capability of your HP-41C by adding up to four additional ''plug-in'' modules. Each module contains 64 data storage registers or 64 registers of program memory. You can increase the HP-41C capacity to a maximum of 319 registers of program memory or 319 data storage registers, or *any* combination!

But That's Not All! The HP-41C has four input/output receptacles (ports). You will be able to plug in the additional program memory/storage modules as well as complete technical application modules (application ''pacs'')—even a HP-67/HP-97-compatible card reader and a thermal printer.

CAUTION

Always turn the HP-41C off before inserting or removing any plug-in extensions or accessories. Failure to do so could damage both the calculator and the accessory.

Part I

Using Your HP-41C Calculator

Getting Started

Your basic HP-41C is shipped fully equipped; the batteries will be installed by you or your dealer. If you turn on your HP-41C and the BAT annunciator in the display appears, or batteries are not installed, refer to Batteries, page 240.

Operating Keys

[ON] Key

To begin, press [ON]. The [ON] key turns the HP-41C power on and off. In order to conserve battery power, the HP-41C will automatically turn itself off after 10 minutes of inactivity. You can turn it on again by simply pressing [ON].

Each time the HP-41C is turned on, it "wakes up" in normal or USER mode; whichever was active when the calculator was turned off. If you were in PRGM (program) or ALPHA (alphabetic) mode when the calculator turned off, when you turn it back on again, these modes will not be active.

[USER] Mode Key

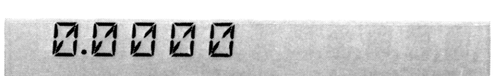
The [USER] mode key allows you to customize your HP-41C, placing functions where *you* want them on the keyboard. When you press [USER], the USER annunciator in the display turns on to let you know the calculator is in USER mode. To take the HP-41C out of USER mode, simply press [USER] again; the USER annunciator will turn off. Try it now:

Keystrokes **Display**

[USER]

Places calculator in USER mode; your customized HP-41C keyboard becomes active. USER annunciator turns on.

[USER]



Second press takes the HP-41C out of USER mode; all "normal" functions on the HP-41C keyboard become active. Annunciator turns off.

When the HP-41C is in USER mode, all keys that have not been reassigned retain their normal mode functions. ("Normal mode" means that the calculator is *not* in PRGM, ALPHA or USER mode.) Normal mode functions are the ones printed above and on the faces of the keys.

[PRGM] Mode Key

When the calculator is in PRGM mode, keystrokes are recorded as program instructions. Programming and PRGM (mode) are covered in detail in part II of this handbook.

[ALPHA] Mode Key

ALPHA mode is an exciting HP-41C feature that allows you to use both numbers and letters as well as several special characters. When you press [ALPHA], the primary keyboard functions become the alphabetic characters printed in blue on the lower face of the keys. In addition, the ALPHA annunciator will turn on to show you that the calculator is in ALPHA mode. To take the HP-41C out of ALPHA mode, simply press [ALPHA] again.

Display

Initial Display

Should you see ***MEMORY LOST*** in the display the first time you turn the HP-41C on, do not worry—it means that power to the continuous memory of the calculator has been interrupted at some time. Merely press [←] (the correction key) to clear the error, then continue. When power to continuous memory is interrupted, all information you placed into the HP-41C is lost.

Whenever the HP-41C is turned on, the display shows the number or ALPHA characters that were in the display before you turned the calculator off.

Display Capacity

The HP-41C display has 12 full character positions. You can put up to 24 characters in the display. As you place a string of ALPHA characters in the display that is larger than 11 characters, the HP-41C automatically "scrolls" the characters off to the left (more about this later). For example, place the calculator into ALPHA mode, and press the following keys:

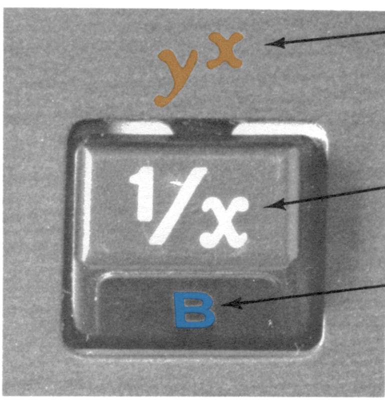
Keystrokes	Display	
[ALPHA]		Places the HP-41C into ALPHA mode and turns the ALPHA display annunciator on.
ABCDEFGHIJK	***ABCDEFGHIJK_***	The display now contains 11 full characters.
L	***BCDEFGHIJKL_***	Now 12 characters.
M	***CDEFGHIJKLM_***	Now 13 characters.
[ALPHA]	***0.0000***	Takes the HP-41C out of ALPHA mode.

The Owner's Handbook

Numbers shown in most examples and problems in this handbook are displayed to four decimal places, like this *0.0000.* As you will soon see, numbers can be displayed in a variety of formats, but if you want the HP-41C display to look like the ones shown on the next few pages, press ■ [FIX] 4 now.

Keyboard

Each key on the keyboard can perform several different functions. The particular functions that are available on the keyboard depend on the status of the calculator. If the HP-41C is in "normal" mode, that is, not in PRGM, USER, or ALPHA mode, the available functions are the ones printed on the face of the key and above the key.

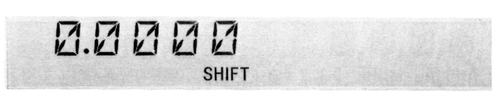


To select the function above the key, first press the gold ■ (shift) key, then press the function key.

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

The character printed in blue on the lower face of the key is only available in ALPHA mode, and not in normal mode. ALPHA mode is covered in detail later.

You can always tell when you have pressed the ■ (shift) key; a SHIFT annunciator in the display shows each time you press ■. The annunciator turns off when the shifted function is executed or if you press ■ again. The SHIFT annunciator looks like this:



Function Names

When you press and hold down a function key momentarily, a name for that function will appear in the display. When you hold the key down for longer than about a half second, ***NULL*** appears in the display. This means that the function has been cancelled. By pressing and holding a key you can look at the function name without actually executing the function! For example, compute the reciprocal of 10.

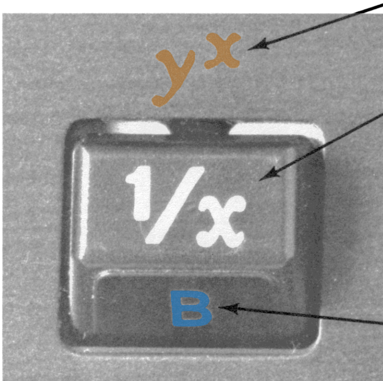
Keystrokes	Display	
10	10_	
[1/x]	1 / X	Press and hold the [1/x] key for a moment, then release it. Notice that the function name remains in the display while you hold the key down, and the function is executed when you release it.
	0.1000	

Now nullify a function by holding it down for more than about a half second.

Keystrokes	Display	
10	10_	
[1/x]	1 / X	Press and hold [1/x] until the name disappears from the display and *NULL* appears. When you release the key, the function is *not* executed.
	NULL	
	10.0000	Previous contents of the display are returned to the display.
■ [CLX]	0.0000	Clears the display.

The ALPHA Keyboard

When you place the HP-41C into ALPHA mode ([ALPHA]), a special alphanumeric keyboard becomes active. The characters printed in blue on the lower face of each key are what you get when you press a key. The functions *printed* on the face and above the keys are no longer active. In addition, an ALPHA character (not printed on the key) becomes available as a shifted key. So, when the HP-41C is in ALPHA mode...

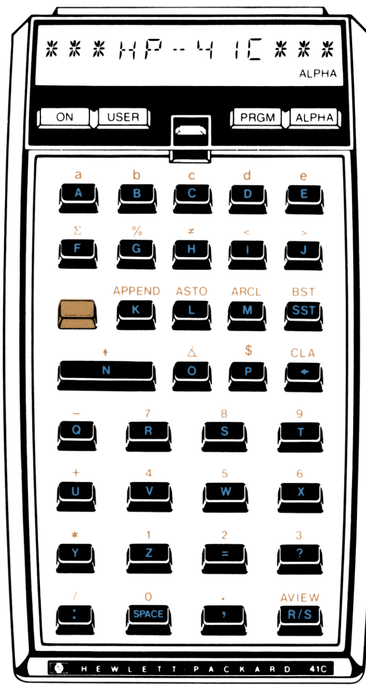


...the function printed above the key is no longer active...

...and the function printed on the face of the key is no longer active. A shifted ALPHA character is now associated with the key (but not printed on it). To select the character associated with this key, press ■ and the key (the shifted character on the illustrated key is lower case b). These shifted characters are shown on page 19.

The *primary* function of each key is now the ALPHA mode character printed in blue on the lower face of each key. To select this character, simply press the key.

Here is what the complete ALPHA keyboard looks like (for easy reference, the complete ALPHA keyboard is also reproduced on the HP-41C Quick Reference Card, on the back of the calculator and in the function index at the end of this handbook). Note that the ALPHA characters shown here on the top of the keys are not actually printed on the keys.



Let's write a word in the display to see how ALPHA mode works.

Keystrokes	**Display**	
ALPHA ■ CLA		Places HP-41C into ALPHA mode and clears the display. Primary functions are now the characters printed in blue on the lower face of each key. Shifted characters are not printed on the keys (see page 18).
F	*F_*	When you press a key, the letter printed in blue on the lower face of the key is placed in the display.
U	*FU_*	
E	*FUE_*	
L	*FUEL_*	
ALPHA	*0.0000*	Takes HP-41C out of ALPHA mode. The HP-41C remembers the string, FUEL.

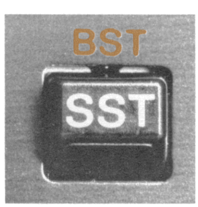
Shifted functions in ALPHA mode are shown in the illustration on page 19.

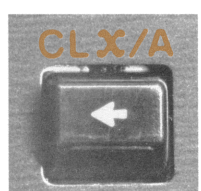
Try it now:

Keystrokes	Display	
ALPHA	**FUEL**	Places HP-41C in ALPHA mode. The string, **FUEL**, returns.
H	**H_**	Begin the new string. The previous string is lost.
P	**HP_**	H and P are primary characters.
–	**HP–_**	– is a shifted character.
4	**HP–4_**	
1	**HP–41_**	4 and 1 are shifted characters.
CLA		Blanks the display.
ALPHA	**0.0000**	Takes calculator out of ALPHA mode. The HP-41C keyboard is now in "normal" mode; all functions printed above and on the face of the keys are now active, and ALPHA mode characters are no longer available.

You can recall the ALPHA characters you have keyed into the display by pressing VIEW in ALPHA mode. This is actually the AVIEW *(ALPHA view)* function. Viewing ALPHA strings is covered in section 3.

Regardless of the mode the calculator is in, the key is always the shift function. There are two other keys on the HP-41C that always remain the same, both the function on the face of the key and the shifted function. (An exception to this is when these keys are reassigned in USER mode. This is covered in detail in section 4.) These two keys are:





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). If you want your display to be the same as those shown here, press FIX 4.

As you key in a number, notice how the HP-41C prompts you for each number with an _ (underscore).

To key in the number 30.6593:

Keystrokes	Display	
30.6593	30.6593 _	The number 30.6593, is in the display.

Numbers that you key in in ALPHA mode are only ALPHA characters and cannot be used in number operations. For example, ALPHA ■ 4 ALPHA produces the ALPHA character 4. You cannot perform arithmetic operations on ALPHA numbers.

Numbers entered in ALPHA mode are ALPHA characters and cannot be used in number functions (i.e., + , √x , LOG).

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:

Keystrokes	Display
CHS	-30.6593 _

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 negative number now in the display back to positive:

Keystrokes	Display
CHS	30.6593 _

Exponents of Ten

You can key in numbers with powers of 10 by pressing EEX *(enter exponent of 10)* followed by number keys to specify the exponent of 10. (Negative exponents are covered later.) Again, notice how the HP-41C prompts you for the number and the exponent. For example, to key in Avogadro's number (6.0222×10^{26}kmol^{-1}):

Keystrokes	Display	
■ CLX	0.0000	
6.0222	6.0222 _	The HP-41C prompts you for each digit.
EEX	6.0222 _	Then prompts you for the exponent.
2	6.0222 2 _	
6	6.0222 26	Avogadro's number (6.0222×10^{26}kmol^{-1}).

Clearing Operations

The [CLx/A] Key

[CLx/A] is a dual purpose key that is used to clear the display. When the HP-41C is in ALPHA mode and you press [CLx/A], only the [CLA] *(clear ALPHA)* function is performed. The display is *blanked* when you press [CLA] in ALPHA mode.

When the HP-41C is *not* in ALPHA mode, that is, in normal mode, pressing [CLx/A] performs only the [CLx] function. The display (X-register) is cleared to zero when you press [CLx] in normal mode. (Clearing registers is covered later, so don't worry about it yet.)

First, since we are still in normal mode, let's clear the display (X-register) to zero.

Keystrokes	Display	
	6.0222 26	The number from the previous example.
■ [CLx]	0.0000	Clears the display (X-register) to zeros.

Now, to see how [CLA] works in ALPHA mode, write the word SOLAR in the display and then clear it:

Keystrokes	Display	
[ALPHA]		Places the calculator in ALPHA mode.
SOLAR	SOLAR_	The word.
■ [CLA]		Blanks the display.
[ALPHA]	0.0000	Takes the HP-41C out of ALPHA mode.

The [←] (Correction) Key

You can delete one character at a time in the display using the [←] key. In ALPHA mode, each press of [←] deletes one right-most character. Notice how the "_" (underscore) prompt moves back. For example:

Keystrokes	Display	
[ALPHA]		Places the HP-41C in ALPHA mode.
HYDVO	HYDVO_	The example word with an error.
[←]	HYDV_	One right-most character deleted.

Keystrokes	Display	
[←]	***HYD _***	Another right most character deleted.
RO	***HYDRO _***	The corrected word.
[ALPHA]	***0.0000***	Takes the HP-41C out of ALPHA mode.

In normal mode, when you are keying in a number, you can use [←] to delete and correct digits in the number. For example, key in Joule's constant (the equivalent of a Btu in ft-lb), 778.26. Again, notice how the "_" prompt moves.

Keystrokes	Display	
778.36	***778.36 _***	Whoops, Joule's constant is 778.26.
[←]	***778.3 _***	One right most character deleted.
[←]	***778. _***	Another character deleted.
26	***778.26 _***	The correct Joule's constant.
■ [CLX]	***0.0000***	

In both ALPHA and normal modes, [←] *only works as a character-by-character correction key when the _ prompt is in the display*. If the _ prompt is not present in normal mode, then pressing [←] clears the X-register to zeros (like a [CLX]). The [←] key always deletes one character at a time when you are keying in ALPHA characters.

The [←] key can be used in many different situations to aid you in correction of entries and error recovery. You will learn more about the [←] function as you progress through the handbook.

To clear the entire calculator (all programs, registers, assignments, flags, etc.) with the "master clear:" turn the HP-41C off, hold down the [←] key, and turn the calculator back on again. The display will show ***MEMORY LOST***.

Functions

In spite of the large number of functions available in the HP-41C, you will find all functions simple to execute:

- When you press and release a function key, the calculator immediately executes that function.

- When you press and hold a function key for less than about a half second, the calculator displays the function name and then executes the function when you release the key.
- When you press and hold a function key for more than about a half second, the calculator first displays the function name and then displays **NULL**. The function is *not* executed when you release the key.

For example, to calculate the number of square meters in a 160-meter square field (160m × 160m or 160^2):

Keystrokes	Display	
160	160_	
■ [x^2]	25,600.0000	The answer.

Now, to find the square root of that result:

Keystrokes	Display	
	25,600.0000	Number from previous operation.
[$\sqrt{x}$]	160.0000	The answer.

[$\sqrt{x}$] and [x^2] are examples of one-number function keys; that is, keys that execute upon a single number. All standard HP-41C functions operate upon either one number or two numbers at a time (except the statistics functions like [Σ+] and [Σ-] —more about these later).

One-Number Functions

To use any one-number function:

- Key in the number.
- Execute the function.

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

Keystrokes	Display	
4	4_	
[$1/x$]	0.2500	When you press and release [$1/x$], the function is executed.

Here are some more one-number function problems. Remember, first key in the number, then execute the function.

1/25	=	**0.0400**	(25 [1/x])
$\sqrt{360}$	=	**18.9737**	(360 [√x])
10^4	=	**10,000.0000**	(4 [■] [10^x])
log 8.31434	=	**0.9198**	(8.31434 [LOG])
71^2	=	**5,041.0000**	(71 [■] [x^2])

Two-Number Functions

Two-number functions must have two numbers present in order for the operation to be performed. Both numbers must be in the calculator before the function is executed. [+], [−], [×], and [÷] are examples of two-number functions.

When you must key in two numbers before performing an operation, use the [ENTER↑] key to separate the two numbers.

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

If you need to key in only one number for a function, you do not 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. Execute the function.

For example, to add 15 and 5:

Keystrokes	**Display**	
15	**15_**	The first number.
[ENTER↑]	**15.0000**	Separate the first number from the second.
5	**5_**	The second number.
[+]	**20.0000**	The function and answer.

Other arithmetic functions are performed the same way:

To Perform	**Keystrokes**	**Display**
15 − 5	15 [ENTER↑] 5 [−]	**10.0000**
15 × 5	15 [ENTER↑] 5 [×]	**75.0000**
15 ÷ 5	15 [ENTER↑] 5 [÷]	**3.0000**

The [y^x] function 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:

1. Key in the first number.
2. Press [ENTER↑] to separate the first number from the second.
3. Key in the second number (the power).
4. Execute the function (press [■] [y^x]).

When working with any function (including [y^x]), you should remember that the displayed number is designated x by the function symbols.
So [$\sqrt{x}$] means square root of the displayed number, [$1/x$] means 1/displayed number, etc.

Let's solve a problem using the [y^x] function. Calculate 4^7:

Keystrokes	**Display**
4	***4_***
[ENTER↑]	***4.0000***
7	***7_***
[■] [y^x]	***16,384.0000***

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

16^4 (16 to the 4th power) =	***65,536.0000***	(16 [ENTER↑] 4 [■] [y^x])
2^{15} (2 to the 15th power) =	***32,768.0000***	(2 [ENTER↑] 15 [■] [y^x])
81^2 (81 squared) =	***6,561.0000***	(81 [ENTER↑] 2 [■] [y^x]) (You could also have done this as a one-number function using [x^2] .)

Chain Calculations

The speed and simplicity of the Hewlett-Packard logic system is 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 (covered in detail in section 3) stores up to four intermediate results inside the HP-41C 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 (17 − 5) × 4.

If you were working the problem with pencil and paper, you would first calculate the intermediate result of (17 − 5)...

$$\cancel{(17 - 5)} \times 4 =$$
$$12$$

...and then you would multiply the intermediate result by 4.

$$\cancel{(17 - 5)} \times 4 =$$
$$12 \times 4 = 48$$

Work through the problem exactly the same way with the HP-41C, one operation at a time. You solve for the intermediate result first...

$$(17 - 5)$$

Keystrokes	Display	
17	**17_**	
[ENTER↑]	**17.0000**	
5	**5_**	
[−]	**12.0000**	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. Complete the problem now by multiplying the intermediate result by 4.

Keystrokes	Display	
	12.0000	The intermediate result is in the display.
4	**4_**	The intermediate result is automatically stored in the HP-41C when you key in this number.
[×]	**48.0000**	Pressing the function ([×]) multiplies the new number and the intermediate result, giving the final answer.

Because the HP-41C stores intermediate results automatically, you don't need to remember them.

Now try these problems. You don't need to clear the display before you start each problem—the HP-41C uses only the numbers for the current problem.

To Solve	Keystrokes	Display
$(5 + 11) \div 8$	5 [ENTER↑]	5.0000
	11 [+]	16.0000
	8 [÷]	2.0000
$(23 \times 6) \div 12$	23 [ENTER↑]	23.0000
	6 [×]	138.0000
	12 [÷]	11.5000
$(9 + 17 - 4 + 23) \div 4$	9 [ENTER↑]	9.0000
	17 [+]	26.0000
	4 [−]	22.0000
	23 [+]	45.0000
	4 [÷]	11.2500

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 $(6 + 5) \times (9 - 3)$ with pencil and paper, you would:

$$(6+5) \times (9-3)$$

First solve for the contents of these parentheses...

...and then for these parentheses...

...and then you would multiply the two intermediate answers together.

You work through the problem the same way with your HP-41C. First solve for the intermediate result of $(6 + 5)$:

Keystrokes	Display	
6 [ENTER↑]	6.0000	
5 [+]	11.0000	Intermediate result.

Now perform $(9 - 3)$: (Since you must 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.)

Keystrokes	Display	
9 [ENTER↑]	9.0000	
3 [−]	6.0000	Intermediate result.

Next, multiply the intermediate answers together for the final answer.

Keystrokes	Display	
[×]	66.0000	The final answer.

Notice that you didn't have to write down or remember the intermediate answers before you multiplied. The HP-41C 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.

Now try these problems. Remember to work through them as you would with pencil and paper, but don't worry about intermediate answers—they are handled automatically by the HP-41C.

$$(16 \times 38) - (13 \times 11) = 465.0000$$
$$(27 + 63) \div (33 \times 9) = 0.3030$$
$$(\sqrt{(16.38 \times 5)}) \div .05 = 180.9972$$
$$4 \times (17 - 12) \div (10 - 5) = 4.0000$$

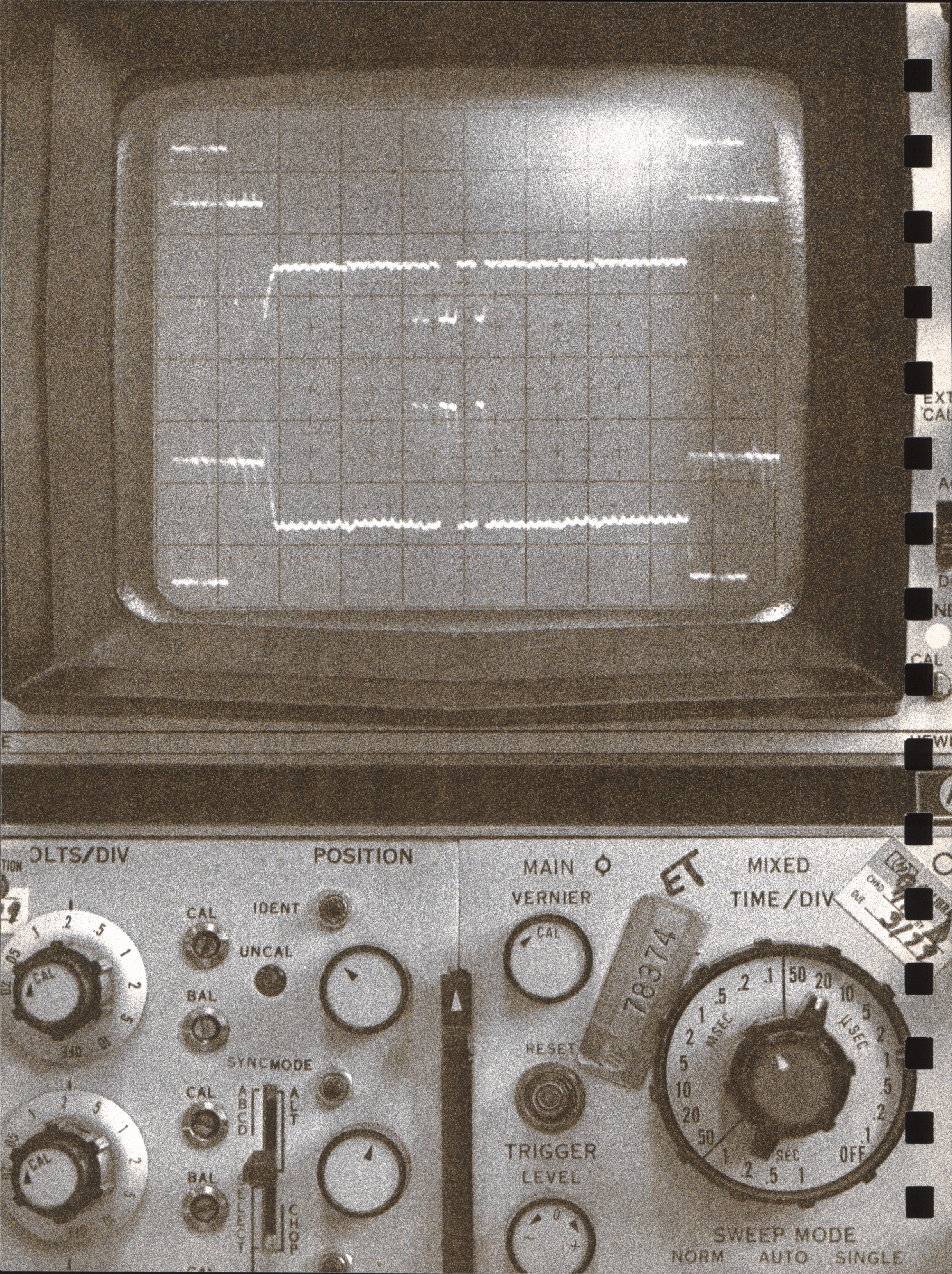
Before You Continue...

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

It is the RPN system that gives you all of these advantages while you are using the HP-41C:

- You work with only one function at a time. The HP-41C cuts problems down to size instead of making them more complex.
- Functions are executed immediately. 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. You don't have to write down long intermediate answers when you work a problem.
- You can calculate in the same manner you do with pencil and paper. You don't have to think the problem through ahead of time.
- There is no need to worry about parentheses in the calculation; RPN eliminates the necessity for entering parentheses.

The Hewlett-Packard RPN system takes just a few minutes to learn. But you'll be amply rewarded by the ease with which you and your calculator will glide through the longest, most complex equations. With the HP-41C, the investment of a few moments of learning yields a lifetime of mathematical dividends. Work carefully through this handbook so you can get the greatest value from your new HP-41C.


POSITION
IDENT
CAL
UNCAL
BAL
SYNC MODE
ALT
CHOP
MAIN
VERNIER
RESET
TRIGGER
LEVEL
ET
78374
MIXED
TIME/DIV
MSEC
μSEC
SEC
OFF
SWEEP MODE
NORM AUTO SINGLE

Section 2

Display Control

The HP-41C provides many display capabilities for both numbers and ALPHA characters. You can control the format of how all numbers are seen in the display. But regardless of the display options in effect, the HP-41C always represents each number internally as a 10-digit mantissa and a 2-digit exponent of 10. Thus when the calculator is set to display only four digits past the decimal point, the fixed constant pi, which is always represented internally as $3.141592654 \times 10^{00}$, will appear in the display as 3.1416.

For example, when you compute 2π, you might *see* the answer to only four decimal places:

Keystrokes	Display
2 ■ [π] [×]	6.2832

However, inside the calculator all numbers have 10-digit mantissas and 2-digit exponents of 10. So the calculator *actually* calculates using full 10-digit numbers:

$2.000000000 \times 10^{00}$ ■ [π] $3.141592654 \times 10^{00}$ [×]

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

$$\underbrace{6.2832}_{\text{You see only these digits...}}\underbrace{85308 \times 10^{00}}_{\text{...but these digits are also present internally.}}$$

Display Format Control

There are three functions, [FIX], [SCI], and [ENG], that allow you to control the manner in which numbers appear in the HP-41C display.

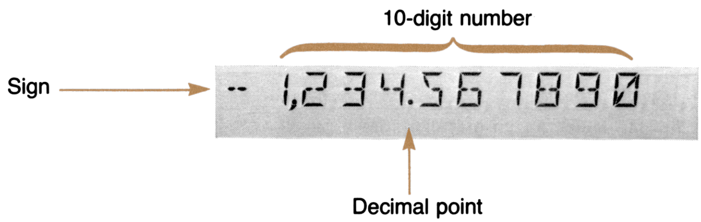
[FIX] displays numbers in fixed decimal point format, while [SCI] permits you to view numbers in scientific notation format. [ENG] displays numbers in engineering notation, with exponents of 10 shown in multiples of three (e.g., 10^3, 10^{-6}, 10^{12}). By pressing a digit key (0 through 9) after any of these display control functions, you specify the number of decimal digits displayed. The HP-41C will actually *prompt* you with an _ (underscore) for the number (0 through 9) when you press the display format function.

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

When you specify a display mode by pressing [FIX], [SCI], or [ENG], the Continuous Memory of the HP-41C "remembers" that format. The format remains the same until you change it; even while the HP-41C is turned off.

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) after the decimal point. The HP-41C will prompt you with *FIX _* to let you know that you must next enter a digit.



Let's put a number in the display so you can see how fixed-point display looks:

Keystrokes	Display	
2.24136	*2.24136 _*	The number.
■ [FIX]	*FIX _*	The display shows the function ([FIX]) and prompts you for the digit with an _.
2	*FIX 2*	When you satisfy the prompt, the display shows the function when you hold the 2 key down briefly...
	2.24	...and then shows the actual format when you release the key. The number is rounded off to two decimal places. Internally, however, the number maintains its original value of $2.241360000 \times 10^{00}$.
■ [FIX]	*FIX _*	The function and prompt.
0	*2.*	The number is rounded off to 0 decimal places.
■ [FIX] 9	*2.241360000*	The formatted number. Trailing zeros are added to fill out the full nine decimal places.
■ [FIX] 4	*2.2414*	The display rounds upward if the first *hidden* digit is 5 or greater.

Later, in section 14, you will learn how to control the way commas and decimal points are used in displayed numbers. In FIX format, the HP-41C normally displays numbers with commas separating groups of numbers like this: ***99,187,224.00***. The HP-41C can also display numbers without the comma separators, like this: ***99187224.00***. For European users, the format can even be changed to display numbers with separators and decimal notation like this: ***99.187.224,00*** or without separators, like this: ***99187224,00***. If you wish to change the way your HP-41C presently displays numbers, turn to section 14 and read about the decimal point flag and the digit grouping flag.

Scientific Notation Display

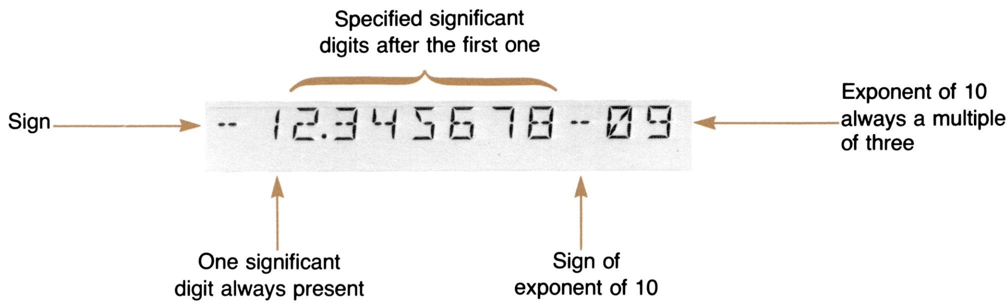
In scientific notation each number is displayed with a single digit to the left of the decimal point. This number is followed by a specified number of digits (up to 7) to the right of the decimal point and multiplied by a power of 10. The calculator prompts you for the decimal digit specification with ***SCI _***.

Scientific notation is selected by pressing ■ SCI followed by a digit key to specify the number of decimal places to which the number is rounded. For example, place the speed of light (299,792,500 m/s) in the display and set the calculator to scientific notation.

Keystrokes	Display	
299792500	***299,792,500 _***	Speed of light in a vacuum.
■ SCI	***SCI _***	The function and the prompt.
3	***2.998 08***	Indicates 2.998×10^8. Notice that the display rounds if the first *hidden* mantissa digit is 5 or greater.
■ SCI 0	***3. 08***	Indicates 3×10^8.

Engineering Notation Display

Engineering notation is similar to scientific notation except that engineering notation shows all exponents as 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 [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, key in 28.17939×10^{-16} and change the number of significant digits displayed to see what happens to the number. Remember that the HP-41C prompts you (with *ENG _*) for the number of significant digits.

Keystrokes	Display	
28.17939	*28.17939 _*	
[EEX] [CHS] 16	*28.17939 -16*	The number.

Keystrokes	Display	
■ ENG	ENG_	The display function and prompt.
2	2.82 -15	Engineering notation display. Number appears in the display rounded off to two significant digits after the omnipresent first one. Power of 10 is proper multiple of three.
■ ENG 3	2.818 -15	Display is rounded off to third significant digit after the first one.
■ ENG 0	3. -15	Display is 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 maintain the exponent of 10 as a multiple of three. For example, multiplying the number now in the calculator by 10 twice causes the decimal point to shift to the right twice without altering the exponent of 10:

Keystrokes	Display	
■ ENG 2	2.82 -15	The number.
10 ×	28.2 -15	The decimal point shifts.
10 ×	282. -15	

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 again.

Keystrokes	Display	
10 ×	2.82 -12	The decimal point shifts. Power of 10 shifts to 10^{-12}. Display maintains two significant digits after the first one.
■ CLX	0.00 00	Clears the display.
■ FIX 4	0.0000	Sets the calculator back to FIX 4.

Automatic Display Switching and Scrolling

The HP-41C automatically switches the display from fixed point notation to scientific notation whenever the number is too large or too small to be seen with a fixed decimal point. This keeps you from missing unexpectedly large or small answers.

After automatically switching from fixed point to scientific notation, the display automatically reverts back to the fixed point display originally selected when new numbers come into the display. Note that automatic switching occurs only between fixed and scientific notation displays—engineering notation display must be selected with the ENG .

Any time the HP-41C must display a single line of information which exceeds the 12-character display, the calculator automatically "scrolls" the line through the display to the left so that you can see the complete line.

Annunciators

The HP-41C display contains seven "annunciators" or key words that tell you the status of the calculator. Each annunciator tells you something about how the calculator is operating at that moment.

BAT USER GRAD SHIFT 0 1 2 3 4 PRGM ALPHA

BAT (Battery) Low Power Annunciator

If the BAT annunciator is displayed, this means that you have about 5–15 days of operating time left (using alkaline batteries). The best thing to do when BAT turns on is to put HP-41C batteries on your shopping list. (Refer to Batteries, page 240).

USER Mode Annunciator

When you press the USER key to set the HP-41C to USER mode, the USER annunciator in the display turns on. This lets you know that your customized keyboard has become active. Functions that you have assigned to the keyboard become active and the normal functions on those keys are no longer active. For an introductory discussion of USER mode, turn to Operating Keys on page 15. USER mode is covered in detail in section 4.

GRAD-RAD Mode Annunicator

When you execute the GRAD function, the HP-41C is placed in GRADs trigonometric mode and the GRAD annunciator turns on. When you execute the RAD function, the HP-41C is placed in RADians mode and the RAD portion of the display annunciator turns on. Function execution is covered in section 4 and trigonometric modes are covered in detail in section 6.

01234 Flag Status Annunciators

If flags 0, 1, 2, 3, or 4 are set in a program or from the keyboard, the corresponding display annunciator turns on. A flag annunciator showing in the display indicates a set (true) flag. Don't worry about flags yet—they are covered in detail later in this handbook.

Shift Key Annunciator

Any time you press the ■ (shift) key, the **SHIFT** annunciator turns on. The annunciator turns off again when you press ■ or the shifted function is executed.

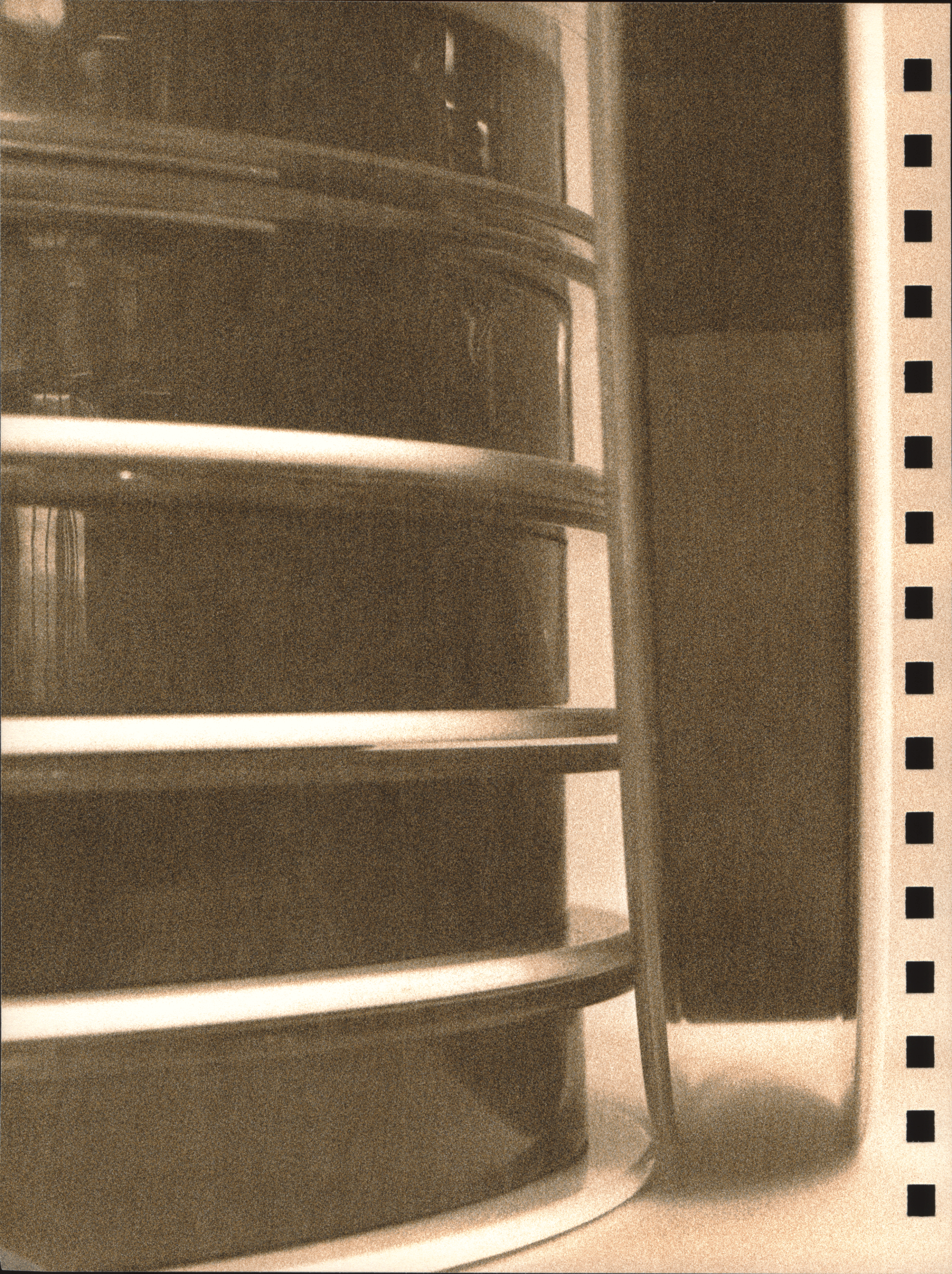
PRGM (Program) Mode Annunciator

Pressing [PRGM] places the HP-41C into PRGM (program) mode and turns on the **PRGM** display annunciator. Pressing [PRGM] again takes the calculator out of PRGM mode and turns off the annunciator. PRGM mode and programming are covered in part II of this handbook, so don't be concerned about the **PRGM** annunciator now.

ALPHA (Alphabetic) Mode Annunciator

When you place the HP-41C into ALPHA mode by pressing [ALPHA], the **ALPHA** annunciator turns on. When the **ALPHA** annunciator is turned on, you are assured that the ALPHA keyboard is active. The ALPHA keyboard was covered in section 1, pages 18-20.

The convenience of the HP-41C display annunciators allow you to concentrate on the problem at hand—there is no need to remember the status of the calculator. Just look at the display; you can immediately see all HP-41C operation conditions.



Section 3

Automatic Memory Stack and ALPHA Register

This section covers the detailed operation of the automatic memory stack and the ALPHA register. If you wish to learn how the stack and ALPHA register work, and how you can take advantage of some of the more powerful features of the HP-41C, we suggest that you work through this section. Otherwise, you may wish to skip this section for now and continue with section 4, Using HP-41C Functions.

The Automatic Memory Stack

Automatic storage of intermediate results is the reason that the HP-41C makes solution of even the most complex equations simple. The automatic storage is made possible by the Hewlett-Packard memory "stack."

Here is what the registers of the automatic memory stack look like:

The	T	*0.0000*	
Automatic	Z	*0.0000*	
Memory Stack	Y	*0.0000*	
	X	*0.0000*	(Displayed)

When you are in normal mode, that is, not in PRGM, USER, or ALPHA mode, numbers that appear in the display are the same as the contents of the X-register in the calculator.

Each register in the stack holds a 10-digit number and its 2-digit exponent of 10. ALPHA characters and their relationship to the stack are covered later. For now, let's work with just numbers.

Basically, numbers are stored and manipulated in the HP-41C "registers." Each number, no matter how few digits (e.g., 0, 1, 5) or how many (e.g., 3.14159265, −15.78352, or 1.7588028×10^{11}), occupies one entire register. We label these registers X, Y, Z, and T. They are "stacked," like shelves, one on top of the other, with the X-register on the bottom and the T-register on top.

The contents of these registers, as well as *all* other information in the HP-41C, are maintained by the calculator's Continuous Memory. Even when the HP-41C is turned off, the values stored in the stack registers are all "remembered" by the calculator.

When you execute a function, the result is always placed in the X-register (the display). So when you compute the reciprocal of 5...

Keystrokes	Display
5 [1/x]	*0.2000*

...the result is placed in the X-register and is seen in the display. The contents of the stack registers now look like this:

T	0.0000	
Z	0.0000	
Y	0.0000	
X	0.2000	(Displayed)

The Display and ALPHA Register

We have just seen how you can execute a function and how the result is placed in the X-register and seen in the display.

But if you are in ALPHA mode, any characters you key in are placed into a special ALPHA register as well as the display. The ALPHA register is *separate* from the automatic memory stack. The automatic memory stack is *not* disturbed when you key in ALPHA characters.

To see what is in the ALPHA register, simply place the HP-41C into ALPHA mode. In ALPHA mode, the ALPHA register is always displayed.

The ALPHA register can hold up to 24 characters, 12 more than the display. In any combination of full characters and periods, colons and commas, the largest number of characters you can place in the ALPHA register is 24.

When you key in a string of ALPHA characters longer than the display (12 characters), the HP-41C automatically "scrolls" the characters through the display to the left. If at *any* time you wish to see the complete contents of the ALPHA register, simply press ■ AVIEW *(ALPHA view)* in ALPHA mode.

Try it now:

Keystrokes	Display	
ALPHA		
SCROLL EXAM	SCROLL EXAM_	Watch how the HP-41C scrolls characters off the left of the display.
P	CROLL EXAMP_	
L	ROLL EXAMPL_	
E	OLL EXAMPLE_	
■ AVIEW	SCROLL EXAMP	AVIEW lets you view the entire string any number of times.
	CROLL EXAMPL	
	ROLL EXAMPLE	
■ CLA		Blanks the display.
ALPHA	0.2000	The X-register is again displayed.

The APPEND function (■ K in ALPHA mode) enables you to build on to a string in the ALPHA register. You can add characters to a string already in the ALPHA register by placing the HP-41C into ALPHA mode, pressing APPEND (■ K) and then key in the desired additional characters.

Try using APPEND now:

Keystrokes	Display	
ALPHA		
ADD	ADD_	The initial string.
ALPHA	0.2000	Takes the HP-41C out of ALPHA mode.
ALPHA	ADD	Places the HP-41C back into ALPHA mode.
■ APPEND (■ K)	ADD_	This enables you to continue adding characters to the string into ALPHA register.
ITION	ADDITION	The entire string.
ALPHA	0.2000	

If you don't press APPEND before adding new characters, the new characters will clear the previous string from the ALPHA register. For example:

Keystrokes	Display	
ALPHA	ADDITION	The old string.
RUN	RUN_	The new string clears the previous string from the ALPHA register.
ALPHA	0.2000	

Function Names and the Display

Each time you press and hold a function key for a moment, a name describing that function appears in the display. When you release the key, the name goes away and the function is executed.

If you press and hold a function key for longer than about a half second, the name will appear, and then will be replaced by the word ***NULL***. ***NULL*** indicates that the function has been nullified and will *not* be executed when you release the key. This allows you to preview function names and quickly recover from keystroke errors.

Keys that you use to key in numbers (CHS, EEX, • and 0 through 9) and ALPHA characters do not prompt with a name in the display. These keys execute when the key is pressed—they cannot be nullified.

Clearing the ALPHA- and X-Registers

You can clear the contents of the ALPHA register in ALPHA mode by pressing ■ CLA. CLA *(clear ALPHA)* clears the ALPHA register and leaves the automatic memory stack intact.

When the calculator is in *normal mode*, pressing ■ [CLX] *(clear X)* clears the X-register and display to zeros.

For example, the stack (automatic memory stack) now looks like this (with data intact from the previous example):

T	0.0000	
Z	0.0000	
Y	0.0000	
X	0.2000	(Displayed)

Pressing ■ [CLX] now clears the X-register (display). Notice that the function name appears when you press and hold [CLX] for a moment.

Keystrokes	Display	
■ [CLX]	CLX 0.0000	When you press and hold the key for a moment, the function name appears in the display.

T	0.0000	
Z	0.0000	
Y	0.0000	
X	0.0000	(Displayed)

Editing Display Entries

The [←] *(correction)* function lets you "back up" when you have made an error. Simply stated, any time you make an error and you want to remove numbers or ALPHA characters (ALPHAs) you keyed in, or numbers left by a function, press [←].

If the _ (underscore) prompt follows any ALPHA string or number in the display, you can delete one character or digit at a time from that string or number by pressing [←]. If the _ prompt does *not* follow the displayed ALPHA string or number, pressing [←] clears the display.

While you are keying in numbers in normal mode, pressing [←] deletes one right-most number at a time. [←] clears the display to zeros if you delete all of the numbers in the display.

For example, key in a number, edit it, and then delete it entirely using [←]. Note the movement of the _ prompt.

Keystrokes	Display	
5.6	**5.6_**	The number and _ prompt.
[←]	**5._**	One right-most digit deleted.
7	**5.7_**	The edited number.
[←] [←]	**5_**	Delete the 7 and the decimal point.
[←]	**0.0000**	Deleting the last number clears the X-register to zeros.

While you are keying in ALPHAs, pressing [←] also deletes one right-most character at a time, but *blanks* the display when you delete all of the characters. Again, notice the movement of the _ prompt.

Keystrokes	Display	
[ALPHA] ABB	**ABB_**	The ALPHA string.
[←]	**AB_**	One character deleted.
C	**ABC_**	The corrected string.
[←] [←] [←]	_	When the last ALPHA is deleted, the display is blanked, leaving the _ prompt. The stack is not disturbed.
[ALPHA]	**0.0000**	Return to normal mode.

To aid in recovering from other keystroke errors, [←] lets you clear the X-register with a single press.

Keystrokes	Display	
2 [√x]	**1.4142**	The result.
[←]	**0.0000**	The X-register has been cleared to zeros (no _ prompt was present).

And to clear a function requiring input:

Keystrokes	Display	
[RCL]	**RCL _ _**	The function and prompt.
9	**RCL 9_**	Whoops, you decide not to do this.
[←]	**RCL _ _**	The input may be deleted and changed.
[←]	**0.0000**	Or you may clear the entire operation.

Using [←] is easy and convenient, and you will learn how [←] is used in other ways for correction as you continue reading this handbook.

Manipulating Stack Contents

The [R↓] *(roll down)*, [R↑] *(roll up)*, and [x≷y] *(x exchange y)* functions allow you to review the stack contents or to move data within the stack for computation at any time. Note that [R↑] is one HP-41C function that is not on the keyboard. It is executed from the display or assigned to a key for execution. Execution of functions from the display and assigning functions to keys is covered in section 4.

Reviewing the Stack

To see how the [R↓] function works, first key in the numbers 1 through 4:

Keystrokes	Display
4 [ENTER↑]	***4.0000***
3 [ENTER↑]	***3.0000***
2 [ENTER↑]	***2.0000***
1	***1 _***

So the stack now looks like this:

T	***4.0000***	
Z	***3.0000***	
Y	***2.0000***	
X	***1 _***	(Displayed)

Now press [R↓]:

Keystrokes	Display
[R↓]	***2.0000***

The stack now looks like this:

T	***1.0000***
Z	***4.0000***
Y	***3.0000***
X	***2.0000***

When you press [R↓], the stack contents shift downward one register. The last number in the X-register rotates around to the T-register. When you press [R↓] again, the stack contents again roll downward one register.

Keystrokes	Display
[R↓]	3.0000

The stack now looks like this:

T	2.0000
Z	1.0000
Y	4.0000
X	3.0000

Continue pressing [R↓] until the stack returns to the original position.

Keystrokes	Display
[R↓]	4.0000
	T 3.0000
	Z 2.0000
	Y 1.0000
	X 4.0000
[R↓]	1.0000
	T 4.0000
	Z 3.0000
	Y 2.0000
	X 1.0000

Four presses of the [R↓] function will roll the stack down four times, returning the stack contents to their original registers.

The [R↑] *(roll up)* function works the same way as [R↓] except that [R↑] rolls the stack contents *up* instead of down.

Exchanging x and y

The [x≷y] *(x exchange y)* function exchanges the contents of the X- and Y-registers without changing the contents of 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	4.0000	T	4.0000
Z	3.0000	Z	3.0000
Y	2.0000	Y	1.0000
X	1.0000	X	2.0000

Try it now:

Keystrokes	Display
[x≷y]	2.0000

Notice that whenever you move numbers in the stack using one of the data manipulation functions the actual stack registers maintain their positions. Only the contents of the registers are shifted. Later, in section 6, you will learn how to exchange the X-register with any other storage register in the HP-41C.

The [ENTER↑] Key

When you key numbers into the calculator, you must tell the calculator when you have finished keying in one number and are ready to key in the next number. You do this using the [ENTER↑] key.

In addition to letting the calculator know you are finished keying in a number, pressing [ENTER↑] also moves the number into the stack. Here is what happens when you key in a number and press [ENTER↑]:

Keystrokes	Display	
987.3	987.3_	The number.
[ENTER↑]	987.3000	

First, when you key in the number, the stack is changed...

... from this ...		... to this.	
T	4.0000	T	3.0000
Z	3.0000	Z	1.0000
Y	1.0000	Y	2.0000
X	2.0000	X	987.3000

Then, when you press [ENTER↑], the number is pushed into the Y-register. The contents of the stack are then changed...

... from this ...		... to this.	
T	3.0000	T	1.0000
Z	1.0000	Z	2.0000
Y	2.0000	Y	987.3000
X	987.3000	X	987.3000

The value in the X-register is duplicated and *pushed* into the Y-register. The numbers in Y and Z are pushed up to Z and T, respectively, and the number in T is lost off the top of the stack.

Immediately after you press [ENTER↑], the X-register is prepared for a new number, and that new number writes over the number in the X-register.

Now, continue by keying in a new number.

Keystrokes	**Display**	
537.91	*537.91 _*	The new number is in the X-register.

The [ENTER↑] key has separated the digits of the first number from the digits of the second number and the stack now looks like this:

T	*1.0000*
Z	*2.0000*
Y	*987.3000*
X	*537.91*

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

Clearing the Stack

[CLST] *(clear stack)* clears each of the automatic memory stack registers to zeros. [CLST] can either be executed from the display or assigned to a key location and executed by pressing that key in USER mode. [CLST] is most useful if it is assigned to a key for USER mode execution. USER mode and display execution are both covered in section 4.

One-Number Functions and the Stack

One-number functions operate upon the number in the X-register only. The contents of the Y-, Z-, and T-registers are not affected when a one-number function is executed.

For example, key in the following numbers and execute the [√x] function:

Keystrokes	**Display**	
■ [CLX]	*0.0000*	
27.93	*27.93 _*	A number is in the X-register.
[ENTER↑]	*27.9300*	Places the number in Y-register.
167.54	*167.54 _*	A new number is in the display.
[√x]	*12.9437*	The answer is in the display and the X-register.

Here is what happens when you executed the [√x] function:

First, after you keyed in the numbers, the stack looked like this (the T- and Z-registers are shown cleared to zero for clarity):

0.0000	
0.0000	
27.9300	
167.54 _	(Displayed)

Then, when you pressed [√x], the result, the square root of the number in the X-register, was placed in the X-register (displayed).

0.0000	
0.0000	
27.9300	
12.9437	(Displayed)

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

The HP-41C performs arithmetic operations by positioning the numbers in the stack the same way you would on paper. For instance, if you wanted to add 17 and 46 you would write 17 on the paper and then write 46 underneath it, like this:

$$\begin{array}{r} 17 \\ +46 \\ \hline \end{array}$$

and then you would add, like this:

$$\begin{array}{r} 17 \\ +46 \\ \hline 63 \end{array}$$

Numbers are positioned the same way in the calculator. Here's how it is done.

Keystrokes	Display	
■ [CLX]	**0.0000**	Clears the displayed X-register.
17	**17 _**	17 is keyed into the X-register.
[ENTER↑]	**17.0000**	17 is copied from X into Y.
46	**46 _**	46 writes over the 17 in the display.
[+]	**63.0000**	The result is in X and the display.

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

Chain Calculations

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 part of a new number. Also, the stack automatically "drops" numbers into position when you perform two-number operations.

To see how it works, let's solve 21 + 38 + 19 + 53 = ?

For purposes of simplification, this example shows the stack cleared to zeros.

Keystrokes		Stack Contents	
■ CLX		0.0000	
21	T	0.0000	21 is keyed in.
	Z	0.0000	
	Y	0.0000	
	(Displayed) X	21 _	
ENTER↑	T	0.0000	21 is copied into Y.
	Z	0.0000	
	Y	21.0000	
	(Displayed) X	21.0000	
38	T	0.0000	38 is keyed in.
	Z	0.0000	
	Y	21.0000	
	(Displayed) X	38 _	
+	T	0.0000	38 and 21 are added together. The answer, 59, is in X and the display.
	Z	0.0000	
	Y	0.0000	
	(Displayed) X	59.0000	
19	T	0.0000	19 is keyed in and the 59 is automatically raised into Y.
	Z	0.0000	
	Y	59.0000	
	(Displayed) X	19 _	

Keystrokes		Stack Contents	
[+]	T	0.0000	59 and 19 are added together. The answer, 78, is in X and the display.
	Z	0.0000	
	Y	0.0000	
	(Displayed) X	78.0000	
53	T	0.0000	53 is keyed in and the 78 is automatically raised into Y.
	Z	0.0000	
	Y	78.0000	
	(Displayed) X	53 _	
[+]	T	0.0000	78 and 53 are added together. The final answer, 131, is in X and the display.
	Z	0.0000	
	Y	0.0000	
	(Displayed) X	131.0000	

After any calculation or number manipulation, the stack automatically lifts when a new number is keyed in. Because operations are performed when functions 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 $9.999999999 \times 10^{99}$). When the range of the calculator is exceeded, the HP-41C immediately indicates ***OUT OF RANGE*** in the display. Later you will learn how to instruct the HP-41C to ignore these types of overflows.

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 [■] [CLX] to clear the displayed X-register. Now, again solve 21 + 38 + 19 + 53 = ?

Keystrokes		Stack Contents	
21	T	0.0000	21 is keyed in.
	Z	0.0000	
	Y	0.0000	
	(Displayed) X	21 _	
[ENTER↑]	T	0.0000	21 is copied into Y.
	Z	0.0000	
	Y	21.0000	
	(Displayed) X	21.0000	
38	T	0.0000	38 is keyed in.
	Z	0.0000	
	Y	21.0000	
	(Displayed) X	38 _	

Keystrokes		Stack Contents	
ENTER↑	T	0.0000	38 is entered into Y. 21 is lifted up to Z.
	Z	21.0000	
	Y	38.0000	
	(Displayed) X	38.0000	
19	T	0.0000	19 is keyed in.
	Z	21.0000	
	Y	38.0000	
	(Displayed) X	19_	
ENTER↑	T	21.0000	19 is copied into Y. 21 and 38 are lifted up to T and Z respectively.
	Z	38.0000	
	Y	19.0000	
	(Displayed) X	19.0000	
53	T	21.0000	53 is keyed in.
	Z	38.0000	
	Y	19.0000	
	(Displayed) X	53_	
+	T	21.0000	19 and 53 are added together and the rest of the stack drops. 21 drops to Z and is also duplicated in T. 38 and 72 are ready to be added.
	Z	21.0000	
	Y	38.0000	
	(Displayed) X	72.0000	
+	T	21.0000	38 and 72 are added together and the stack drops again. Now 21 and 110 are ready to be added.
	Z	21.0000	
	Y	21.0000	
	(Displayed) X	110.0000	
+	T	21.0000	110 and 21 are added together for the final answer and the stack continues to drop.
	Z	21.0000	
	Y	21.0000	
	(Displayed) X	131.0000	

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 in the displayed X-register.

This automatic lift and drop of the stack give you tremendous 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:

$$\{37\times[(5\div18)+(5\times.13)]\}\div3.87$$

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

Experienced HP calculator users have learned that by starting every problem at its innermost set of 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-41C you have tremendous versatility in the order of execution.

For example, you could solve some problems by working through them in left-to-right order, but not all problems can be solved correctly this way. The best way to work any problem is to begin with the innermost parentheses and work outward. So, to solve the problem above:

Keystrokes	Display	
5 [ENTER↑]	*5.0000*	
18 [÷]	*0.2778*	Result of $(5\div18)$.
5 [ENTER↑]	*5.0000*	
.13 [×]	*0.6500*	Result of $(5\times.13)$.
[+]	*0.9278*	Result of $[(5\div18)+(5\times.13)]$.
37 [×]	*34.3278*	Result of $37\times[(5\div18)+(5\times.13)]$.
3.87 [÷]	*8.8702*	Result of $\{37\times[(5\div18)+(5\times.13)]\}\div3.87$.

LAST X

In addition to the four stack registers that automatically store intermediate results, the HP-41C also contains a separate automatic register, the LAST X register. This register preserves the value that was last in the display before the execution of a function. To place the contents of the LAST X register into the displayed X-register again, press ■ [LASTX].

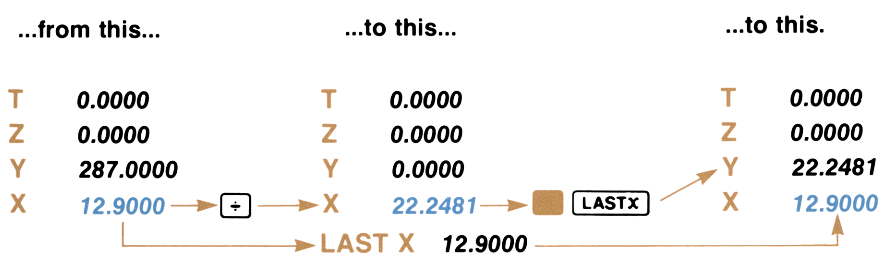
Recovering From Mistakes

[LASTX] makes it easy to recover from keystroke mistakes, such as executing the wrong function or keying in the wrong number. For example, divide 287 by 13.9 after you have mistakenly divided by 12.9.

Keystrokes	Display	
287 [ENTER↑]	287.0000	
12.9 [÷]	22.2481	Oops! The wrong number.
■ [LASTX]	12.9000	Retrieves the last entry.
[×]	287.0000	You are back at the beginning.
13.9 [÷]	20.6475	The correct answer.

Remember, if you key in the wrong digits and discover them prior to executing a function, you can use [←] to edit the number.

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



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 [LASTX], you do not have to key that number into the calculator again.

For example, calculate

$$\frac{96.704 + 52.394706}{52.394706}$$

Keystrokes	Display	
96.704 [ENTER↑]	96.7040	
52.394706 [+]	149.0987	Intermediate answer.
■ [LASTX]	52.3947	Recalls 52.394706 to the X-register.
[÷]	2.8457	The answer.

Constant Arithmetic

You may have noticed that whenever the stack drops because of a two-number operation (not a [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 (a growth factor of 1.15). If he starts a sample culture of 1000, what will be the bacteria population at the end of each day for five consecutive days?

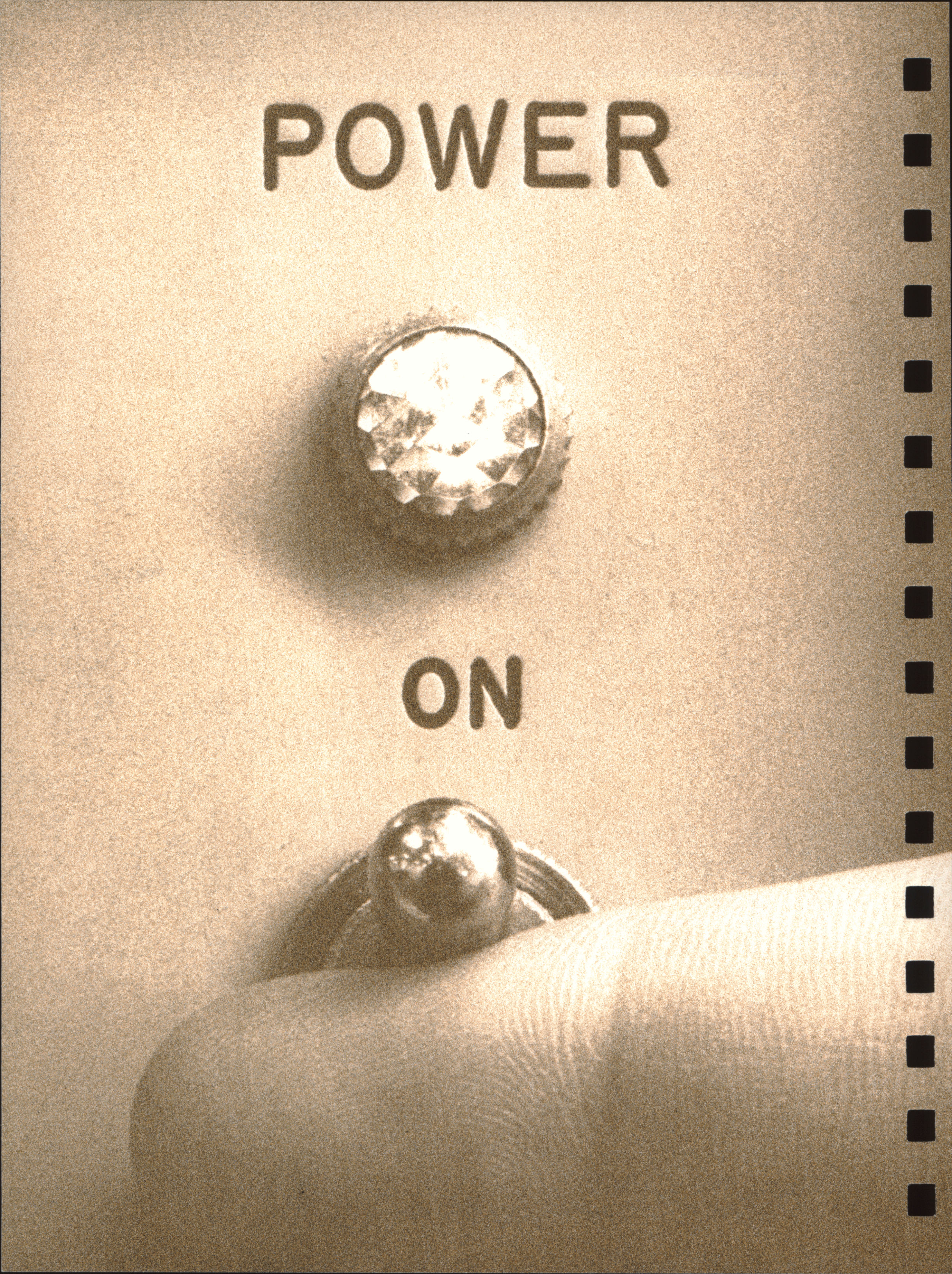
Method: Put the growth factor (1.15) in the Y-, Z-, and T-registers and put the original population (1000) in the X-register. Thereafter, you get the new daily population whenever you press [×].

Keystrokes	Display	
1.15	*1.15_*	Growth factor.
[ENTER↑]	*1.1500*	
[ENTER↑]	*1.1500*	
[ENTER↑]	*1.1500*	Growth factor is now in T.
1000	*1,000_*	Starting population.
[×]	*1,150.0000*	Population after 1st day.
[×]	*1,322.5000*	Population after 2nd day.
[×]	*1,520.8750*	Population after 3rd day.
[×]	*1,749.0063*	Population after 4th day.
[×]	*2,011.3572*	Population after 5th day.

When you press [×] the first time, you calculate 1.15 × 1000. The result (1,150) 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↓] function simply rotates the contents of the stack registers; it does not rewrite any number, but merely shifts the numbers that are already in the stack.

(This page intentionally left blank.)


POWER
ON

Section 4

Using HP-41C Functions

As you may have noticed by now, not all of the functions available in the HP-41C are printed on the keyboard. In all, the HP-41C has over 130 standard functions, 68 of which are immediately accessible by pressing function keys on the keyboard.

The rest of the HP-41C functions are accessible in different ways: from the display or by assigning them to the USER mode keyboard. You simply press [XEQ] *(execute)* and enter the function name into the display in ALPHA mode. Or even easier, you can assign the function name to a key location using the [ASN] *(assign)* function and execute the function at the press of a single key in USER mode.

With a couple of exceptions, *all* functions in the HP-41C can be executed in this manner. Section 6 lists and explains most standard HP-41C functions except for programming functions. In addition, the function index in the back of this handbook (page 267) lists *all* HP-41C standard functions.

Executing Functions From the Display

Here is how it is done. Let's compute the factorial ([FACT]) of 6. [FACT] is one of the functions not available on the normal keyboard.

To begin, key in the number 6 and press [XEQ]. When you press [XEQ], the HP-41C will place the word **XEQ** and _ (underscore) prompts in the display, like this:

Keystrokes	**Display**	
6	**6_**	The number.
[XEQ]	**XEQ __**	The HP-41C is asking: *Execute what?*

All you do now is place the name of the function you wish to execute into the display. Initially, the HP-41C prompts you with __ for a two-digit numeric label. As soon as you press [ALPHA] to enter your function name, the prompt changes to a single _, prompting you for ALPHA characters one at a time. When the HP-41C prompts you for an ALPHA character, simply press the keys associated with the desired characters. Later, in part II, you will see how to use [XEQ] to execute programs with numeric labels by specifying a label number instead of an ALPHA name.

Now, to compute the factorial of 6, key in the letters of the function name:

Keystrokes	Display	
[ALPHA]	XEQ _	Places the HP-41C into ALPHA mode.
FACT	XEQ FACT _	Tells the HP-41C that you want to execute the [FACT] *(factorial)* function.
[ALPHA]	720.0000	When you take the HP-41C out of ALPHA mode, the function in the display is executed. The answer is in the displayed X-register, just as with any other function.

Let's try another function. When you execute a function that requires some input, such as [FIX] (which requires a number from 0 through 9), the HP-41C will prompt for the input. (Note that [FIX] may also be executed directly from the keyboard.)

For example, set the calculator to [FIX] 6.

Keystrokes	Display	
[XEQ]	XEQ _ _	The HP-41C prompts: *Execute what?*
[ALPHA]	XEQ _	Place the HP-41C into ALPHA mode...
FIX	XEQ FIX _	...and spell the function name, FIX.
[ALPHA]	FIX _	The HP-41C now prompts you for the input required by [FIX].
6	720.000000	The [FIX] function is executed when you enter the required digit.

Any function requiring input, such as the [FIX] function shown above, is executed when you enter the last required digit. [FIX] requires one digit, so it executes when that digit is entered. Some other functions require two or three digits, and they are executed when the final required digit is entered.

Note that the contents of the ALPHA register are *not* disturbed when you execute a function from the display.

Function Editing and Correction

On the HP-41C, you can edit function names before you execute, or even terminate completely, using [←]. For example:

Keystrokes	Display	
[XEQ]	**XEQ _ _**	Terminate [XEQ] by pressing [←].
[←]	**720.000000**	The function is terminated and the value in X is displayed.
[XEQ]	**XEQ _ _**	Begin again.
[ALPHA] ENT	**XEQ ENT _**	Use [←] to edit the function name.
[←]	**XEQ EN _**	Characters are deleted one at a time.
G	**XEQ ENG _**	The corrected function name.
[←] [←] [←]	**XEQ _**	You are back to the [XEQ] function.
[←]	**720.000000**	Pressing [←] again terminates [XEQ] and returns the HP-41C to normal mode.
■ [FIX] 4	**720.0000**	Return to [FIX] 4.
■ [CLX]	**0.0000**	Clear the displayed X-register.

Errors

If you attempt to execute a function (using [XEQ]) whose name does not exist in the calculator, the HP-41C will display ***NONEXISTENT***. For example, if you attempt to execute ***SINE***, the calculator will display ***NONEXISTENT***. In the HP-41C, the sine function is spelled ***SIN***.

Functions that require numeric data can not operate on ALPHA characters. If a function requiring numeric data attempts to execute using ALPHA characters, the HP-41C displays ***ALPHA DATA***. A complete listing of all HP-41C error messages and their meaning is given in appendix E.

The HP-41C Catalogs

The HP-41C has three catalogs of functions. One catalog contains all functions and programs that you have written and stored in program memory. Another catalog contains all functions that become active when you plug in extensions to the HP-41C like application modules or other accessories. And the third catalog contains all of the standard HP-41C functions (this catalog contains the bulk of the functions you will be using).

The CATALOG Function

You can list the contents of *any* of the HP-41C catalogs by pressing ■ CATALOG. The calculator then prompts you for one of the following catalog numbers:

The User Catalog	CATALOG 1
The Extension Catalog	CATALOG 2
The Standard Function Catalog	CATALOG 3

When you execute the CATALOG function and specify a catalog number, it begins at the top of the specified catalog and lists the name for each function in the catalog.

Entries in the catalogs are organized as follows:

The User Catalog (1)	By top-to-bottom order in program memory. Newest programs at the bottom.
The Extension Catalog (2)	Grouped by extension.
The Standard Function Catalog (3)	Alphabetical.

To execute the CATALOG function, press ■ CATALOG. The HP-41C will prompt you for the catalog number with **CAT _**. For example, list the entire standard function catalog.

Keystrokes	Display	
■ CATALOG	**CAT _**	The HP-41C prompts: Which catalog?
3	**CAT 3**	The listing begins when you enter the catalog number.
	+	
	−	
	.	
	.	
	.	
	X↑2	
	Y↑X	The last function in the catalog.

User Catalog

As explained above, the user catalog (CATALOG 1) contains all of the programs that you have stored into program memory. CATALOG 1 also has another special capability that helps you locate programs in program memory. As the listing of CATALOG 1 progresses, the calculator is positioned to the location in program memory of the presently displayed program name. Don't be concerned with this feature now, it is covered in detail in part II of this handbook.

Stopping the Catalog Listing

You need not always list a catalog to the end. You can stop the listing at any point by pressing [R/S] *(run/stop)*. You can then use [BST] *(back step)* or [SST] *(single step)* to locate the desired function. Or you can even press [R/S] again to continue the listing.

If you wish to completely terminate the listing, press [R/S] and then [←].

Keystrokes	Display	
[■] [CATALOG] 3	+	
	−	
	.	
	.	
	.	
	GRAD	
[R/S]	GTO	Press [R/S] to stop (not terminate) the listing.
[■] [BST]	GRAD	Back step.
[SST]	GTO	Single step forward.
[SST]	HMS	Single step again.
[R/S]	HMS+	Restart the listing.
	.	
	.	
	.	
[R/S] [←]	0.0000	Press [R/S] to stop the catalog listing, then press and hold [←] to terminate the listing.

Once the [CATALOG] listing is halted (by [R/S]), pressing any other function terminates the catalog listing and the pressed function is executed.

Pressing and holding any key other than [R/S] or [ON] while the catalog is running slows the catalog listing down for viewing. The key pressed is not executed.

USER Mode Functions

You may remember from the brief descriptions in sections 1 and 2 that USER mode allows you to customize your HP-41C. USER mode lets you place functions on the keyboard where *you* want them. The way this is accomplished is through the use of the [ASN] *(assign)* function. Using [ASN], you specify a function name and a keyboard location for that function name. Once a function is assigned to a keyboard location, all you do to execute it is to place the HP-41C into USER mode and press the reassigned key.

The only key *locations* that cannot be reassigned are: ■, ON, USER, PRGM, and ALPHA. Any function that appears in the CATALOG can be reassigned to any location. Numbers and ALPHA characters, however, cannot be reassigned. The ALPHA mode functions (AVIEW, ASTO, ARCL) *can* be reassigned to the keyboard for execution in USER mode.

If you attempt to assign (using ASN) a function whose name does not exist in the calculator, the HP-41C will display ***NONEXISTENT***. The ASN function cannot be recorded as an instruction in program memory.

There are 68 key locations that can be reassigned. To assign or reassign a function to a key location:

1. Press ■ ASN. The HP-41C prompts you for the function name with ***ASN _***.
2. Press ALPHA to place the HP-41C in ALPHA mode and enter the name of the function you wish to assign.
3. Press ALPHA to place the HP-41C back into normal mode.
4. Press the key (or ■ and the key) to which you wish the function assigned. If you hold the key down, the display will show function name and the reassigned key by keycode.

Keycodes are a row-column identification of a key location. For example, the keycode for the LN key is 15. The 1 indicates the first row and the 5 indicates the fifth key.

The keycodes for shifted key locations are keycodes prefixed with a − (minus sign). For example, the keycode for the e^x key (shifted LN) is −15. The − indicates a shifted key, the 1 indicates the first row, and the 5 indicates the fifth key.

For example, assign the MEAN function to the $\sqrt{x}$ key.

Keystrokes	Display	
■ ASN	*ASN _*	The HP-41C prompts: *Assign what?*
ALPHA	*ASN _*	Places the HP-41C into ALPHA mode.
MEAN	*ASN MEAN _*	The function name you wish to assign to a key location.
ALPHA	*ASN MEAN _*	The HP-41C prompts: *To which key?*
$\sqrt{x}$	*ASN MEAN 13*	Press and hold $\sqrt{x}$ for a moment to see the assignment. MEAN is now assigned to row 1 column 3, the $\sqrt{x}$ key location.

When you reassign a function to a key location, you may wish to write the function name in the appropriate place on an overlay (provided with your HP-41C) and place the overlay on the keyboard. Also included with your new HP-41C are pre-printed sticky-back labels printed with the name of each standard HP-41C function. When you assign one of these functions to the keyboard, simply place the pre-printed label in the appropriate place on an overlay. This will help you remember where you have placed functions on your customized HP-41C.

In addition, the calculator itself helps you remember the names and locations of reassigned functions! When you press and hold a reassigned key in USER mode, the *reassigned function name* appears in the display as a reminder.

> **Note:** Key assignments of standard 41C functions are stored in program memory and use registers allocated to program memory.

Returning to the Normal Mode Function

To reassign a key to its original normal mode function, simply press ■ [ASN] [ALPHA] [ALPHA] and that key. For example, in section 1, you assigned a HEAT program to the [Σ+] key. To return the [Σ+] function to that key:

Keystrokes	Display	
■ [ASN]	ASN _	The HEAT program is no longer assigned to the [Σ+] key. [Σ+] is now the accumulation function in both USER and normal modes.
[ALPHA] [ALPHA]	ASN _	
[Σ+]	0.0000	

Using Reassigned Functions

Any function you have reassigned to a key location may be used when the HP-41C is placed into USER mode. When you press [USER], all functions you have assigned or reassigned to the keyboard become active. The standard functions located in those key locations are no longer available. If a key location has not been reassigned, it retains its normal function in USER mode.

Let's try a sample problem. In the previous example, you assigned the [MEAN] function to the [√x] key.

Cross-country runner Joel Dimor is training for a 26-mile marathon in Mexico City. Joel knows that the pace he sets for the race will determine how well he holds up in the final miles. He decides to run 10 miles five different times to see how well he paces himself. Below is a summary of the timings from the five runs.

	1st 10-Mile Run	2nd 10-Mile Run	3rd 10-Mile Run	4th 10-Mile Run	5th 10-Mile Run
Timing in Minutes	52.60	53.55	51.25	50.65	48.76

Using the following keystrokes, determine the average (mean) timing for the five runs. (Don't be concerned quite yet with how the [Σ+] function works, it is covered in detail in section 6.) Place the HP-41C into USER mode by pressing [USER]. This lets you use the [MEAN] function that is assigned to the [√x] key location.

Keystrokes	Display	
[USER]	*0.0000*	Places HP-41C in USER mode. All customized reassigned functions become active. Notice that the USER annunciator in the display is also turned on.
■ [CLΣ]	*0.0000*	
52.6000 [Σ+]	*1.0000*	
53.5500 [Σ+]	*2.0000*	
51.2500 [Σ+]	*3.0000*	
50.6500 [Σ+]	*4.0000*	
48.7600 [Σ+]	*5.0000*	
[MEAN] ([√x])	*MEAN*	Press and hold the key for a moment. Notice how the HP-41C shows the name of the function *assigned* to that key location and not the name of the function printed on the key.
	51.3620	Joel ran an average of 51.3620 minutes in his five ten-mile runs (that's about 5.1 minutes per mile); a pretty good pace for the marathon.
[USER]	*51.3620*	Takes the HP-41C out of [USER] mode. All normal key functions now become active on the keyboard.

The [MEAN] function remains assigned to the [√x] key location in USER mode until you change the key assignment. This exciting feature of the HP-41C allows you to customize your calculator by assigning the functions you use the most to the USER mode keyboard. And you can always use the normal key functions by simply pressing [USER] again to place the HP-41C back into normal mode.

Note: When you assign a function to a key location, it remains there until you change it by assigning a new function to that location.

If you turn your HP-41C off while it is in USER mode, when you turn it back on again, it *remains* in USER mode. This lets you customize your HP-41C and use the customized keyboard easily.*

---

* Execution of the *normal* mode functions on the top rows of keys in *USER* mode may take several seconds. To shorten this time, press ■ [GTO] [•] [•]. The reason for this is covered in detail in part II.

Section 5

Storing and Recalling Numbers and ALPHA Strings

Your HP-41C comes standard with 63 storage registers. You can add memory modules to increase the number of registers to a total of 319.

In the HP-41C, program memory also uses storage registers for the storage of program instructions. In fact, you can control the amount of memory space that is allocated to both data storage registers and program memory. You will learn how to control these allocations later in this section. *When you first turn the HP-41C on, it has 17 primary storage registers and 46 registers of program memory.*

The HP-41C data storage registers allow you to manually store and recall numbers and ALPHA strings for use in later calculations or in programs. These registers are independent of the automatic memory stack and LAST X registers. Like most functions, you can use these storage registers either from the keyboard or as part of a program. All information in the storage registers is preserved by the Continuous Memory of the calculator.

The diagram below shows all potential data storage registers. *This diagram shows data storage registers in their maximum memory allocation.* Remember, unless you have added additional memory modules, your HP-41C has up to 63 primary storage registers. The addresses of the primary storage registers are indicated by the numbers 00 through 99. The addresses of the extended storage registers are indicated by the numbers (100) through (318).

Automatic Memory Stack		Primary Data Storage Registers		Extended Data Storage Registers	
T		R_{00}	The standard HP-41C has 63 (R_{00}-R_{62}) primary storage registers.	$R_{(100)}$	You can add up to four memory modules, bringing the total to 100 primary and 219 extended storage registers.
Z	ALPHA	R_{01}		$R_{(101)}$	
Y		R_{02}		$R_{(102)}$	
X	LAST X	.		.	
		.		.	
		R_{62}		.	
		R_{63}		.	
		R_{64}		.	
		.		.	
		.		.	
		R_{99}		$R_{(318)}$	

If all memory modules were allocated to storage registers, each additional module would account for the following register addresses:

Standard: R_{00}-R_{62}
Module 1: R_{63}-R_{99}, $R_{(100)}$-$R_{(126)}$
Module 2: $R_{(127)}$- $R_{(190)}$
Module 3: $R_{(191)}$- $R_{(254)}$
Module 4: $R_{(255)}$- $R_{(318)}$

Storing and recalling numbers and ALPHA strings in the extended storage registers is explained in section 13 (page 197).

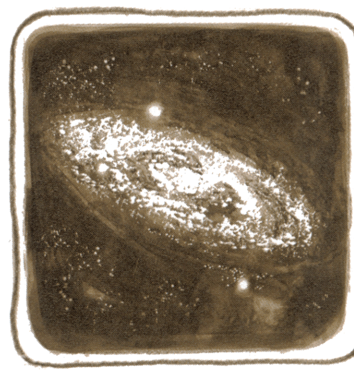
Primary Storage Registers

Storing Numbers

To store a number that is in the X-register into any primary storage register (00 through 99):

1. Press [STO]. The HP-41C will prompt you for the address number with **STO _ _**.
2. Press the number keys of the applicable register address (00 through 99). Address numbers must be 2 digits, e.g., 01, 02, or 50. The operation is performed when you enter the second digit.

For example, to store 2,200,000 (the distance in light-years of the Great Spiral Galaxy in Andromeda from Earth) in register R_{12}:



Keystrokes	**Display**	
2200000	*2,200,000 _*	The number.
[STO]	*STO _ _*	The HP-41C prompts: *In which register?*
12	*2,200,000.000*	The number is stored in R_{12}.
[■] [CLX]	*0.0000*	Clears the displayed X-register.

Notice that when a number is stored, it is merely copied into the storage register, so 2,200,000.000 also remains in the X-register. Storing a number does not change the contents of the automatic memory stack.

Recalling Numbers

Numbers are recalled from storage registers back into the displayed X-register in much the same way they are stored. To recall a number from a primary storage register (00 through 99):

1. Press [RCL]. The HP-41C prompts you for the register address with **RCL _ _**.

2. Press the number keys of the applicable register address (00 through 99). Addresses must be 2 digits, e.g., 01, 02, or 50.

For example, to recall the distance to Andromeda's Great Spiral stored in register R_{12}:

Keystrokes	Display	
RCL	RCL _ _	The HP-41C prompts: *Recall from which register?*
12	2,200,000.000	The function is performed when the second digit is entered. The distance to Great Spiral is now in the displayed X-register.

Recalling a number into the X-register causes the stack contents to lift. That is, the previous X value is lifted into the Y-register, the previous Y into the Z-register, the previous Z into the T-register. The previous value in the T-register is lost off the top of the stack.

Storing and Recalling in Stack Registers

Using the HP-41C, you can even store and recall numbers into and from the stack and LAST X registers. All you have to do is press [•] (decimal point) and X, Y, Z, T, or L (for LAST X) as the register address. When the HP-41C prompts you for the address, simply press the associated letter key (X, Y, Z, T, or L)—there is no need to enter ALPHA mode. For example, to store the number 19 into the Z-register of the stack:

Keystrokes	Display	
19	19_	The number is in the X-register.
STO	STO _ _	The HP-41C prompts: *In which register?*
[•]	STO ST_	Now the HP-41C prompts: *In which stack register?*
Z	19.0000	The number is stored in the Z-register.
■ CLX	0.0000	

Now, recall the value from the Z-register:

Keystrokes	Display	
RCL	RCL _ _	The HP-41C prompts: *From which register?*
[•]	RCL ST_	Now the prompt is: *From which stack register?*
Z	19.0000	The number is recalled from Z.

Storing ALPHA Strings

ALPHA strings that you have placed in the ALPHA-register can be stored into any storage registers, even into the stack registers. (An ALPHA string is a series of ALPHA characters.) In ALPHA mode, the shifted functions on the [STO] and [RCL] keys are the [ASTO] *(ALPHA store)* and [ARCL] *(ALPHA recall)* functions. All you do is press ■ [ASTO] or ■ [ARCL] and specify the register address. The HP-41C prompts you with **ASTO __** and **ARCL __**.

[ASTO] stores the left-most six characters in the ALPHA register into the specified register. An additional function, [ASHF] (ALPHA shift), helps you store strings longer than six characters by shifting the contents of the ALPHA register *left* six characters. When you execute [ASHF], the first six characters in the ALPHA register are *lost*. [ASHF] is most useful in programs and is covered in part II of this handbook.

The [ASTO], [ASHF], and [ARCL] functions operate on the ALPHA register only. The stack is not disturbed by these operations unless you specify a stack register address (more about this in a moment).

To store an ALPHA string that is in the ALPHA register into any primary storage register:

1. In ALPHA mode, press ■ [ASTO] (press ■ [STO] in ALPHA mode). The HP-41C will prompt you for the address with **ASTO __**
2. Press the number keys of the address of the desired register (00 through 99). Since the HP-41C prompts you for the register address, you need not go out of ALPHA mode to key in the numbers.

For example, to store the ALPHA string MICRO into register R_{05}:

Keystrokes	Display	
[ALPHA]		
MICRO	*MICRO_*	
■ [ASTO]	*ASTO __*	The HP-41C prompts: *In which register?*
05	*MICRO*	MICRO is stored into R_{05}.
■ [CLA]		Blanks the ALPHA register.

The string MICRO is now stored into R_{05}. Remember, each storage register can hold a maximum of six ALPHA characters.

Recalling ALPHA Strings

Now, to recall an ALPHA string that is stored in any storage register: (Remember, [ARCL] does not disturb the stack, it only brings strings into the ALPHA register.)

1. In ALPHA mode, press [ARCL] (press ■ [RCL] in ALPHA mode). The HP-41C will prompt you for the address with **ARCL __**.
2. Key in the desired register address (00 through 99).

For example, to recall the string stored in register R_{05} (The HP-41C should still be in ALPHA mode.):

Keystrokes	Display	
ARCL	ARCL _ _	The HP-41C prompts: From which register?
05	MICRO _	The string is recalled from R_{05}.

ARCL always adds the recalled strings to whatever is already in the ALPHA register. For example, recall the string from R_{05} again.

Keystrokes	Display	
ARCL	ARCL _ _	
05	MICROMICRO _	The string is recalled again from R_{05} and is added to the string already in the ALPHA register.

It is a good idea to remove unwanted ALPHA characters from the ALPHA register *before* you use ARCL. Simply press CLA in ALPHA mode to accomplish this.

Keystrokes	Display	
	MICROMICRO _	
CLA		The ALPHA register has been cleared.
ALPHA	19.0000	

ALPHAs and the Stack

Stack registers and LAST X can be specified as ASTO and ARCL register addresses. Any time you wish to specify a stack register or LAST X as an address, simply press . (decimal point) and the desired register letter key (X, Y, Z, T, or L) in response to the function prompt. For example:

Keystrokes	Display	
ALPHA		
ENERGY	ENERGY _	The string.
ASTO	ASTO _ _	The prompt: *In which register?*
. T	ASTO T	Stores ENERGY in stack register T.
	ENERGY	
CLA		Clears the displayed ALPHA register.

Now, recall the string:

Keystrokes:	**Display:**	
■ ARCL	*ARCL _ _*	
. T	*ARCL T*	Contents of stack register T are recalled into the displayed ALPHA register. Hold the T key down for a moment to see the ***ARCL T*** prompt.
	*ENERGY_*	
■ CLA		
ALPHA ■ CLX	*0.0000*	Returns to normal mode and clears the displayed X-register.

VIEW Function

When in normal, USER, or ALPHA modes, you can view the contents of any HP-41C register without disturbing the stack. You simply press ■ VIEW and specify a register address. For example, to view the contents of R_{12} without disturbing the stack:

Keystrokes	**Display**	
■ VIEW	*VIEW _ _*	The HP-41C prompts: *View which register?*
12	*2,200,000.000*	The stack has not been disturbed.

The stack and LAST X registers can also be viewed in the same manner. Simply press . and X, Y, Z, T, or L (for LAST X) in response to the prompt.

In ALPHA mode, when you use VIEW, the AVIEW *(ALPHA view)* function is executed. AVIEW simply places the contents of the ALPHA register into the display.

If you use ARCL to recall a number (not ALPHA characters and not ALPHA numbers) from a register, that number will simply appear as the corresponding ALPHA characters. Numbers with exponents will appear with the exponent prefixed with the letter E. For example:

Keystrokes	**Display**	
23 STO 00	*23.0000*	
ALPHA		
■ ARCL 00	*23.0000_*	The number now appears as ALPHA characters and is not valid for arithmetic functions.
■ CLA		
ALPHA	*23.0000*	The original *number* is in X.
68 EEX 93	*68 93*	
STO 01	*6.8000 94*	

Keystrokes:	Display:	
ALPHA		
■ ARCL 01	6.8000E94_	The number now appears as ALPHA characters and is not valid for arithmetic functions. The exponent is marked with E.
■ CLA		
ALPHA	6.8000 94	The original *number* is in X.

Defining Storage Register Configurations

As you read at the beginning of this section, you can control the amount of HP-41C memory that is allocated to both data storage registers and program memory. The SIZE function allows you to specify the number of data storage registers you wish to have allocated. Remember, your basic HP-41C has up to 63 data storage registers and you can add memory modules for a total of up to 319.

When you execute SIZE, the HP-41C prompts you for a three-digit number from 000 through 319.

If you attempt to increase the allocation of storage registers and there is not enough unused space in program memory for this increase, the HP-41C will display ***PACKING*** and then ***TRY AGAIN***. After you execute SIZE again, if the HP-41C again displays ***PACKING*** and ***TRY AGAIN***, this means that the reallocation is not possible until program instructions are deleted from program memory.

If you decrease the allocation of data storage registers, any information in reallocated data storage registers will be lost.

Any attempt to store or recall using a storage register that is not in the current allocation will result in the ***NONEXISTENT*** message in the display. For example, if the HP-41C has 17 storage registers allocated (R_{00} through R_{16}), STO 55 will result in the ***NONEXISTENT*** display.

Clearing Storage Registers

Even though you have recalled or viewed the contents of a storage register, the number or string also remains in the storage register. You can clear storage registers in three ways:

1. To clear the contents of a single storage register, merely store another number there. The original number is cleared by the new number.
2. To clear a storage register, replace the number in it with zero. For example, to clear register R_{12}, press 0 STO 12.
3. To clear all storage registers at once, execute the CLRG *(clear all registers)* function. CLRG clears all currently allocated data storage registers to zeros. CLRG does not alter program memory or the automatic memory stack. CLRG must be assigned to a key for execution or executed from the display.

Remember that because of the Continuous Memory of the HP-41C, all information in the calculator is retained, even when the calculator is turned off.

Use [CLRG] now to clear all currently allocated storage registers (R_{00} through R_{16}).

Keystrokes	Display	
[XEQ]	*XEQ _ _*	
[ALPHA] CLRG [ALPHA]	*6.8000 94*	All currently allocated storage registers have been cleared.

To clear the entire calculator (all programs, registers, assignments, flags, etc.) with the "master clear:" turn the HP-41C off, hold down the [←] key, and turn the calculator back on again. The display will show ***MEMORY LOST***.

Storage Register Arithmetic

Arithmetic can be performed upon the contents of all storage registers by executing [STO] followed by the arithmetic function followed in turn by the register address. For example:

Operation	Result
[STO] [+] 01	Number in X-register is added to the contents of register R_{01}, and the sum is placed into R_{01}. The display execution form of this is [ST+].
[STO] [−] 02	Number in X-register is subtracted from the contents of R_{02}, and the difference is placed into R_{02}. The display execution form is [ST−].
[STO] [×] 03	Number in X-register is multiplied by the contents of R_{03}, and the product is placed into R_{03}. The display execution form of this is [ST×].
[STO] [÷] 04	Number in R_{04} is divided by the number in the X-register, and the quotient is placed into R_{04}. The display execution form of this is [ST÷].

When storage register arithmetic operations are performed, the HP-41C prompts for the register address, and the answer is written into the selected storage register. Unless specified as a register address, the stack remains unchanged.

Storage Register Arithmetic and the Stack

You can also specify the stack or LAST X registers for storage register arithmetic by simply pressing [•] (decimal point) and X, Y, Z, T, or L (for LAST X) as the register address. For example, place the number 50 in the X-register and add the number to itself:

Keystrokes	Display	
50	*50 _*	The X-value.
[STO] [+]	*ST + _ _*	The HP-41C prompts: *In which register?*

Keystrokes:	Display:	
[•]	ST + ST_	[•] specifies the stack. The HP-41C now prompts: *In which stack register?*
X	100.0000	The value in X, 50, is added to itself.

Storage Register Overflow

If you attempt a storage register operation that would cause the magnitude of a number in any of the storage registers to exceed $9.999999999 \times 10^{99}$, the operation is not performed and the HP-41C display immediately indicates **OUT OF RANGE**. When you press [←], the error condition is cleared and the last value in the X-register before the error is displayed. The storage registers and the stack all contain the values they held before the error-causing operation was attempted.

For example, if you store 7.33×10^{52} in R_{01} and attempt to use storage register arithmetic to multiply that value by 10^{50}, the display will show **OUT OF RANGE**

Keystrokes	Display
7.33	7.33_
[EEX] 52	7.33 52
[STO] 01	7.3300 52
[EEX] 50	1 50
[STO] [×] 01	OUT OF RANGE

To clear the overflow and return the HP-41C to the status prior to the error-causing condition, press [←].

Keystrokes	Display	
[←]	1.0000 50	Contents of X-register.
[RCL] 01	7.3300 52	Contents of R_{01}.

Later, in section 14 of this handbook, you will learn how to tell the HP-41C to ignore these kinds of range errors.

Section 6

Functions

The Standard Function Catalog

The HP-41C has over 130 internal functions that allow you to compute answers to problems quickly and accurately. You can list this set of functions at any time by pressing [shift] [CATALOG] 3.

This section gives a brief explanation of most HP-41C standard functions (except programming functions, which are presented in part II) along with some example problems. All of the functions in this section can be stored and executed in program memory as part of a user program unless otherwise noted. Remember, execution of all functions not on the keyboard is simple when you assign the functions to the USER mode keyboard for execution (refer to section 4).

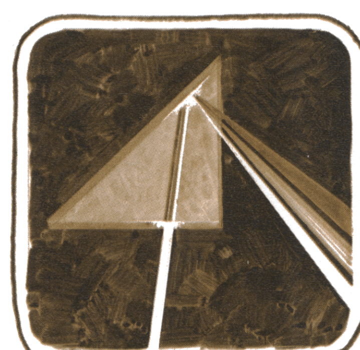
General Mathematics Functions

Changing the Sign of a Number

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. You can also change the sign of either a negative or positive nonzero number in the display by pressing [CHS]. For example, key in 2.54 and change the sign of the number.

Keystrokes	Display	
2.54	2.54_	The number.
[CHS]	-2.54_	The sign is changed.
[CHS]	2.54_	Sign is changed back again.

To change the sign of an exponent of a number, you must use [CHS] immediately after keying in the exponent (before you perform some operation that terminates digit entry). As soon as digit entry is terminated, [CHS] changes the sign of the mantissa of the number, not the exponent. For example, key in the Rydberg constant (1.0973731×10^7, a universal constant used in spectroscopy) and change the sign of the exponent.



Keystrokes	Display	
■ [CLX]	0.0000	
1.0973731	1.0973731 _	
[EEX] 7	1.0973731 7_	Rydberg constant.
[CHS]	1.0973731 -7_	Sign of the exponent is changed.
[CHS]	1.0973731 7_	Sign of the exponent is changed back again.

Rounding a Number

As you know, when you change display formats with one of the display control functions ([FIX], [SCI], or [ENG]), the number maintains its full value (10 digits multiplied by a two-digit exponent of 10) no matter how many digits you see. When you execute the [RND] *(round)* function, however, the number that is in the display becomes the *actual* number in the calculator. For example, round the Rydberg constant, now in the display, to two digits beyond the decimal point in [SCI] format.

Keystrokes	Display	
	1.0973731 7_	Rydberg constant still in X-register.
■ [SCI] 2	1.10 07	The display format. Remember, the number in the HP-41C is still in its full-accuracy form internally.
[XEQ]	XEQ __	The HP-41C prompts: *Execute what?*
[ALPHA] RND [ALPHA]	1.10 07	The [RND] function is executed.
■ [SCI] 6	1.100000 07	The [SCI] 6 display shows that the number has been rounded internally.
■ [FIX] 4	11,000,000.00	Display mode set to [FIX] 4.

Absolute Value

Some calculations require the absolute value, or magnitude, of a number. To obtain the absolute value of the number in the X-register, execute the [ABS] function. For example, to calculate the absolute value of -3:

Keystrokes	Display	
3 [CHS]	-3_	
[XEQ]	XEQ __	
[ALPHA] ABS [ALPHA]	3.0000	$\lvert -3 \rvert$.

To compute the absolute value of +3:

Keystrokes	Display	
XEQ	XEQ _ _	
ALPHA ABS ALPHA	3.0000	|+3|.

Integer Portion of a Number

To extract and display the integer portion of a number, execute INT *(integer)*. For example, to extract only the integer portion of the number 123.4567:

Keystrokes	Display	
123.4567	123.4567 _	
XEQ	XEQ _ _	
ALPHA INT ALPHA	123.0000	Only the integer portion of the number remains.

When INT is executed, 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, execute the FRC *(fraction)* function. For example, to extract only the fractional portion of the number 123.4567 used above:

Keystrokes	Display	
■ LASTX	123.4567	Summons the number from LAST X.
XEQ	XEQ _ _	
ALPHA FRC ALPHA	0.4567	Only the fractional portion of the number is displayed.

When FRC is executed, the integer portion of the number is lost. The entire number is again preserved in the LAST X register.

The Modulo Function (Remainder)

Executing MOD *(modulo)* performs y mod x (the equation is $y-[<y/x>\times x]$, where $<>$ denotes the largest integer less than or equal to the indicated result), which divides y by x and gives the remainder of the division. So, when you place numbers in the X- and

Y-registers, the y value is divided by x and the remainder is placed back into the X-register. For example, find 128 modulo 10:

Keystrokes	Display	
128 ENTER↑	128.0000	The y-value.
10	10_	The x-value.
XEQ		
ALPHA MOD ALPHA	8.0000	The value is in X.

Performing y mod x when x=0 returns an answer of y.

Reciprocals

To calculate the reciprocal of a number in the X-register, key in the number, then execute the 1/x *(reciprocal)* function. For example, to calculate the reciprocal of the number 1.7588028×10^{11} (the electron charge-to-mass ratio):

Keystrokes	Display	
1.7588028 EEX 11	1.7588028 11	The number.
1/x	5.6857 -12	The reciprocal.

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

Example: In an electrical circuit, three resistors are connected in parallel and a single resistor is connected in series with the parallel circuit. The resistors in parallel have values of 2.7 kilohms, 5.6 kilohms, and 7.5 kilohms, and the series resistor has a value of 680 ohms. What is the total resistance of the circuit?

$$R_t = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}} + R_4 = \frac{1}{\frac{1}{2700} + \frac{1}{5600} + \frac{1}{7500}} + 680$$

Keystrokes	Display	
2700 1/x	0.0004	
5600 1/x	0.0002	
+	0.0005	
7500 1/x	0.0001	
+	0.0007	Sum of the reciprocals.
1/x	1,465.6844	The reciprocal of the sum of the reciprocals.
680 +	2,145.6844	Addition of the series value yields the answer in ohms.

Factorials

The FACT function permits you to handle permutations and combinations with ease. To calculate the factorial of a positive integer in the X-register, execute the FACT function. For example, calculate the number of ways that six people can line up for a photograph.

$$P_6^6 = 6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

Keystrokes	Display	
6	6_	
XEQ	XEQ __	
ALPHA FACT ALPHA	720.0000	The answer.

The HP-41C displays ***OUT OF RANGE*** for factorials of numbers greater than 69.

Square Roots

To calculate the square root of a number in the X-register, execute the $\sqrt{x}$ function. On the keyboard the function label looks like this: $\sqrt{x}$. And when you execute the function from the display or reassign the function, the name is SQRT .

Keyboard execution: $\sqrt{x}$

Display execution: SQRT

Find the square root of 16 using the $\sqrt{x}$ key on the keyboard:

Keystrokes	Display
16	16_
$\sqrt{x}$	4.0000

Now find the square root of the result using SQRT in the display:

Keystrokes	Display
	4.0000
XEQ	XEQ __
ALPHA SQRT ALPHA	2.0000

Squaring

To square a number in the X-register, execute the [x^2] and when you execute the function from the display the name is [X↑2] (using the up arrow, the shifted function on the [ENTER↑] key, in ALPHA mode).

Keyboard execution: [x^2]

Display execution: [X↑2]

For example, find the square of 27 using the keyboard [x^2] function:

Keystrokes	Display
27 ■ [x^2]	729.0000

Now, find the square of that number using the display execution form:

Keystrokes	Display	
	729.0000	
[XEQ]	XEQ _ _	
[ALPHA]		
X ■ ↑ ■ 2		The up-arrow is on the shifted N key in ALPHA mode (on [ENTER↑] key).
[ALPHA]	531,441.0000	

Using Pi

The value of pi accurate to 10 places (3.141592654) is provided as a fixed constant in your HP-41C. Merely press ■ [π] on the keyboard or execute [PI] from the display whenever you need it in a calculation.

Keyboard execution: [π]

Display execution: [PI]

For example, calculate the surface area of Ganymede, one of Jupiter's 12 moons, using the formula $A = \pi d^2$. Ganymede has a diameter (d) of 3200 miles.

Keystrokes	Display	
3200	3200 _	
■ [x^2]	10,240,000.00	
■ [π]	3.1416	The quantity pi.
[×]	32,169,908.78	Area of Ganymede in square miles.

Now, using the display execution form, PI, find the area of Europa, a moon of Jupiter with a diameter of 1950 miles:

Keystrokes	Display	
1950	*1950_*	
■ x^2	*3,802,500.000*	
XEQ	*XEQ __*	
ALPHA PI ALPHA	*3.1416*	The quantity pi.
×	*11,945,906.07*	Area of Europa in square miles.

Percentages

The % *(percent)* function 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 the percent rate.
4. Press ■ %.

Example: About 94% of the weight of a tomato is water. If a particular tomato weighs 500 grams, what amount of the weight of the tomato is water?

Keystrokes	Display	
500	*500_*	The base number.
ENTER↑	*500.0000*	
94	*94_*	The percent of water.
■ %	*470.0000*	The weight in grams of water in a 500-gram tomato.

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

... from this ...				... to this.		
T	*0.0000*			T	*0.0000*	
Z	*0.0000*			Z	*0.0000*	
Y	*500.0000*	Base. →	%	→ Y	*500.0000*	Base.
X	*94.0000*	Rate. →		→ X	*470.0000*	Answer.

Notice that the calculated answer writes over the percent rate in the X-register, and the base number is preserved in the Y-register.

Since the total tomato weight is still in the Y-register and the weight of the water in the tomato is in the X-register, the weight of the remainder can be obtained by simply subtracting:

Keystrokes	Display	
	470.0000	The weight of the water.
[−]	30.0000	The gram weight of solid matter in the 500-gram tomato.

Percent of Change

The [%CH] *(percent of change)* function is a two-number function that calculates the percent increase or decrease from a number in the Y-register to the number in the X-register. To find the percent of change:

1. Key in the base number (usually, the number that happens first in time).
2. Press [ENTER↑].
3. Key in the second number.
4. Execute the [%CH] function from the display. Percent of change is calculated as $\%CH = \left[(x - y)\ 100\right] \div y$. (When $y = 0$ a value of $9.999999999 \times 10^{99}$ is placed in the X-register and the calculator displays **OUT OF RANGE**.)

Example: Tomato grower Flem Snopes has found that he can decrease the amount of water in the tomatoes he is growing. His typical tomato weighs about 500 grams. He has found that only 430 grams of the total weight is water, compared to 470 grams of water in the tomato from the previous example. What is the percent of change of water amount between the average tomato and Snopes' tomato?

Keystrokes	Display	
470	470_	The weight of the water of the first (500-gram) tomato.
ENTER↑	470.0000	
430	430_	The weight of the water in Snopes' tomato.
XEQ	XEQ __	
ALPHA		
■ %CH		
ALPHA	-8.5106	Percent decrease in weight of water in Snopes' tomato.

Unary of X

SIGN is a function that returns 0, −1, or 1 to the X-register depending on the value presently in X.

If the value in X is ALPHA characters, SIGN returns 0 to X.

If the value in X is less than zero (negative), SIGN returns −1 to X.

If the value in X is zero, SIGN returns 1 to X.

If the value in X is greater than zero (positive), SIGN returns 1 to X.

The original value of X is preserved in LAST X.

Trigonometric Functions

Trigonometric Modes

When you are using trigonometric functions, angles can be assumed by the HP-41C to be in decimal degrees, radians, or grads. Unless you specify otherwise with one of the trigonometric mode functions, the HP-41C assumes that angles are in decimal degrees. When you specify a trigonometric mode, the HP-41C remains in that mode until you change it, even while the HP-41C is turned off.

To select radians mode, execute the RAD *(radians)* function before using a trigonometric function. The RAD annunciator in the display will turn on to remind you that you are in radians mode.

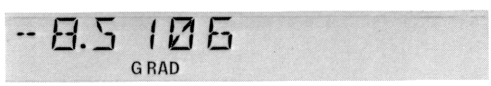
To select grads mode, execute the GRAD *(grads)* function before using a trigonometric function. The GRAD annunciator in the display will turn on to remind you that you are in grads mode.

To select decimal degrees mode, execute the DEG *(degrees)* function before using a trigonometric function. Since the HP-41C normally assumes that angles are in decimal degrees, no display annunciator shows.

To see the RAD and GRAD annunciators in the display...

Keystrokes	Display	
XEQ	XEQ __	
ALPHA RAD ALPHA	-8.5106	Notice that the RAD annunciator turns on. (The number in the display remains from the previous example.)

Keystrokes	Display	
XEQ	XEQ __	
ALPHA GRAD ALPHA	-8.5106	Notice that the GRAD annunciator turns on.
■ CLX	0.0000	



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

Trigonometric Functions

There are 6 trigonometric functions provided by the HP-41C. Both the keyboard form and the display execution form of the function are given: keyboard form first.

- SIN (sine)*
- SIN^{-1} or ASIN (arc sine)
- COS (cosine)
- COS^{-1} or ACOS (arc cosine)
- TAN (tangent)
- TAN^{-1} or ATAN (arc tangent)

Each of these trigonometric functions assume that angles are entered 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 execute the function. For example, find the cosine of 35 degrees.

* In the HP-41C π is truncated to 10 digits. So, the sine of π radians is -4.1×10^{-10}. This is correct for π of 10 digits accuracy.

Keystrokes	Display	
XEQ	XEQ __	
ALPHA DEG ALPHA	0.0000	Sets the HP-41C to degrees mode.
35	35_	
COS	0.8192	

Now, find the arc sine in radians of .964.

Keystrokes	Display	
XEQ	XEQ __	
ALPHA RAD ALPHA	0.8192	Number remains from previous example. The HP-41C is now in radians mode, the RAD display annunciator is on.
.964	.964_	
■ SIN⁻¹	1.3017	Radians.

Next, find the tangent of 43.66 grads.

Keystrokes	Display	
XEQ	XEQ __	
ALPHA GRAD ALPHA	1.3017	The HP-41C is now in grads mode and the GRAD display annunciator is on. Number remains from previous example.
43.66	43.66_	
TAN	0.8183	
XEQ	XEQ __	
ALPHA DEG ALPHA	0.8183	Sets the HP-41C back to degrees mode.

Degrees/Radians Conversions

The D-R *(degrees to radians)* and R-D *(radians to degrees)* functions are used to convert angles between degrees and radians. To convert an angle specified in degrees to radians, key in the angle and execute D-R . If you expect to be using these functions regularly, it is a good idea to assign them to the keyboad for execution in USER mode. For example, to change 45 degrees to radians:

Keystrokes	Display	
45	45_	
XEQ	XEQ __	
ALPHA		
D ■ -R		
ALPHA	0.7854	Radian.

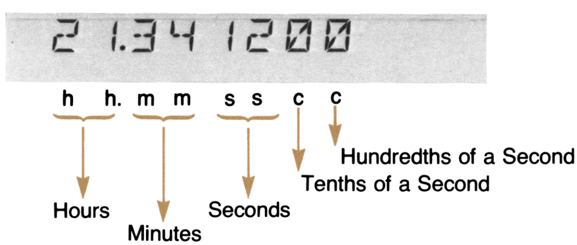
To convert the angle specified in radians to decimal degrees, key in the angle and execute the [R-D] function from the display. For example, to convert 4 radians to decimal degrees:

Keystrokes	Display	
4	4_	
[XEQ]	XEQ _ _	
[ALPHA]		
R -D		
[ALPHA]	229.1831	Decimal degrees.
[CLX]	0.0000	

Hours, Minutes, Seconds/Decimal Hours Conversions

Using the HP-41C, you can change time specified in decimal *hours* to *hours, minutes, seconds* format by executing the [HMS] (decimal *hours* to *hours, minutes, seconds)* function. You can also change from *hours, minutes, seconds* to decimal *hours* by executing the [HR] *(hours, minutes, seconds* to decimal *hours)* function. Both of these functions are executed using [XEQ] or assigned to a key for execution in USER mode.

When a time is displayed in *hours, minutes, seconds* format, the digits specifying *hours* occur to the left of the decimal point, while the digits specifying the *minutes, seconds, and fractions of seconds* occur to the right of the decimal point.



Before you begin the examples, assign [HMS] to the [LN] key location and assign [HR] to the [e^x] key location. Then place the HP-41C into USER mode.

Keystrokes	Display
[ASN]	ASN _
[ALPHA] HMS [ALPHA]	ASN HMS _
[LN]	ASN HMS 15
	0.0000
[ASN]	
[ALPHA] HR [ALPHA]	ASN HR _
[e^x]	ASN HR -15
	0.0000
[USER]	0.0000

To convert from decimal *hours* to *hours, minutes, seconds,* simply key in the value for decimal *hours* and execute HMS. For example, to change 21.57 hours to *hours, minutes, seconds:*

Keystrokes	Display	
21.57	21.57_	
HMS (LN)	21.3412	This is 21 hours, 34 minutes, 12 seconds.

Notice that the display is *not* automatically switched to show you more than four digits after the decimal point. Unless you change it, the display format remains the same as prior to the problem.

To convert from *hours, minutes, seconds* to decimal *hours,* simply key in the value for *hours, minutes, seconds* in that format, and execute the HR function. For example, to convert 167 hours, 22 minutes, and 15.68 seconds to its decimal equivalent:

Keystrokes	Display	
167.221568	167.221568_	This is 167 hours, 22 minutes, 15.68 seconds.
■ HR (e^x)	167.3710	This is 167.3710 hours.

Using the HMS and HR functions, 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 19.34 decimal degrees to *degrees, minutes, seconds.*

Keystrokes	Display	
19.34	19.34_	The angle.
HMS (LN)	19.2024	This means 19 degrees, 20 minutes, 24 seconds.

Example: Convert 9 degrees, 9 minutes, 59.3 seconds to its decimal equivalent.

Keystrokes	Display	
9.09593	9.09593_	The angle.
■ HR (e^x)	9.1665	Answer in decimal degrees.
USER	9.1665	
■ CLX	0.0000	

Adding and Subtracting Time and Angles

To add or subtract decimal hours, merely key in the numbers for the decimal hours and press [+] or [−]. To add *hours, minutes, seconds,* use the [HMS+] *(add hours, minutes, seconds)* or [HMS−] *(subtract hours, minutes, seconds)* function. Both of these functions are executed using [XEQ] or by assigning them to a key for execution in USER mode.

Likewise, angles specified in *degrees, minutes, seconds* are added and subtracted using the [HMS+] and [HMS−] functions.

Assign [HMS+] and [HMS−] to the [LOG] and [10^x] keys, respectively, for execution in USER mode.

Keystrokes	Display
■ [ASN]	ASN _
[ALPHA]	
HMS ■ +	
[ALPHA]	ASN HMS + _
[LOG]	ASN HMS + 14
	0.0000
■ [ASN]	ASN _
[ALPHA]	
HMS ■ −	
[ALPHA]	ASN HMS − _
■ [10^x]	ASN HMS − −14
	0.0000
[USER]	0.0000

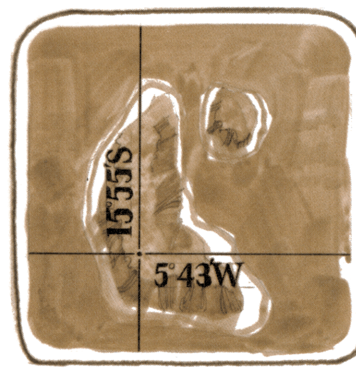
Example: Find the sum of 45 hours, 10 minutes, 50.76 seconds and 24 hours, 49 minutes, 10.95 seconds, then subtract 7 hours, 23 minutes, 11 seconds from that result.

Keystrokes	Display
45.105076	45.105076 _
[ENTER↑]	45.1051
24.491095	24.491095 _
[HMS+] ([LOG])	70.0002
7.2311	7.2311 _
■ [HMS−] ([10^x])	62.3651

Keystrokes:	Display:	
■ FIX 6	62.365071	FIX 6 display lets you see the whole number.
■ FIX 4	62.3651	Display set to FIX 4.
USER	62.3651	
■ CLX	0.0000	

In the HP-41C, trigonometric functions assume angles in decimal degrees, decimal radians, or decimal grads. If you want to use any trigonometric functions on angles given in *degrees, minutes, 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 inhabitated 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 (St. Helena).

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 inhabitated land is:

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

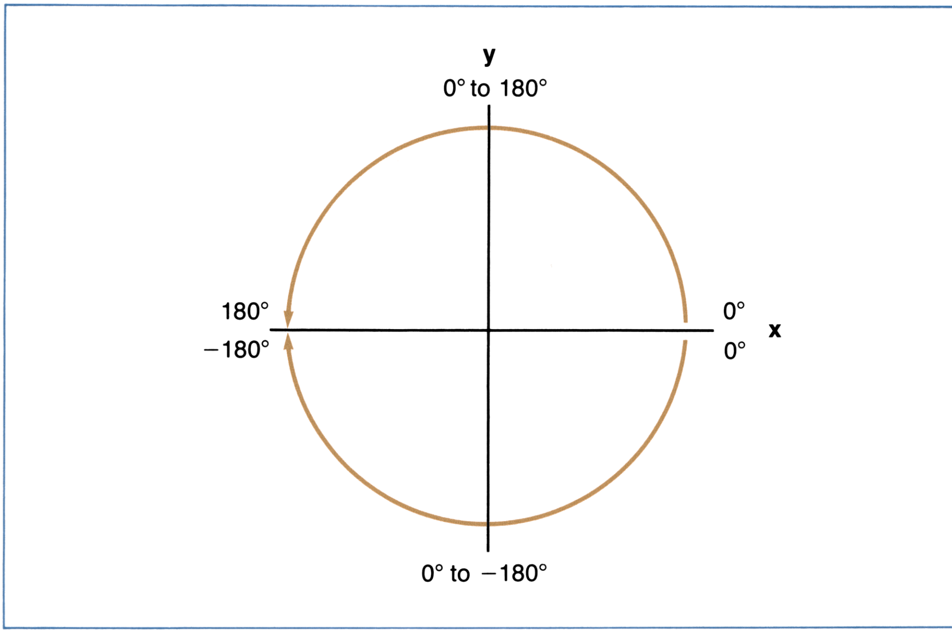
Since the HR function is still assigned to the e^x key location, simply switch to USER mode.

Keystrokes	Display	
USER	0.0000	
5.43	5.43 _	The HP-41C is still in DEG mode.
■ HR (e^x)	5.7167	
12.18	12.18 _	
■ HR (e^x)	12.3000	
−	-6.5833	
COS	0.9934	
15.55	15.55 _	
■ HR (e^x)	15.9167	
STO 01	15.9167	
COS	0.9617	
×	0.9553	
37.03	37.03 _	
■ HR (e^x)	37.0500	
STO 00	37.0500	
COS	0.7981	
×	0.7625	
RCL 00 SIN	0.6025	
RCL 01 SIN	0.2742	
×	0.1652	
+	0.9277	
■ COS^{-1}	21.9235	
60 ×	1,315.4110	Distance in nautical miles that Odysseus must sail to visit Penelope.
USER	1,315.4110	
■ CLX	0.0000	

Polar/Rectangular Coordinate Conversions

Two functions are provided in the HP-41C for polar/rectangular coordinate conversions. Angle θ is assumed to be in decimal degrees, radians, or grads, depending upon the trigonometric mode first selected by the DEG, RAD, or GRAD functions.

In the HP-41C, 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↑**.
3. Key in the x-coordinate.
4. Execute **R-P** *(rectangular to polar)*. Magnitude r is placed in the X-register and angle θ is placed in the Y-register. To display the θ value, press **x≷y**.

When you execute the **R-P** function, the stack contents are changed...

	... from this ...			... to this.
T	*t*		T	*t*
Z	*z*		Z	*z*
Y	*y-coordinate* →	**R-P**	→ Y	*angle* θ
X	*x-coordinate* →		→ X	*magnitude r*

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

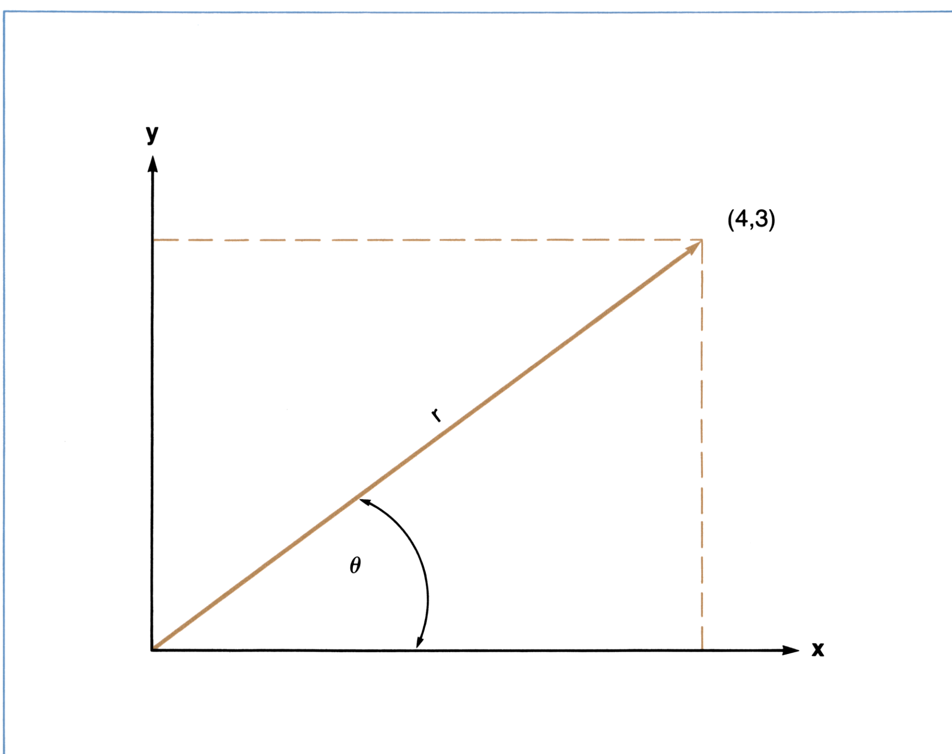
1. Key in the value for the angle.
2. Press [ENTER↑].
3. Key in the value for the magnitude r.
4. Execute [P→R] *(polar to rectangular)*. The x-coordinate is placed in the X-register and the y-coordinate is placed in the Y-register. To display the y-coordinate value, press [x⇄y].

When you execute the [P→R] function, the stack contents are changed...

... from this ...			... to this.	
T	*t*		T	*t*
Z	*z*		Z	*z*
Y	*angle* θ	→ [P→R] →	Y	*y-coordinate*
X	*magnitude r*	→ [P→R] →	X	*x-coordinate*

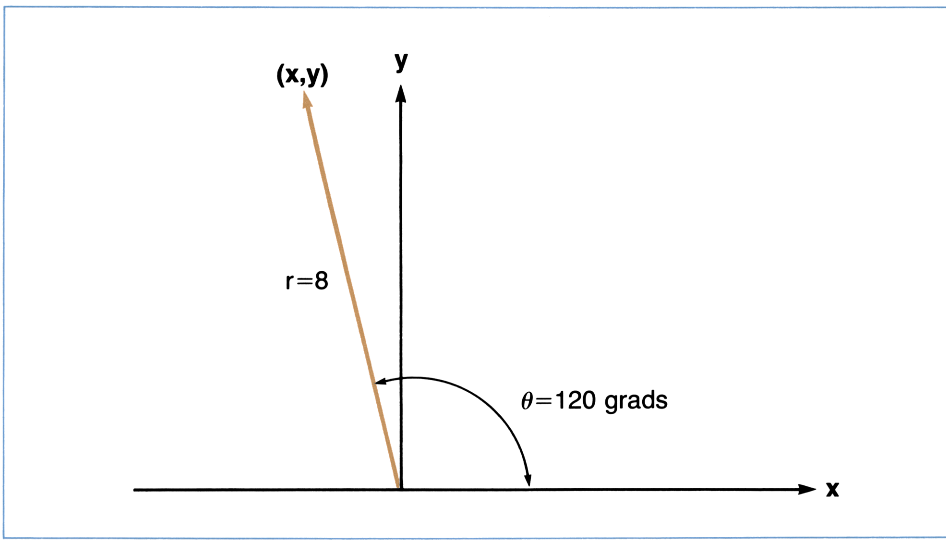
After you execute [R→P] or [P→R], you can press [x⇄y] to place the calculated angle θ or the calculated y-coordinate into the X-register for viewing or further calculation.

For example, convert rectangular coordinates (4,3) to polar form with the angle expressed in radians.



Keystrokes	Display	
XEQ	XEQ _ _	
ALPHA RAD ALPHA	0.0000	Radians mode selected.
3 ENTER↑	3.0000	The y-coordinate is entered into the Y-register.
4	4 _	x-coordinate keyed in.
■ R-P	5.0000	Magnitude r.
x⇄y	0.6435	Angle θ in radians.

Now convert polar coordinates (8, 120 grads) to rectangular coordinates.



Keystrokes	Display	
XEQ	XEQ _ _	
ALPHA GRAD ALPHA	0.6435	Grads mode selected. Displayed result remains from previous example.
120 ENTER↑	120.0000	Angle θ is placed into the Y-register.
8	8 _	Magnitude r is keyed in.
■ P-R	-2.4721	The x-coordinate.
x⇄y	7.6085	The y-coordinate.
XEQ		
ALPHA DEG ALPHA	7.6085	Returns the HP-41C to DEGrees mode.

Logarithmic and Exponential Functions

The HP-41C computes both natural and common logarithms as well as their inverse functions (antilogarithms). The logarithmic functions are: (Notice that the keyboard execution and display execution forms of the natural and common antilog functions are different.)

Natural log	Keyboard and display: $\boxed{\text{LN}}$	Takes the log of the value in the X-register to base e (2.718...).
Natural antilog	Keyboard execution: $\boxed{e^x}$ Display execution: $\boxed{\text{E↑X}}$	Raises e (2.718...) to the power of the value in the X-register. Press 1 $\boxed{e^x}$ to display value of e.
Common log	Keyboard and display: $\boxed{\text{LOG}}$	Computes the log of the value in the X-register to base 10.
Common antilog	Keyboard execution: $\boxed{10^x}$ Display execution: $\boxed{\text{10↑X}}$	Raises 10 to the power of the value in the X-register.
Natural log (for arguments close to one)	Display execution: $\boxed{\text{LN1+X}}$	Computes ln (1 + X), where X is a number very close to zero. $\boxed{\text{LN1+X}}$ provides greater precision than $\boxed{\text{LN}}$ when you are finding that the natural log of numbers close to one. Example: To find the natural log of $(1 + 4.25 \times 10^{-6})$, key in 4.25×10^{-6} and execute $\boxed{\text{LN1+X}}$. Answers are displayed in $\boxed{\text{SCI}}$ format.
Natural antilog (for arguments close to zero)	Display execution: $\boxed{\text{E↑X-1}}$	Computes $(e^x) - 1$, where X is a number very close to zero. $\boxed{\text{E↑X-1}}$ provides greater precision than $\boxed{e^x}$ for numbers very close to zero. Example: To calculate $(e^{4.25 \times 10^{-6}}) - 1$, key in 4.25×10^{-6} and execute $\boxed{\text{E↑X-1}}$. Answers are displayed in $\boxed{\text{SCI}}$ format.

Let's work an example using LOG. The village of Musser has installed a 12-o'clock whistle in the firehouse steeple near the center of the village. If the sound level at the steeple (2.2 meters from the whistle) is 138 decibels, will residents near the edge of the town, three kilometers away, be able to hear the lunch whistle? The equation giving the sound level at the edge of town is:

$$L = L_0 - 20 \log_{10} (r/r_0)$$

$$L = 138 - \left[20 \log_{10} (3000/2.2)\right]$$

Where:

L_0 is the sound level at a point near the source (138 dB),

r_0 is the distance from the near point to the source (2.2 m),

L is the sound level at a distant point, and

r is the distance from the distant point to the source (3 km).

Keystrokes	Display	
3000 ENTER↑	3,000.0000	
2.2 ÷	1,363.6364	
LOG	3,1347	
20 ×	62.6940	
CHS	-62.6940	
138 +	75.3060	The sound level 3 kilometers from the firehouse is about 75 dB, well above the level of normal conversation.

The Exponential Function

The y^x function ($Y \updownarrow X$ if you execute it from the display), 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 3^7 (that is $3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3$):

Keystrokes	Display
3 ENTER↑ 7	7_
■ y^x	2,187.0000

Or to calculate $19^{-.0473}$:

Keystrokes	Display
19 ENTER↑	19.0000
.0473 CHS	-.0473_
■ y^x	0.8700

And to calculate $(-16.13)^3$:

Keystrokes	Display
16.13 CHS ENTER↑	-16.1300
3	3_
■ y^x	-4,196.6534

In conjunction with $1/x$, y^x provides a simple way to extract roots. For example, find the cube root of 7: (This is equivalent to $7^{1/3}$.)

Keystrokes	Display	
7 ENTER↑	7.0000	
3 $1/x$	0.3333	Reciprocal of 3.
■ y^x	1.9129	Cube root of 7.

Example: In her study of fish ladders, Jeanneau Colly must determine the rate of water flow down a short spillway on the upper Umpqua River. If the average rate is too great, the salmon run up the Umpqua will be disturbed. Colly finds that the following equation gives the approximate rate of water flow down that spillway:

$$V = \left[(1.49/0.015)\ 1.94^{.67}\right]\ (\sin 38)^{1/2}$$

Keystrokes	Display	
1.49 [ENTER↑]	1.4900	
0.015 [÷]	99.3333	
1.94 [ENTER↑]	1.9400	
.67 [■] [y^x]	1.5589	
[×]	154.8539	
38 [SIN]	0.6157	
2 [1/x]	0.5000	
[■] [y^x]	0.7846	
[×]	121.5047	The flow over the spillway is about 122 cubic feet per second, easy swimming for the average salmon in this spillway.
[■] [CLX]	0.0000	

Statistical Functions

Accumulations

Executing the [Σ+] function automatically gives you several different sums and products of the values in the X- and Y-registers at once. In order to make these values accessible for sophisticated statistics problems, they are automatically placed by the calculator into a block of six storage registers that you define with the [ΣREG] function.

When you execute [ΣREG], the HP-41C prompts you for a register address with *ΣREG _ _*. The address you specify defines the beginning of a block of six statistical registers.

If you have not specified a block of statistical registers using the [ΣREG] function, the statistical registers will automatically be R_{11} through R_{16}. But if you change the location of the statistical registers, that change remains in effect until you change it again, even while the HP-41C is off.

Before you begin any calculations using the [Σ+] key, you should first clear the storage registers used in accumulations by executing the [CLΣ] (clear statistical registers) function.

When you key in a number and press the [Σ+] key, the calculator performs each of the following operations:

1. The number in the X-register is added to the contents of the first statistical register (the first statistical register is presently defined as R_{11}).
2. The square of the number in the X-register is added to the contents of the second statistical register (presently defined as R_{12}).
3. The number in the Y-register of the stack is added to the contents of the third statistical register (presently defined as R_{13}).

4. The square of the number in the Y-register is added to the contents of the fourth statistical storage register (presently defined as R_{14}).
5. The number in the X-register is multiplied by the number in the Y-register, and the product is added to the contents of the fifth statistical storage register (presently defined as R_{15}).
6. The number 1 is added to the contents of the last statistical register (now defined as R_{16}). After all of the above steps are performed by the calculator, the total number in the last statistical register is placed into the display and the X-register.

When you execute [Σ+], the stack and statistical storage register contents are changed...

...from this...				...to this.			
T	t	R_{11}	0.0000	T	t	R_{11}	Σx
Z	z	R_{12}	0.0000	Z	z	R_{12}	Σx^2
Y	y	R_{13}	0.0000	Y	y	R_{13}	Σy
X	x	R_{14}	0.0000	X	n	R_{14}	Σy^2
		R_{15}	0.0000			R_{15}	Σxy
LAST X	0.0000	R_{16}	0.0000	LAST X	x	R_{16}	n

To use *any* of the summations individually at any time, you can recall the contents of a statistical storage register into the X-register by pressing [RCL] and the register address. Or you can recall the contents of the desired storage register into just the display by pressing [■] [VIEW] followed by the statistical register address. Remember that [VIEW] does not disturb the stack registers.

When execution of [Σ+] or [Σ-] causes the contents of any of the statistics registers to exceed $9.999999999 \times 10^{99}$, execution of the function is completed, the contents of all of the statistics registers are updated, and $9.999999999 \times 10^{99}$ is placed in the register or registers that overflowed.

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

Keystrokes	Display	
[■] [CLΣ]	0.0000	Clears the statistical registers (presently R_{11} through R_{16}).
7 [ENTER↑]	7.0000	
5 [Σ+]	1.0000	First pair is accumulated, n = 1.
5 [ENTER↑]	5.0000	
3 [Σ+]	2.0000	Second pair is accumulated; n = 2.

Keystrokes	Display	
9 ENTER↑	9.0000	
8 Σ+	3.0000	Third pair is accumulated; n = 3.
RCL 11	16.0000	Sum of x values in R_{11}.
RCL 12	98.0000	Sum of squares of x values in R_{12}.
RCL 13	21.0000	Sum of y values in R_{13}.
RCL 14	155.0000	Sum of squares of y values in R_{14}.
RCL 15	122.0000	Sum of products of x and y values in R_{15}.
RCL 16	3.0000	Number of entries (n=3).
■ CLX	0.0000	

Note: If your data $\{x_i\}$ or $\{y_i\}$ contains many redundant leading digits, you should refrain from copying them into the calculator. For example, if your x-data is $\{999999999, 1000000001, 1000000002\}$, you should enter the x-data as $\{-1, 1, 2\}$ and add the redundant digits to any x-related answer produced.

Mean

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

When you execute MEAN:

1. The mean of x is calculated using the data accumulated in the first and last statistical registers. (These are the registers that contain Σx and n; presently defined as R_{11} and R_{16}.) The resultant value for mean of x is placed in the X-register.
2. The mean of y is calculated using the data accumulated in the third and last statistical registers. (These are the registers that contain Σy and n; presently defined as R_{13} and R_{16}.) The resultant value for mean of y is placed in the Y-register. Simply press x⇄y to bring that value into the X-register for use.

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

Standard Deviation

The SDEV function is used to calculate the sample standard deviation (a measure of dispersion around the mean) of data accumulated in the statistical registers.

When you execute SDEV:

1. The sample standard deviation of x is calculated using data accumulated in the statistical registers containing Σx, Σx^2, and n. (These registers are presently defined as R_{11}, R_{12}, and R_{16}.) The resultant x value is placed in the X-register.
2. The sample standard deviation of y is calculated using data accumulated in the statistical registers containing Σy, Σy^2, and n. (These registers are presently defined as R_{13}, R_{14}, and R_{16}.) The resultant y value is placed in the Y-register. Simply press x⇄y to place the y value in the X-register for use.

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

Remember, when you use Σ+ to accumulate values into the statistics registers, the values in the X- *and* Y-registers are accumulated. If you do not intend on using the value in the Y-register (you are accumulating only single-variable data), be sure to clear both the X- and Y-registers as well as the statistics registers *before* you accumulate the data using Σ+.

Example: Below is a chart of monthly maximum and minimum winter (October-March) rainfall for a 79-year period in Corvallis, Oregon. What are the average maximum and minimum rainfalls and the standard deviation of the maximum and minimum rainfalls? Rainfall amounts are given in inches.

	October	November	December	January	February	March
Maximum	9.70	18.28	14.47	15.51	15.23	11.70
Minimum	0.10	0.22	2.33	1.99	0.12	0.43

Keystrokes	Display	
▮ CLΣ	0.0000	Clears the statistical registers (still defined as R_{11} through R_{16}).
0 ENTER↑	0.0000	Clears the X- and Y-registers.
9.7 ENTER↑	9.7000	
.10 Σ+	1.0000	First entry. Number of data pairs is now 1.
18.28 ENTER↑	18.2800	
.22 Σ+	2.0000	Second entry. Number of data pairs is now 2.
14.47 ENTER↑	14.4700	
2.33 Σ+	3.0000	
15.51 ENTER↑	15.5100	
1.99 Σ+	4.0000	
15.23 ENTER↑	15.2300	
.12 Σ+	5.0000	
11.70 ENTER↑	11.7000	
.43 Σ+	6.0000	Number of pairs is now 6 (n = 6).

Keystrokes:	Display:	
XEQ	XEQ _ _	
ALPHA MEAN ALPHA	0.8650	Average minimum inches of rainfall per month (mean of x) is in the X-register.
x≷y	14.1483	Average maximum inches of rainfall per month (mean of y) is in the display.
XEQ	XEQ _ _	
ALPHA SDEV ALPHA	1.0156	Standard deviation of minimum rainfall per month (x values) is in the X-register.
x≷y	3.0325	Standard deviation of maximum rainfall per month (y values) is in the display.

The illustration below shows what happens in the stack when you execute MEAN or SDEV. The contents of the stack registers are changed...

... from this ...			... to this.	
T	*t*	→	T	*t*
Z	*z*	→	Z	*z*
Y	*y*	→ lost.	Y	MEAN *of y or* SDEV *of y*
X	*x*		X	MEAN *of x or* SDEV *of x*
	x	→	LAST X	*x*

Deleting and Correcting Data

If you key in an incorrect value and have not executed Σ+, press CLX or ← to delete the incorrect number or digits, and key in the correct number.

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

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

The corrected values for mean and standard deviation are now obtainable by executing the MEAN and SDEV functions.

For example, suppose that you discover a recording error in the data you have gathered on the maximum and minimum rainfalls in Corvallis, Oregon, and you discover that the maximum and minimum values for January are actually 16.61 and 1.99, rather than 15.51 and 1.99. To account for the change in mean and standard deviation values:

Keystrokes	Display	
15.51 [ENTER↑]	15.5100	The incorrect y value.
1.99	1.99_	The incorrect x value.
[■] [Σ-]	5.0000	Incorrect values have been deleted and the number of entries is now 5 (n = 5).
16.61 [ENTER↑]	16.6100	The correct y value.
1.99	1.99_	The correct x value.
[Σ+]	6.0000	The correct values have been summed and the number of entries is now 6.
[XEQ]	XEQ __	
[ALPHA] MEAN [ALPHA]	0.8650	The correct mean of the minimum rainfall per month (mean of x).
[x⇄y]	14.3317	The correct mean of the maximum rainfall per month (mean of y).
[XEQ]	XEQ __	
[ALPHA] SDEV [ALPHA]	1.0156	The correct standard deviation of the minimum rainfall per month (x-values).
[x⇄y]	3.1618	The correct standard deviation of the maximum rainfall per month (y-values).

Operational and General Functions

Audible Tone Functions

The HP-41C is equipped with two functions that allow you to produce audible tones: [BEEP] and [TONE].

When you press [BEEP], the HP-41C produces a series of audible tones.

[TONE], when followed by a number from 0 through 9, will produce a single audible tone. However, [TONE] allows you to control the kind of sound produced. A lower number (0, 1, 2, 3, 4) produces a lower pitched sound, and a higher number (5, 6, 7, 8, 9) produces a higher pitched sound.

Decimal/Octal Conversions

The [OCT] *(decimal to octal)* and [DEC] *(octal to decimal)* functions allow you to convert numbers that are in the X-register to their decimal or octal equivalents. For example, to convert the octal number 326 to its decimal equivalent.

Keystrokes	Display
326	***326_***
[XEQ]	***XEQ __***
[ALPHA] DEC [ALPHA]	***214.0000***

To convert the decimal number 8962 to its octal equivalent:

Keystrokes	Display
8962	***8,962_***
[XEQ]	***XEQ __***
[ALPHA] OCT [ALPHA]	***21,402.0000***

If you attempt to use [OCT] when x is noninteger or the absolute value of x is greater than 1,073,741,823 (decimal), the display will show ***DATA ERROR***. If you attempt to use [DEC] when x is noninteger or the number to be converted contains an 8 or 9, the display will show ***DATA ERROR***. The largest octal number that can be converted is 7,777,777,777.

Exchanging X and Any Register

Earlier in this handbook you learned how [x⇄y] exchanges the contents of the X-register with the contents of the Y-register. Using [X<>] you can exchange the contents of X with the contents of any storage register, including the rest of the stack (T, Z, and Y), and LAST X.

To exchange X with another stack register or LAST X, execute [X<>], press [•] (decimal point) to specify the desired register (T, Z, Y, X, or L for LAST X).

To exchange X with any numbered register from 00 through 99, simply execute [X<>] and supply the two-digit register address.

Paper Advance

This special function, [ADV], is used in the HP-41C when you have the optional printer plugged into an input/output port on the HP-41C.

[ADV] causes the printer paper to advance one line, if the printer is plugged into the HP-41C. In the absence of a printer, [ADV] does nothing. Please consult the owner's handbook included with the printer for additional functions and information.

The HP-41C has five functions that are used to control the operating status of the calculator. They are [ON], [OFF], [AON], [AOFF], and [PRGM]. Notice that [ON] and [PRGM] cannot be recorded as instructions in a program. User mode is controlled either by the [USER] key or by a special USER mode flag. You will learn more about flags in section 14.

Power ON

When you press the [ON] *key,* it simply toggles the HP-41C power on and off. You may remember from section 1 that the HP-41C automatically turns itself off after 10 minutes of inactivity to conserve battery power. When you execute the [ON] *function* ([XEQ] [ALPHA] ON [ALPHA]), the turn-off feature is disabled and the HP-41C will no longer automatically turn itself off. The [ON] function stays in effect until you turn the HP-41C off.

Power OFF

When executed from the display or in a program, the [OFF] function simply turns the HP-41C power off.

PRGM Mode

[PRGM], which toggles the HP-41C in and out of program mode, can only be executed by pressing the [PRGM] key on the HP-41C keyboard. There is no display execution form of [PRGM]. In addition, [PRGM] cannot be recorded as an instruction in a program.

ALPHA Mode

The [AON] *(ALPHA mode on)* function places the HP-41C into ALPHA mode, and [AOFF] *(ALPHA mode off)* takes the HP-41C out of ALPHA mode. [AON] and [AOFF] are most useful in programs. In addition, notice that [AON] and [AOFF] perform the same function as the [ALPHA] *key* on the keyboard.

PART II
Programming the HP-41C

Section 7

Simple Programming

Even though the HP-41C has many powerful functions, you may wish to perform operations that are not already contained in the calculator. If you have read through the introduction to this handbook, you have already seen how you can increase the capability of the HP-41C greatly by writing your own programs.

Once these programs are stored into the calculator's program memory, they can be executed exactly like *any* of the standard HP-41C functions.

The HP-41C even allows you to define the arrangement of the keyboard. You can completely customize the calculator by writing your own specialized functions and assigning them to the keyboard locations *you* specify.

After most of the explanations and examples in this part of the handbook, you will find problems to work that let you practice programming the HP-41C. These problems are not essential to your basic understanding of the calculator, and they can be skipped if you like. But we urge you to work them. Each problem has been designed to increase your proficiency in programming and use of the HP-41C.

If you are familiar with other Hewlett-Packard handheld calculators, you still may wish to work through part II of this handbook. The HP-41C has many new capabilities that you can take advantage of in your programs. Programming the HP-41C is simple, just like on all other HP handheld calculators.

Note that in programming, there are usually several ways that a problem can be solved. So after you complete this handbook, you may find that you will be able to solve many of the problems faster, or in fewer instructions, than we have shown in our illustrations.

Now let's begin programming!

What Is a Program?

A program is little more than a series of keystrokes that you would press to solve a problem manually. Except that when you program, the calculator *remembers* the keystrokes as you enter them, then executes all of the specified keystrokes whenever you wish. Because of the special capabilities of the HP-41C, programs that you write can be treated just like any other function on the calculator.

Creating a Program

If you read the introduction of this handbook, you created, loaded, and ran a program that calculated the heat loss from a cylindrical water heater. Now let's create, load, and run another program to show you how to use some of the other features of the HP-41C.

One value you needed in order to calculate the heat loss from the water heater was the surface area of the cylinder. Let's begin the next problem by calculating just the surface area of the top of the cylinder, which, of course, is a circle. The formula for the area of a circle is $A=\pi r^2$.

To calculate the area of the circle manually you would first key in the radius r, and then square it by pressing $\boxed{x^2}$. Next you would press $\boxed{\pi}$ to summon the quantity pi. Finally you would multiply the squared radius and the quantity pi together by pressing $\boxed{\times}$.

Remember that a program is little more than the keystrokes you would press to solve the problem manually. So, in the program, the keys you press to solve the problem are the same as the keys you press to solve the problem manually. You will load these keystrokes into program memory:

$\boxed{x^2}$
$\boxed{\pi}$
$\boxed{\times}$

In addition, your program will contain two other operations, $\boxed{\text{LBL}}$ and $\boxed{\text{END}}$.

The Beginning of a Program

The beginning of each program you write should be "named" or labeled with a string of ALPHA characters, *or* a two-digit number. These program labels enable you to keep track of, and easily use, the programs you write. In a few moments you will learn how to use $\boxed{\text{LBL}}$ *(label)* to label your programs. First, there are a few things that you should know about labels.

Program labels that are ALPHA characters can consist of any seven ALPHA characters *except* , (comma), . (period), : (colon).

Used as program labels, the single letters A through J and a through e have a special "local label" function in the HP-41C. These single letters should not be used as the first label in your program. They are most useful when used inside programs. Don't be concerned with local labels now—they are covered in detail in section 12. For now, just remember not to label your main programs with A through J and a through e.

Program labels that are numbers must be two digits. Number labels are most often used to label subroutines. Use of numeric labels is covered later.

The HP-41C makes labeling programs easy. (Later, you will see how the calculator actually prompts you for the label characters.) While you are keying in an ALPHA label, the calculator *ignores* improper characters (e.g., , . :) and does not accept any more than seven characters. The HP-41C does not accept any more than two digits in a numeric label.

Here are some examples of proper and improper program labels:

Proper ALPHA	**Proper Numeric**
TRIGO1	00
GO	83
A (Used as a local label.)	06

Improper ALPHA	**Improper Numeric**
RUN. (Illegal period in name.)	1 (Too few digits.)
COMPUTER (Too many characters.)	382 (Too many digits.)

Label Usage. Following are some considerations that you may find helpful in labeling your programs.

- Numeric labels can be used any number of times, even in the same program.
- If you label and execute a program with the same name used by the HP-41C for one of the HP-41C standard functions (e.g., DEG, ABS, etc.) or for a program in a plug-in application module, the calculator will first search program memory for the program name. If it is found, the HP-41C will execute the named *program*. If the name is not found as a label in program memory, the HP-41C will then execute the standard HP-41C function or the application module function having the same name.

The Complete Program

The complete program to solve for the area of a circle (one end of our cylindrical water heater) given its radius is now:

■ LBL ALPHA CIRCLE ALPHA	Assigns the name (CIRCLE) to and defines beginning of the program.
■ x^2	Squares the radius.
■ π	Summons pi.
×	Multiplies r and π to give the area of the circle.
END	Defines end of program space in memory and stops the program (more about END later).

Loading a Program

When the HP-41C is in PRGM *(program)* mode, 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 the following operations can be loaded into program memory for later execution.*

CLP (clear program)	SIZE (number of storage registers)
← (correction)	PRGM (program mode key)
BST (back step)	GTO • (go to line number)
SST (single step)	CATALOG (catalog list)
DEL (delete program lines)	ON (continuous power)
ASN (assign)	ON (power on key)
USER (USER mode key)	COPY (copy or download program)
	GTO • • (go to end of program memory)

All other functions are loaded into the calculator as program instructions to be executed later. Functions on the keyboard are loaded by simply pressing the associated keys. Functions not on the keyboard are loaded by assigning the function to a key and pressing that key in USER mode, or using XEQ and the function name—just like you would if you were executing the function manually. (Refer to section 4 if you need to refresh your memory.)

To load the complete program into the calculator:

1. Press PRGM to place the HP-41C into program mode.
2. Press ■ GTO • • to set the HP-41C to an unused portion of program memory.

Using GTO • •. When you press GTO • •, the calculator is positioned to the end of program memory (after the last existing program in program memory), and is ready for you to begin keying in the instructions of your program. The display will show **00 REG nn**. The nn indicates the number of registers that are unused in program memory (more about this later).

In addition to positioning the calculator to the end of program memory, GTO • • also checks to see if the last program you keyed in was terminated with an END instruction. If an END was not keyed in as the last instruction of that program, GTO • • automatically inserts one. In this way the HP-41C automatically maintains program memory for you!

You can see that GTO • • is extremely useful. Before you begin keying in a program, simply press GTO • •. When you are finished, press GTO • • and the calculator tells you how many registers are left in program memory before and after you key in your program.

Keystrokes	Display	
PRGM	00 REG 46	Places the HP-41C into program mode.
■ GTO • •	00 REG 46	The HP-41C is now ready for you to begin programming.

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

■ LBL ALPHA CIRCLE ALPHA
■ x^2
■ π
×

Press the first keys, ■ LBL, of the program.

Keystrokes	Display
■ LBL	01 LBL _ _

The digits that appear at the left of the display indicate the *program memory line number* being shown at any time. We will learn more about "lines" later in this section. Now press the ALPHA keys necessary to complete the instruction.

Keystrokes	Display
ALPHA CIRCLE ALPHA	01 LBL ^{T}CIRCLE

Any time a program line contains an ALPHA label or ALPHA string, the HP-41C places ^{T} (raised t, for "text") in the display following the program line number. Notice that as you press function keys for the program, the HP-41C prompts you for the input, just like in normal mode operation.

Now load the rest of the program:

Keystrokes	Display
■ x^2	02 X↑2
■ π	03 PI
×	04 *

Now press ■ GTO • • . This places an END at the end of the program (in line 5) and tells you how many registers are left in program memory. Notice the ***PACKING*** display appears momentarily—packing is covered in detail later.

Keystrokes	Display	
■ GTO • •	PACKING 00 REG 44	This places an END in line 5 and tells you how many registers are left in program memory.

The program for solving the area of a circle (named CIRCLE) is now loaded into program memory.

Running a Program

To run a program you can either execute it using the [XEQ] key, or you can assign it to a key and execute it by pressing that key in USER mode. Let's try it both ways. You will find the USER mode operation saves you time and keystrokes.

When you run a program, the HP-41C has two program execution annunciators that appear in the display. As program execution progresses, a ⟩- appears in the display. Each time the program executes a program label, the ⟩- moves across the display one position to the right. When the ⟩- is in the last position on the right of the display, the ⟩- resets back to the left of the display.

As an additional aid, the HP-41C also turns on the **PRGM** annunciator in the display while a program is executing. When the program has completed execution, the **PRGM** annunciator turns off.

After a program executes an [AVIEW] or [VIEW], the ⟩- will not appear, but the **PRGM** annunciator will be displayed.

These aids provide an indication to you that the calculator is executing a program. You never have any doubt during the execution of a long program; you can easily determine that the calculator is operating.

Take the HP-41C out of **PRGM** mode now by pressing [PRGM]. Notice that the **PRGM** display annunciator turns off.

Keystrokes	Display
[PRGM]	0.0000

Next, use the CIRCLE program you created to find the area of two circles with radii of 14 inches and 0.55 meters:

Keystrokes	Display	
14	14_	The first radius in inches.
[XEQ]	XEQ __	The HP-41C prompts: *Execute what?*
[ALPHA] CIRCLE [ALPHA]	615.7522	The answer in square inches.
.55	.55_	The second radius in meters.
[XEQ]	XEQ __	The prompt.
[ALPHA] CIRCLE [ALPHA]	0.9503	The answer in square meters.

Now, assign CIRCLE to the [LN] key location and find the area of two more circles with radii of 10.7 inches and 0.439 meters.

Keystrokes	Display	
■ ASN	ASN _	The HP-41C prompts: *Assign what?*
ALPHA CIRCLE ALPHA	ASN CIRCLE _	*Assign CIRCLE to which key location?*
LN	SN CIRCLE 15	The CIRCLE function is assigned to row 1, column 5 (LN). You can see the keycode assignment if you hold the key down momentarily.
USER	0.9503	Places the HP-41C in USER mode. Any functions you have assigned to the keyboard become active. The displayed number remains from the previous example.
10.7 CIRCLE (LN)	359.6809	Since CIRCLE is assigned to LN, when you press LN in USER mode, CIRCLE is executed. The answer is shown in square inches.

Now compute the area of the second circle. But this time, hold the function key down momentarily. Notice that the HP-41C prompts you with the USER mode function name. (When the calculator is set to normal mode and you press and hold the key, the HP-41C prompts you with the normal mode function name.)

Keystrokes	Display	
.439 CIRCLE (LN)	ᵀCIRCLE	Hold the key down momentarily.
	0.6055	Square meters.
■ CLX	0.0000	
USER	0.0000	Takes the HP-41C out of USER mode.

USER mode execution is that simple! It lets you execute functions you have written just like any other function on the HP-41C, and *you* control the keyboard location. To completely customize your HP-41C, you simply assign programs and functions to the locations you specify.

Unlike the standard HP-41C functions (which can *each* be assigned to several key locations), you can only assign a program that you have written to a single key location. The last key assignment that you specify is the only one that applies.

Included with your new HP-41C are some aids to help you label the keyboard for USER mode operation. There are plastic overlays on which you can write function names, and there are pre-printed sticky-back labels printed with the name of each standard HP-41C function. When you reassign a function to the keyboard, simply write the function name on an overlay, or if the function is a standard HP-41C function, place its corresponding label in place on an overlay. When the calculator is in USER mode, simply put the overlay in place. Notice also that blank sticky-back labels are provided so you can write on them and stick them in place on an overlay.

The reassigned keys remain reassigned in USER mode until you clear the corresponding programs from program memory or reassign the key location again. For example, [CIRCLE] will remain assigned to the [LN] key location until you clear [CIRCLE] from program memory or reassign the [LN] key again.

Program Memory

You may remember from section 5 that program memory and storage registers both store information in the calculator's memory. Memory can be defined for use either as program memory *or* storage registers. When a portion of memory is defined for use as program memory, the calculator stores the program information in these registers. A single, complete operation stored into program memory is called an *instruction* or line.

What Are Instructions and Lines?

The HP-41C has been designed so that you need not worry about program memory structure—all you need to do is key in your program instructions—the HP-41C takes care of the memory, automatically. If you find that you need to know the relationship between instructions and program memory, appendix D lists all HP-41C instructions and the byte requirements of each, as well as a brief explanation of how program memory is structured.

An *instruction* or line in a program is a series of keystrokes that make up one complete operation in a program. Each complete instruction is given a line number. Line numbers are what appear in the display when you load a program. Depending on the kind of instructions keyed in, you can store up to seven instructions in each program memory register. But again, you need not be concerned with the details of program memory because the HP-41C takes care of them for you.

Instructions consist of a single function and all of the inputs necessary to complete the operation. Complete numbers in a program are treated as single instructions and take up only one line (e.g., 124.75 is one line). Examples of instructions are [COS], [FIX] 6 and [TONE] 3. [COS] alone *is* a complete instruction because it performs a single operation and does not require additional input or data. But [FIX] and [TONE] alone are *not* complete instructions. Since [FIX] and [TONE] both require number inputs to complete the operation, their instructions are not complete until the number is included. [FIX] 4 and [TONE] 8 are examples of complete operations.

When a program line contains an instruction whose name is too long to display all at one time, the HP-41C "scrolls" the information through the display. Section 8 shows how [SST] and [BST] can be used to view these program lines.

The Basic HP-41C and Initial Configuration

The HP-41C comes standard with 63 registers. Initially, the HP-41C allocates 17 of these to data storage registers and the remainder (46) to program memory.

Changing Memory Allocations

If at any time you fill program memory with programs and attempt to load more instructions, the HP-41C will pack program memory and display ***TRY AGAIN*** (more about packing later). When program memory is full, each time you attempt to load an instruction the calculator packs program memory and again displays ***TRY AGAIN***. By executing [SIZE] *(size of data register allocation)* you can change the number of registers that are allocated to program memory and data storage registers to make room for more program instructions (or to change the number of data storage registers).

When you execute [SIZE], the HP-41C prompts you for a three-digit number from 000 through 319. [SIZE] specifies the *total* number of registers allocated to *data storage registers* only. When you change the data storage register allocation, the number of registers in program memory is automatically changed. If you increase the storage register allocation, the number of registers in program memory decreases; if you decrease the number of data storage registers, the number of registers in program memory automatically increases.

Note that if you execute [SIZE] and attempt to decrease the number of registers in program memory when those registers contain program instructions, the HP-41C will pack program memory and display ***TRY AGAIN***. Before you can change program memory into data storage registers, you must clear enough program instructions out of program memory to make room for the reallocation. This prevents you from accidentally losing program instructions when you execute [SIZE].

For example, if you change the number of data storage registers from 17 to 21, program memory automatically decreases in size. You are adding four registers to data storage registers, and that *decreases* the number of registers allocated to program memory by four. Note that data storage registers are numbered 000-318. So [SIZE] 017 allocates R_{00} through R_{16} to data storage registers.

Initial Allocation		New Allocation	
Data Storage Registers	**Registers in Program Memory**	**Data Storage Registers**	**Registers in Program Memory**
17 (R_{00} through R_{16})	46	21 (R_{00} through R_{20})	42

Each register you add as a data storage register removes one register from program memory, and each register you remove from the data storage registers adds one register to program memory.

Keystrokes	Display	
[XEQ]	*XEQ __*	
[ALPHA] SIZE [ALPHA]	*SIZE ___*	*What is the desired data storage register allocation?*
021	*0.0000*	The allocation is now 21 registers as storage registers and 42 registers in program memory.
[XEQ]		
[ALPHA] SIZE [ALPHA]	*SIZE ___*	Returns to the normal allocation.
017	*0.0000*	

The minimum/maximum allocations of registers are 0 data storage registers and 63 registers in program memory (319 registers with four additional memory modules), or 63 data storage registers (319 with four additional memory modules) and 0 registers in program memory.

Continuous Memory

Programs that you write and record in program memory remain there permanently until you explicitly remove them. The Continuous Memory of the HP-41C saves the programs permanently, even when the calculator is turned off.

The [END] Function

As you read earlier, when you enter more than one program into program memory, you should separate those programs using [END]. Following is a short description of how [END] works.

[END] tells the calculator that the end of a program's space in program memory has been reached and all subsequent lines belong to another program. For example, program memory now looks like this:

00	
01 LBLᵀHEAT	
02 30	This program was entered in the introduction of this handbook.
03 ∗	
04 .47	
05 ∗	
06 END	
00	
01 LBLᵀCIRCLE	
02 X↑2	
03 PI	
04 ∗	
05 END	The end of the program and its space in program memory.

Remember that the HP-41C will automatically insert an END for you when you press GTO • •.

When you press GTO • • to begin a new program, the new instructions are added after the last END instruction in program memory. The HP-41C makes program memory management so easy that you need not worry about where programs are positioned in program memory. Just press GTO • • before you begin each program and the positioning is done for you.

There is a permanent END located at the current bottom of program memory. It cannot be deleted and instructions cannot be inserted after it. For this reason, even though the basic HP-41C actually has 64 registers, a portion of one register is consumed by the permanent END, designated *.END.* when displayed. Thus, you see ***00 REG 46*** when you press GTO • • the first time.

For the purposes of this book, a "program" or a "program file" is everything between (and including) the initial LBL for the program and the END of the program.

Clearing Programs

You can clear any program you have loaded into program memory by simply executing CLP (clear program) and specifying the program name.

CLP clears all instructions of a program including the program label and the program's END instruction. For this reason, it is important to include END instructions in your programs. For example, if program memory looked like this ...

00	
01 LBLT TEST1	Program "TEST1."
02 LOG	
03 +	
04 STO 10	
05 RTN	
06 LBLT TEST2	Program "TEST2."
07 LOG	
08 −	
09 STO 11	
10 RTN	

... and you cleared TEST1, *all* of the instructions from line 00 of TEST1 down to the first END (if one existed) would be cleared. But if you include END instructions, you can selectively clear programs from program memory. For example, if program memory looked like this, you could clear just TEST1 or TEST2.

00 **01 LBL⊤TEST1** **02 LOG** **03 +** **04 STO 10** **05 END**	You could clear just these instructions by executing [CLP], specifying TEST1 as the program name, or . . .
00 **01 LBL⊤TEST2** **02 LOG** **03 −** **04 STO 11** **05 END**	. . . you could clear just these instructions by executing [CLP], specifying TEST2 as the program name.

When you execute [CLP] and do not specify a function name (press [ALPHA] [ALPHA]), the HP-41C clears the program the calculator is currently positioned to in program memory.

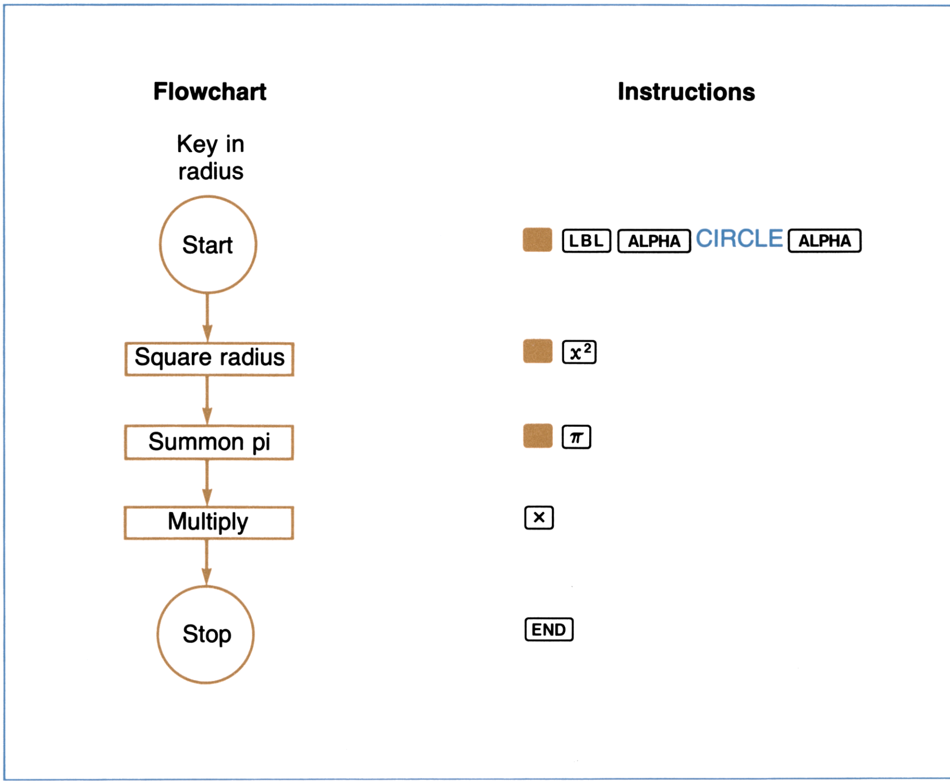
To clear the entire calculator (all programs, registers, assignments, flags, etc.) with the "master clear:" turn the HP-41C off, hold down the [←] key, and turn the calculator back on again. The display will show ***MEMORY LOST***.

Flowcharting Your Programs

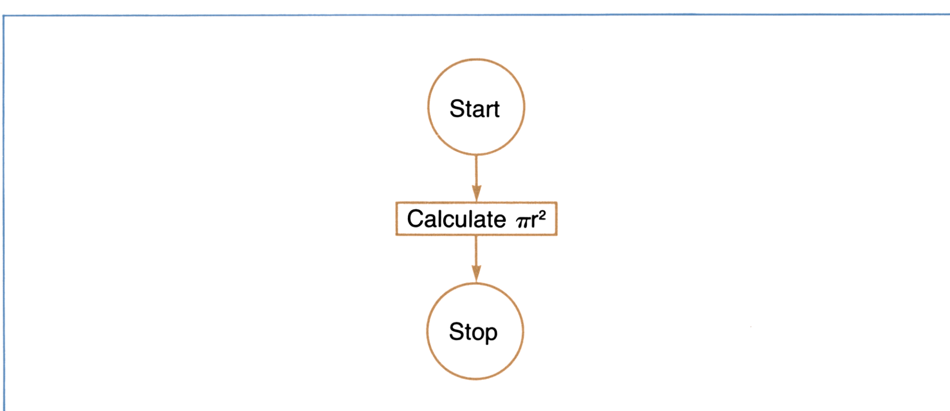
At this point, we digress for a moment from our discussion of the calculator itself to familiarize ourselves with a fundamental programming tool—the flowchart.

Flowcharts are *outlines* of the way a program solves a problem. With over 400 possible lines (2200 on a fully-enhanced HP-41C), it is quite easy to get "lost" while creating a long program, especially if you try to load a program from beginning to end with no breaks. A flowchart can help you design your programs by breaking them down into smaller groups of instructions.

Flowcharts 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:



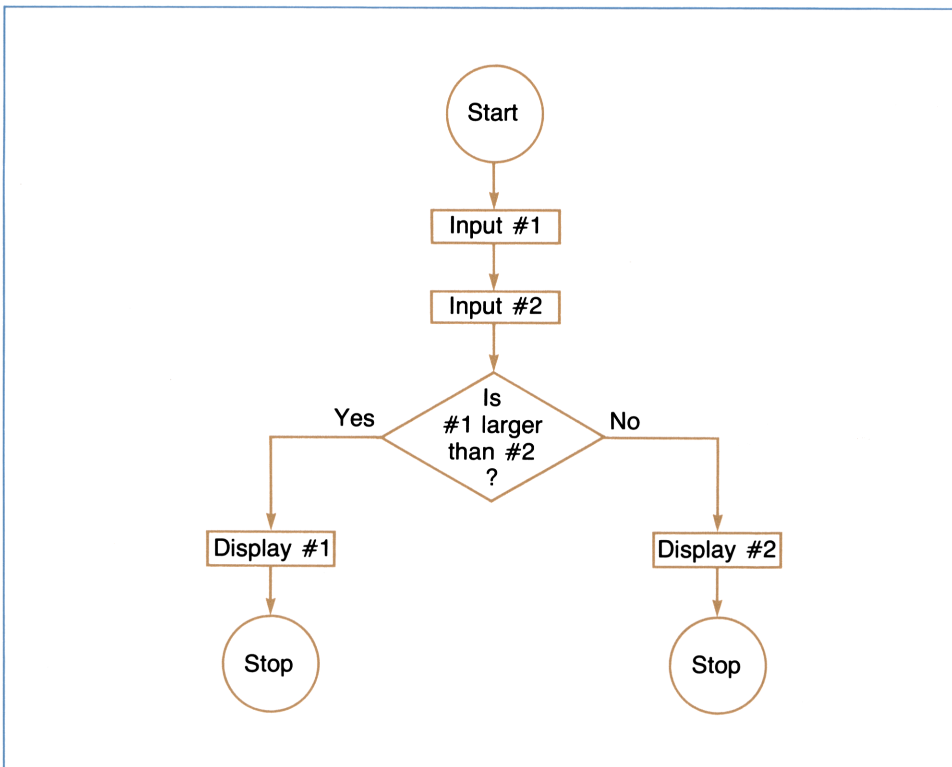
You can see the similarities between the program and the flowchart. At times, a flowchart may duplicate the set of instructions exactly, as shown above. At other times, it may be more useful to have an entire group of instructions represented by a single block in the flowchart. For example, here is another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



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.

Flowcharts are drawn linearily, from top to 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 the functional operations in a program. We use diamonds to represent *decisions,* where the program must decide which of two alternatives to take.

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:



It would be a simple matter to go back and insert groups of instructions for each element of the flowchart. As you work through this handbook, you will become more familiar with flowcharts. Flowcharts will help you to organize, eliminate errors in logic and flow, 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 draw a flowchart and write a function 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 **GTO** **•** **•** before you begin programming. Name the program with ■ **LBL** **ALPHA** RADIUS **ALPHA** and terminate it with **END** (use **GTO** **•** **•**). After you have loaded the program, run it to calculate the radii of circles with areas of 420 square inches, 1.2 square meters, and 0.9095 square meter.

   (Answers: 11.5624 inches, 0.6180 meter, 0.5381 meter.)

2. Write and load a program that will convert temperature in degrees Celsius to degrees Fahrenheit, according to the formula $F=(1.8\times C)+32$. Name the program CTEMP and terminate it with **END**. Convert Celsius temperatures of $-40°$, $0°$, and $18°$.

   (Answers: $-40.0000°$ F, $32.0000°$ F, $64.4000°$ F.)


4
LINE
4

Section 8

Program Editing

Often you may wish to alter or add to a program that you have keyed into the calculator. The HP-41C has several editing functions that permit you to easily change any lines in any of your programs *without* reloading the entire program.

Editing Functions

Here are the HP-41C editing functions and what they do:

CLP *(clear program)* Clears the named program from program memory. If the program or an ALPHA label inside the program has been assigned to a key for USER mode execution, those assignments are also nullified.

← *(correction)* In PRGM (program) mode, deletes keystrokes while you are entering data or ALPHAs, or deletes entire lines that are already stored in program memory.

SST *(single step)* In PRGM mode, **SST** steps forward one line in program memory. In normal or USER mode, **SST** executes the current line and steps forward one line in program memory. Also, while you are using **CATALOG**, **SST** steps forward one entry.

BST *(back step)* In PRGM, normal and USER modes, **BST** steps back one line in program memory; no instructions are executed. Also, while you are using **CATALOG**, **BST** steps back one entry.

GTO **•** *(go to line number or ALPHA label)* When you specify a three-digit line number, sets calculator to that line. When you specify an ALPHA label, sets the calculator to that label. Pressing **GTO** **•** **•** sets the calculator to the end of program memory and tells you the number of unused registers that remain in program memory. This also places an **END** at the end of the previous program in program memory if one is not already present.

SIZE *(size of data storage register allocation)* When you specify a three-digit number indicating an allocation of registers to data storage registers, program memory is automatically adjusted and all remaining registers are allocated to program memory. Any time the HP-41C repeatedly displays *TRY AGAIN*, you must change the number of storage registers (which automatically changes the size of program memory) before you continue. Refer to section 7.

DEL *(delete program memory lines)* When you specify a three-digit number, the HP-41C deletes that number of lines beginning at the current position in memory. The **DEL** function only deletes instructions within a program and up to (but not including) an **END**

instruction. If you specify a delete number that extends across the END of a program, the HP-41C only deletes up to the END of the program and stops. If you attempt to delete more lines than you have allocated, the calculator simply deletes lines up to the end of program memory or an END instruction and stops.

Let's load a program into program memory and use the editing features to modify it.

To determine the heat loss from a cylindrical water heater, you need to know three things: the area of the cylinder, the convective heat transfer coefficient, and the temperature difference between the cylinder surface and the surrounding air. In the introduction to this handbook, you wrote a program (HEAT) that determined the heat loss from the water heater given the area, the heat transfer coefficient, and the temperature difference. In section 7, you wrote a program called CIRCLE to determine the surface area of one end of the cylinder.

Now let's write and load a program that determines the total surface area of the cylinder given its height (h) and radius (r). The formula used is $S=(2\pi r^2)+(2\pi rh)$. Below are the instructions for the program, assuming that the radius and the height have been placed into the X- and Y-registers of the stack, respectively. The name of the program is AREA.

Keystrokes	Display	
PRGM	00 REG 44	Places the HP-41C into program mode. HP-41C is positioned to the top of the previous program you executed.
■ GTO • •	00 REG 44	Sets the HP-41C to the end of program memory and tells you the number of unused registers left in program memory.
■ LBL ALPHA AREA ALPHA	01 LBL^TAREA	Names the program.
STO 01	02 STO 01	Stores the radius (r) into storage register R_{01}.
■ x^2	03 X↑2	Squares the radius (r^2).
■ π	04 PI	Summons the quantity pi.
×	05 *	Multiplies r^2 by π.
2	06 2_	
×	07 *	Computes $2\pi r^2$.

Keystrokes:	Display:	
[x⇄y]	08 X<>Y	Moves the height (h) into the X-register.
[RCL] 01	09 RCL 01	Recalls the radius (r) from storage register R_{01}.
[×]	10 *	Multiplies r and h (rh).
■ [π]	11 PI	Summons the quantity π.
[×]	12 *	Computes πrh.
2	13 2_	
[×]	14 *	Computes 2πrh.
[+]	15 +	Computes $S=(2\pi r^2) + (2\pi rh)$.
■ [GTO] [•] [•]	00 REG 40	Ends the program and tells you how many registers are left in program memory.

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

Initializing a Program

When you initialize a program, all you do is set up all of the required inputs and mode settings prior to the actual running of it. Some programs contain initializing routines that set up the data to run the program. In other programs, like AREA, you may have to initialize the program manually from the keyboard.

In our AREA program, we must place the height (h) into the Y-register of the stack and the radius (r) into the X-register. To initialize AREA with the values of 50 inches for h and 11 inches for r:

Keystrokes	Display	
[PRGM]	0.0000	Takes the HP-41C out of program mode.
50	50_	The h value.
[ENTER↑]	50.0000	The h value is in the Y-register
11	11_	The r value is in the X-register.

The AREA program, which solves for the total area of a cylinder, is now initialized for height of 50 inches and radius of 11 inches.

Running the Program

To run AREA you only have to execute it using XEQ or assign it to the keyboard for single-key execution. For ease of use, let's assign it to the LOG key location and then execute it in USER mode. When you assign a program (that you have stored into program memory) to a key location, the calculator remembers the assignment by storing it with the LBL of the program.

Keystrokes	Display	
■ ASN	ASN _	
ALPHA AREA ALPHA	ASN AREA _	
LOG	11.0000	AREA is now assigned to the LOG key location for USER mode execution.
USER	11.0000	Places the HP-41C into USER mode so you can use the reassigned key.
AREA (LOG)	4,216.0173	The area of the cylinder in square inches.

Now compute the area of a cylindrical water heater that has a height of 58.185 inches and a radius of 9.25 inches.

Keystrokes	Display	
58.185 ENTER↑	58.1850	
9.25	9.25 _	AREA is initialized with a new set of data before execution.
AREA (LOG)	3,919.2861	Total area of the cylinder in square inches.

Let's see how the HP-41C editing functions can be used to examine and modify AREA.

Resetting to the Beginning of a Program

To begin editing a program, you may need to set the calculator to the beginning of that program. There are several ways to do this, depending on the status of the calculator and your personal preference.

To reset to the beginning of the program:

1. In normal or USER mode, if the calculator is already positioned to a line in the desired program (e.g., if you have just executed the program), press ■ RTN . This sets the calculator to line 0 of the current program.
2. In normal, USER, or PRGM mode, if the calculator is already positioned to a line in the desired program (e.g., if you have just executed the program), press ■ GTO • 000. This sets the calculator to line 000 of the current program.

3. In normal, USER *or* PRGM mode, press ■ [GTO] [•] and specify the program name (e.g., ■ [GTO] [•] [ALPHA] AREA [ALPHA] positions the calculator to the ALPHA label named AREA in program memory).

To reset to the beginning of AREA:

Keystrokes	Display	
■ [GTO] [•]		
[ALPHA] AREA [ALPHA]	3,919.2861	Number remains from previous example.

You could also have used ■ [RTN] or ■ [GTO] [•] 000 to reset the calculator to the begining of the AREA program.

Set the HP-41C to PRGM mode to verify that the calculator is now set to the beginning of AREA. Make sure to set the calculator back to normal mode.

Keystrokes	Display	
[PRGM]	01 LBL⊤AREA	Program mode. Line 1 of AREA.
[PRGM]	3,919.2861	Back to normal mode.

Single-Line Execution of a Program

In normal or USER mode, you can execute any program you have stored in program memory one line at a time by pressing the [SST] *(single step)* key.

To execute one line of AREA at a time using a height of 132 centimeters and a radius of 29.21 centimeters, you must first initialize the program:

Keystrokes	Display	
132 [ENTER↑]	132.0000	The height.
29.21	29.21 _	The radius.

Now press [SST] and hold it down to see the instruction in the next line. When you release [SST], the next instruction is executed. (If you hold it down too long, the [SST] will be nullified.)

Keystrokes	Display	
[SST]	01 LBL⊤AREA	Instruction in line 1 is seen when you hold [SST] down.
	29.2100	The [LBL] AREA instruction is executed when you release [SST].

The first instruction of AREA is executed when you press and release [SST]. Continue executing the program line by line by pressing [SST]. When you hold [SST] down, you see the instruction in the next line of the program. When you release [SST] that instruction is executed.

Keystrokes	Display	
[SST]	02 STO 01	The next line.
	29.2100	Executed.
[SST]	03 X↑2	The next line.
	853.2241	Executed.
[SST]	04 PI	
	3.1416	
[SST]	05 *	
	2,680.4826	
[SST]	06 2	
	2.0000	
[SST]	07 *	
	5,360.9651	
[SST]	08 X<>Y	
	132.0000	
[SST]	09 RCL 01	
	29.2100	
[SST]	10 *	
	3,855.7200	
[SST]	11 PI	
	3.1416	
[SST]	12 *	
	12,113.1016	
[SST]	13 2	
	2.0000	
[SST]	14 *	
	24,226.2033	
[SST]	15 +	
	29,587.1684	
[SST]	16 END	
	29,587.1684	

When you press [SST] and reach the [END] of a program, the next press of [SST] positions the calculator back to the beginning of the program. You can see that the use of [END] instructions is important.

You have seen how [SST] can be used in normal or USER mode to execute a program one line at a time. Using [SST] in this manner can help you create programs and locate errors in them. Now let's see how you can use [SST], [BST], and [GTO] [•] nnn in PRGM mode to help you modify a program.

Modifying a Program

Since you have just completed the execution of the AREA program, the calculator is set back to the beginning of the program. You can verify this by placing the calculator into PRGM mode (press [PRGM]). Press [SST] once to see the program label.

Keystrokes	Display	
[PRGM]	*00 REG 40*	
[SST]	*01 LBL⊤AREA*	The line number and instruction are displayed in PRGM mode.

Now let's modify the AREA program so that the X-register contents will automatically be displayed at certain points in the program. We will accomplish this by placing [PSE] (pause) instructions in the program to halt the program and display the contents of the X-register for about one second, then resume execution. (More about [PSE] later.)

Program	
00	
01 LBL⊤AREA	
02 STO 01	
03 X↑2	
04 PI	
*05 **	
06 2	
*07 **	We will insert a [PSE] after this line to display the area of the top of the cylinder ...
08 X<>Y	
09 RCL 01	
*10 **	
11 PI	
*12 **	
13 2	
*14 **	... and a [PSE] after this line to display the area of the cylinder without the top and bottom.
15 +	
16 END	

To begin modifying your program, reset the calculator to line 0 of AREA.

Keystrokes	Display
■ [GTO] [•] 000	*00 REG 40*

Single-Line Viewing Without Execution

You can use SST *(single step)* and BST *(back step)* in PRGM mode to single-step to the desired line of program memory *without* executing the program. Each press of SST steps forward one line in the program, and each press of BST steps back one line in the program.

Both SST and BST operate only within the current program. Pressing SST when the calculator is set to the end of a program positions the calculator back to the beginning of that program. In a similar way, pressing BST when the calculator is set to the top of a program positions the calculator around to the end of *that* program.

00
01 LBLTAREA
16 END

Pressing SST when the calculator is positioned here, moves the calculator back to the beginning of this program.

00
01 LBLTAREA
16 END

Pressing BST when the calculator is positioned at the top of the program moves the calculator to the end of this program.

Lines in a program with names longer than the display are "scrolled" through the display to the left. SST and BST can be used to view all program lines repeatedly, even long instruction names that are scrolled.

Remember, in normal and USER modes, SST is used to *execute* programs one line at a time, and in PRGM mode, SST is used to view programs without execution. However, BST is used for *viewing* only and does not execute in PRGM, normal, or USER modes.

Keystrokes	Display	
	00 REG 40	The top of the program.
SST	01 LBLTAREA	SST moves the calculator forward one line with each press.
SST	02 STO 01	
BST	01 LBLTAREA	BST moves the calculator back one line with each press.

Now, use SST to move the calculator down to line 7 so that you can insert the PSE (pause) instruction.

Keystrokes	Display	
[SST]	02 STO 01	
[SST]	03 X↑2	
[SST]	04 PI	
[SST]	05 *	
[SST]	06 2	
[SST]	07 *	We will insert a [PSE] after line 7.

You can see that the HP-41C is now set at line 7 of program memory. If you press a *recordable* operation now, it will be loaded into the *next* line, line 8, of program memory, and all subsequent instructions will be "bumped" down in program memory.

Thus, to load the [PSE] instruction so that the program will review the contents of the X-register:

Keystrokes	Display	
[XEQ]	08 XEQ _ _	
[ALPHA] PSE [ALPHA]	08 PSE	The [PSE] instruction is now stored in line 8.

Now let's see what happened in program memory when you loaded the [PSE] instruction. With the calculator set to line 7, when you loaded the [PSE], program memory was altered ...

... from this ...	... to this.
00	00
01 LBL$^{\mathsf{T}}$AREA	01 LBL$^{\mathsf{T}}$AREA
02 STO 01	02 STO 01
03 X↑2	03 X↑2
04 PI	04 PI
05 *	05 *
06 2	06 2
07 *	07 *
08 X<>Y	08 PSE
09 RCL 01	09 X<>Y
10 *	10 RCL 01
11 PI	11 *
12 *	12 PI
13 2	13 *
14 *	14 2
15 +	15 *
16 END	16 +
	17 END

The [PSE] instruction was inserted here.

All subsequent instructions are "bumped" down in program memory.

When you inserted an instruction in the program, all instructions after the one inserted are moved down. Note that if you begin adding instructions and the display shows ***TRY AGAIN***, you should attempt to insert the instruction again. If the display again shows ***TRY AGAIN***, you will need to stop and execute the [SIZE] function presented in section 7, to change the number of data storage registers. Decreasing the number of data storage registers will automatically increase the size of program memory. For further explanation, refer to section 7.

Going to a Line Number

It is easy to see that if you wanted to single-step from line 000 to some remote line number in program memory, it would take a great deal of time and a number of presses of the [SST] key. So using the [GTO] [•] nnn function, you can set the calculator to *any* line in the program. (■ [GTO] [•] nnn cannot be recorded as a line in a program.)

Whether the calculator is set to PRGM mode or normal mode, when you press [GTO] [•] nnn, the calculator immediately jumps to the program memory line number specified by the three-digit number nnn. Remember [GTO] [•] nnn always goes to the line number of the current program. If the calculator is not already within the boundary of the desired program, you can easily set it to that program by pressing [GTO] [•] and specifying the program name (e.g., ■ [GTO] [•] [ALPHA] AREA [ALPHA]).

Let's use [GTO] [•] nnn to set the calculator to line 015. We will insert a [PSE] instruction after that line to review the contents of the X-register (which is, at that time, the area of the cylinder without the top and bottom).

Keystrokes	Display	
■ [GTO] [•] 015	*15 **	Line 15 of AREA.
[XEQ]	*16 XEQ __*	
[ALPHA] PSE [ALPHA]	*16 PSE*	The [PSE] instruction.

When you added the [PSE] instruction, the program was altered ...

... from this ...	...to this.	
00	00	
01 LBL^{T}AREA	01 LBL^{T}AREA	
02 STO 01	02 STO 01	
03 X↑2	03 X↑2	
04 PI	04 PI	
05 *	05 *	
06 2	06 2	
07 *	07 *	
08 PSE	08 PSE	
09 X<>Y	09 X<>Y	
10 RCL 01	10 RCL 01	
11 *	11 *	
12 PI	12 PI	
13 *	13 *	
14 2	14 2	
15 *	15 *	
16 +	16 PSE	The PSE instruction was inserted here.
17 END	17 +	All subsequent instructions are moved down in program memory.
	18 END	

To go to a line in a very long program, that is, longer than 999 lines, you press EEX in place of the thousands digit. You then key in the three remaining digits of the line number. For example, to go to line 1,540 of an 1,800 line program, simply press GTO • EEX 540. This long-program addressing is only useful when your HP-41C has been enhanced with memory module extensions.

GTO • EEX 540 = go to line 1540

Specifying a line number for GTO • that is larger than the current program will simply set the calculator to the END of that program.

Running the Modified Program

To run the modified AREA program, you have only to take the calculator out of PRGM mode and, since the calculator is still in USER mode, simply press the LOG key (remember that you assigned AREA to the LOG key location for execution in USER mode).

Run the modified AREA program for values of 78″ (height) and 14″ (radius):

Keystrokes	Display	
[PRGM]	29,587.1684	Takes the HP-41C out of PRGM mode. The displayed number remains from the previous example.
78 [ENTER↑]	78.0000	The h value.
14	14_	The r value.
[AREA] ([LOG])	1,231.5043 6,861.2384 8,092.7427	After reviewing the X-register contents two times during the running program (first to display the area of the cylinder ends, and then to display the area of the cylinder without the ends), the answer in square inches is displayed.

Now run the program again for a height of 2.2789 meters and radius of 0.397 meter. (The final answer is 6.6748 square meters.)

Deleting and Correcting Instructions

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, simply press the nonrecordable function [←] *(correction)* with the calculator set to PRGM mode. (Refer to pages 42-43 to see how [←] works in normal mode.)

When you delete an instruction from program memory using [←], the calculator moves to the line *before* the deleted line and displays it.

For example, if you wanted to modify AREA again so that only the final answer is displayed, you would first delete the [PSE] instruction that is in line 8.

Keystrokes	Display	
[PRGM]	00 REG 38	Places the HP-41C into PRGM mode.
[■] [GTO] [•] 008	08 PSE	Sets the HP-41C to line 8, the location of the first [PSE] (pause).
[←]	07 *	Line 8 is deleted and the calculator moves up to line 7.

You can use [SST] to see that the [PSE] was deleted and all subsequent lines were moved up.

[SST]	08 X<>Y	The [x⇄y] was in 9 but was moved up to 8 when you deleted the [PSE].

When you set the HP-41C to line 8 and pressed [←] to delete the [PSE], the program was altered ...

... from this ..	... to this.
00	*00*
01 LBLᵀAREA	*01 LBLᵀAREA*
02 STO 01	*02 STO 01*
03 X↑2	*03 X↑2*
04 PI	*04 PI*
*05 **	*05 **
06 2	*06 2*
*07 **	*07 **
08 PSE	*08 X<>Y*
09 X<>Y	*09 RCL 01*
10 RCL 01	*10 **
*11 **	*11 PI*
12 PI	*12 **
*13 **	*13 2*
14 2	*14 **
*15 **	*15 PSE*
16 PSE	*16 +*
17 +	*17 END*
18 END	

One [PSE] deleted here.

These instructions all move up.

Now, to delete the [PSE] instruction that is at line 15:

Keystrokes	**Display**	
[GTO] [•] 015	*15 PSE*	
[←]	*14 **	The [PSE] is deleted from line 15 and the HP-41C displays line 14. Subsequent instructions move up.
[PRGM]	*8,092.7427*	Takes the HP-41C out of PRGM mode.

Run AREA in USER mode (press [LOG]) for two cylindrical water heaters with the following dimensions:

1.329 meters (h), 0.4811 meter (r).
(Answer: 5.4716 square meters.)

17.24 feet (h), 9 feet (r).
(Answer: 1,483.8370 square feet.)

In the HP-41C is another editing function that allows you to delete lines from your programs. This function is [DEL] *(delete lines)*. When you execute [DEL], the HP-41C prompts you for a three-digit line number like this: ***DEL _ _ _***. This three-digit number specifies a number of lines to delete from the current program (the program that the calculator is currently positioned to). The calculator deletes the specified number of lines beginning at the current position in the program. DEL operates only in PRGM mode.

So, if you have a 40-line program and you wish to delete 16 lines beginning with line 6, you would first set the calculator to line 6 of the program. Then you would execute DEL and specify 016 to delete 16 lines. With the calculator set to line 6 of our imaginary program, DEL 016 would change the program …

... from this ...	... to this.
00	00
01 LBL^{T}BEGIN	01 LBL^{T}BEGIN
02	02
03	03
04	04
05 RTN	05 RTN
06 LBL 01	06 LBL 02
07	07
:	:
21 RTN	24 END
22 LBL 02	
23	
:	
40 END	

With the calculator set to line 6, DEL 016 would delete 16 lines.

The DEL function will not delete lines beyond an END instruction. For example, if you execute DEL and specify 040 lines, and there are less than 40 lines in the program, the calculator will only delete up to but not including the END instruction.

With the calculator set to line 6 of our imaginary program, DEL 040 would change the program …

... from this ...	... to this.
00	00
01 LBL^{T}BEGIN	01 LBL^{T}BEGIN
02	02
03	03
04	04
05 RTN	05 RTN
06 LBL 02	06 END
07	
08	
09	
10	
:	
24 END	

DEL 040 would only delete up to but not including the END.

[DEL] never deletes more lines than you have in a program (providing the program is terminated with an [END]) and never deletes more lines than there are allocated to program memory.

Correcting Instructions

You can also use [←] to correct keystroke errors while you are keying in instructions of your programs. In fact, [←] works just the same in PRGM mode as it does when you are working problems and keying in numbers and ALPHAs in normal mode.

When you make an error while you are keying in a program instruction, simply press [←] in PRGM mode. Your last keystroke will be deleted.

For example, set the calculator back to line 14 and insert a [PSE] instruction (notice the keystroke error):

Keystrokes	Display	
[PRGM]	00 REG 38	Sets the HP-41C to PRGM mode.
[■] [GTO] [•] 014	14 *	
[XEQ]	15 XEQ __	
[ALPHA] PSF	15 XEQ PSF_	Whoops, this should be PSE, not PSF. When you make an error, simply press [←].
[←]	15 XEQ PS_	Now you can key in the correct letter, E.
	15 XEQ PSE_	
[ALPHA]	15 PSE	The [PSE] is now in line 15.
[PRGM]	1,483.8370	Result remains from example on page 137.

Run the program using a height of 56 inches and a radius of 12 inches.

Keystrokes	Display	
56 [ENTER↑]	56.0000	
12	12_	
[AREA] ([LOG])	4,222.3005	Intermediate answer.
	5,127.0792	
[■] [CLX]	0.0000	
[USER]	0.0000	

Using CATALOG for Positioning

CATALOG 1 lists all of the programs that you have recorded in program memory. In addition, as an aid to positioning the calculator to programs in program memory, as the listing of catalog 1 progresses, the calculator is set to the location in program memory of each program name as it is displayed. When the next program name is displayed, the calculator is positioned to that program in program memory.

CATALOG 1 only lists ALPHA program labels and END instructions.

For example, if the CIRCLE and AREA programs you have placed in program memory are intact, program memory looks like this ...

```
01 LBLTCIRCLE
02 X↑2
03 PI
04 *
05 END
01 LBLTAREA
02 STO 01
03 X↑2
04 PI
05 *
06 2
07 *
08 X<>Y
09 RCL 01
10 *
11 PI
12 *
13 2
14 *
15 PSE
16 +
17 END
```

...when you execute CATALOG 1, you will see the following:

```
LBLTCIRCLE
END
LBLTAREA
END
•END• REG 38
```

(This is the permanent END in program memory.)

By pressing R/S as the listing of CATALOG 1 was in progress, you could stop the listing and the calculator would be positioned to the label or END displayed. You could then press SST or BST to locate and position the calculator to the desired program in program memory.

The PACK Function

For your convenience during editing sessions, the HP-41C inserts extra blank lines in your programs. These blank lines are invisible to you; you cannot see them in program memory. They are placed in your programs to assure that while you are inserting and deleting instructions, the calculator responds to your commands as quickly as possible.

There are several ways that the HP-41C automatically removes these extra lines when you are through editing. This is called "packing." Following is a summary of the times that the HP-41C automatically packs program memory.

1. Any time you execute CLP *(clear program),* program memory is packed.
2. Any time you attempt to insert a line into a program when there is not enough room in program memory, program memory is packed. When the packing is complete, the calculator will display *TRY AGAIN* and you should reinput the desired line.
3. When you press ■ GTO • • program memory will be packed. If there is still not enough room in program memory to insert an END, the calculator will display *TRY AGAIN*. There is now not enough room in program memory for any more instructions and you should change the program memory allocation before you continue.
4. Any time you attempt to assign an HP-41C function to a key using ASN, and there is not enough room in program memory for the HP-41C to record the assignment, program memory will be packed. When the pack is complete, the HP-41C will display *TRY AGAIN* and you should again press the keys necessary to assign the function to a key.

You can cause program memory to be packed at any time by executing the PACK function. (PACK is not programmable.)

A typical pack will take a few seconds. During this time, the display will show *PACKING* The result of packing memory is that the programs will run faster after packing.

Problems

1. The following program calculates the time it takes 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 in the height h in meters into the X-register and the program is executed, the time t in seconds the object takes to fall to earth is computed according to the formula:

$$t = \sqrt{2h/9.8}$$

a. Press ■ [GTO] [•] [•] to set the calculator to the end of program memory and load the program.

```
00
01 LBLTFALL
02 2
03 *
04 9.8
05 /
06 SQRT
07 END
```

b. Run the program to compute the time taken by a stone falling from the top of the Eiffel Tower, 300.51 meters high; from a blimp stationed 1050 meters in the air.

(Answers: 7.8313 seconds; 14.6385 seconds.)

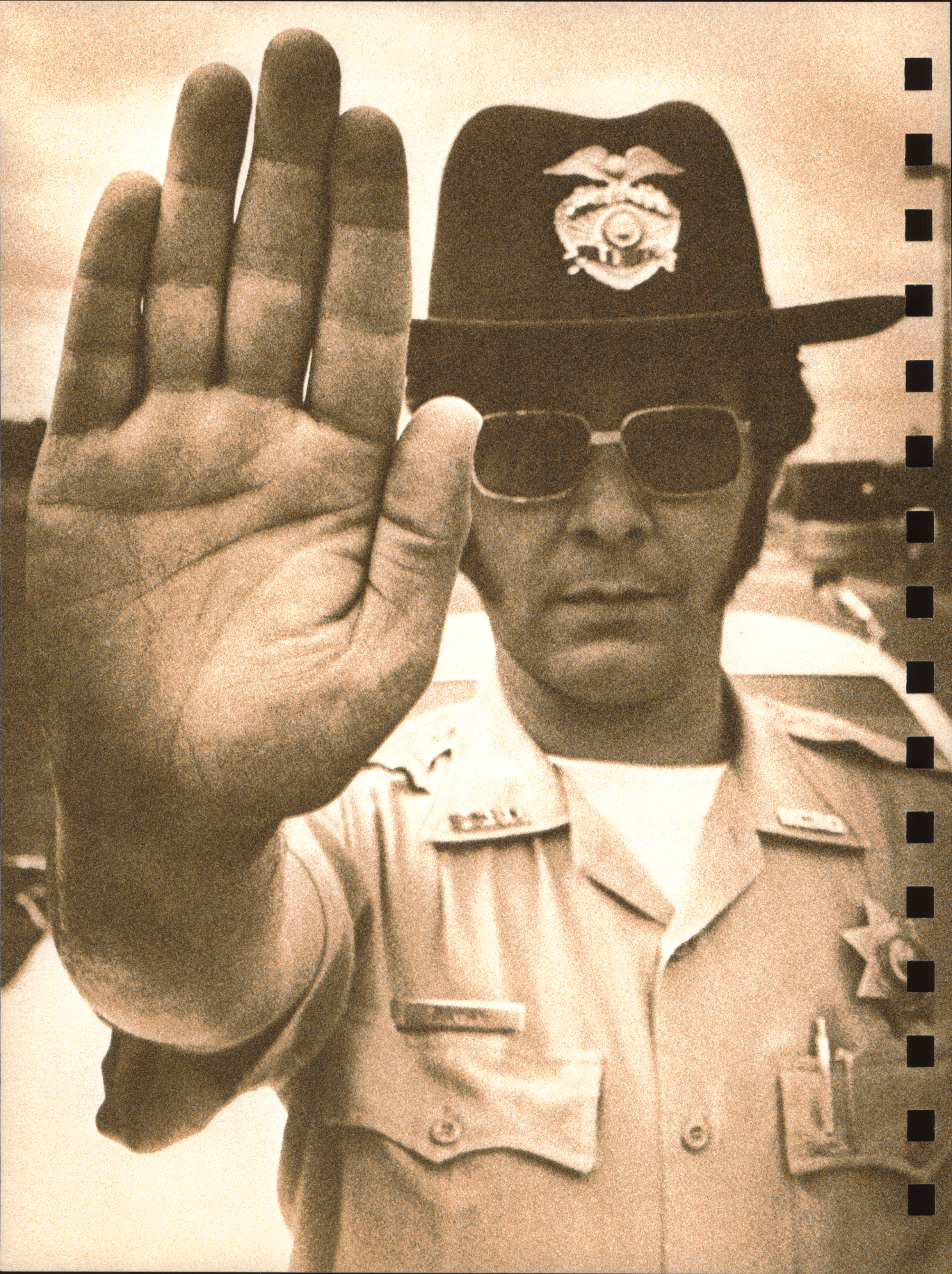
c. Alter the program to compute the time of descent when the height in feet is known, according to the formula:

$t = \sqrt{2h/32.174}$

d. Run the altered program to compute the time taken for a stone to fall from the top of the 550-foot high Grand Coulee Dam and a coin from the top of the 607-foot high Space Needle in Seattle, Washington.

(Answers: 5.8471 seconds; 6.1427 seconds.)

(This page intentionally left blank.)



Section 9

Program Interruptions

In your programs, there may often be occasions when you want to halt execution so that you can key in data, or to pause so that you can quickly view results before the program automatically resumes. This section shows you how to use [STOP] and [PSE] for program interruptions, as well as how the keyboard can be used to stop execution with [R/S], and how an error can halt a running program.

Using [STOP] and [R/S]

The [STOP] function can be placed into a program as an instruction by pressing the [R/S] *(run/stop)* key or by using [XEQ] and spelling the name (STOP). When executed in the program, the [STOP] stops program execution *after* its line of program memory.

The [R/S] function is only a keyboard function, that is, it cannot be recorded as an instruction in a program. However, when you press the [R/S] key in PRGM mode, a [STOP] instruction is recorded in the program. When you press the [R/S] key and the calculator is *not* in PRGM mode:

1. If a program is running, a [STOP] is executed and program execution is halted. The only keys that can halt a running program (from the keyboard) are the [ON] and [R/S] keys.
2. If a program is stopped or not running, [R/S] starts the program running beginning with the current line in the program.

When using [R/S] to halt a running program, remember that only the [R/S] key *location* in the lower right-hand position of the keyboard, performs the run/stop function. This is true even in USER mode, regardless of where [STOP] is assigned or which function is assigned to that location.

Example: The following program calculates the volume of a sphere given its radius. The program stops execution (with [STOP]) to let you key in the value of the radius of the sphere.

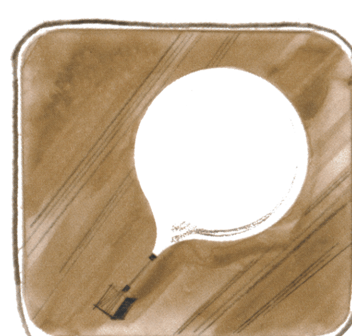
The formula for finding the volume of a sphere is $V=(4\pi r^3)\div 3$.

Keystrokes	Display	
PRGM		Sets the HP-41C to PRGM mode.
■ GTO • •	00 REG 40	Sets the HP-41C to the end of program memory.
■ LBL	01 LBL_ _	
ALPHA SPHERE ALPHA	01 LBLT SPHERE	The program name, SPHERE.
■ CLX	02 CLX	Clears the X-register.
R/S	03 STOP	Stops to key in the radius of the sphere.
3	04 3_	Places 3 into X. This value pushes the radius into the Y-register.
■ y^x	05 Y↑X	Computes r^3.
■ π	06 PI	The value of pi.
×	07 *	Multiplies r^3 by π.
4	08 4_	
×	09 *	Multiplies πr^3 by 4.
3	10 3_	
÷	11 /	Divides $4\pi r^3$ by 3.
■ GTO • •	00 REG 37	Ends the program.
PRGM	0.0000	

Now assign SPHERE to the $\sqrt{x}$ key location.

Keystrokes	Display
■ ASN	ASN _
ALPHA SPHERE ALPHA	ASN SPHERE _
$\sqrt{x}$	0.0000

Run SPHERE to find the volume of a spherical weather balloon with a radius of 21.22 feet. Run SPHERE again to find the volume of an official size ping pong ball with a radius of 1.905 centimeters.



Keystrokes	Display	
USER	0.0000	
SPHERE (√x)	0.0000	The program stops so you can key in the radius of the sphere.
21.22	21.22_	The radius of the spherical balloon.
R/S	40,024.3924	The answer in cubic feet.
SPHERE (√x)	0.0000	
1.905	1.905 _	The radius of the ping pong ball in centimeters.
R/S	28.9583	The volume of the ping pong ball in cubic centimeters.
USER ■ CLX	0.0000	

In the next section (section 10), you will see how ALPHA strings can be used to make prompting for data simple—your programs can actually *ask* you for data.

Using PSE *(Pause)*

The PSE *(pause)* instruction executed in a program momentarily interrupts program execution and displays the contents of the X-register. The length of the pause is slightly less than one second, although more PSE instructions in subsequent lines of a program can be used to lengthen viewing time, if desired.

Each time a pause is executed, the PRGM annunciator blinks one time. This lets you know that the program is running—even during a pause.

During program execution, the only keys that are active are R/S and ON . However, during the execution of a pause, or a string of pauses, the entire keyboard becomes active. You can actually input data to your program during a pause.

Pressing data entry keys during the execution of a pause causes the pause instruction to be executed again (or until you have completed the data entry). Data entry keys are: ALPHA , USER, ■ , • , 0 through 9, CHS , EEX , and all ALPHA characters.

Pressing any other keys during a pause, that is, any keys not associated with data entry, causes the pause to terminate and program execution halts. The pressed function *is* executed.

Keyboard Stops

As you know, pressing R/S from the keyboard during a running program halts that program. The program may halt after any line—if you set the calculator to PRGM mode after a program is halted, you will see the line number and the instruction of the next line to be executed.

When a program is halted, you can resume execution by pressing **R/S** from the keyboard in normal mode. When you press **R/S**, the program begins execution with the next line as though it had never stopped at all.

Error Stops

If the HP-41C attempts to execute any error-causing operation during a running program, execution halts and the HP-41C displays an error message. For example, if a program attempts division by zero, the calculator displays *DATA ERROR*. If the program calculates a number too large for the calculator to handle, the HP-41C displays *OUT OF RANGE*.

To see the line in the program containing the error-causing instruction, briefly set the calculator to PRGM mode. Setting the HP-41C to PRGM mode clears the error, as does pressing **←**. You can then make the necessary changes to ensure proper execution.

The HP-41C has several functions that allow you to control how the calculator reacts to these and other errors. Section 14 of this handbook covers these error conditions in detail.

Problem

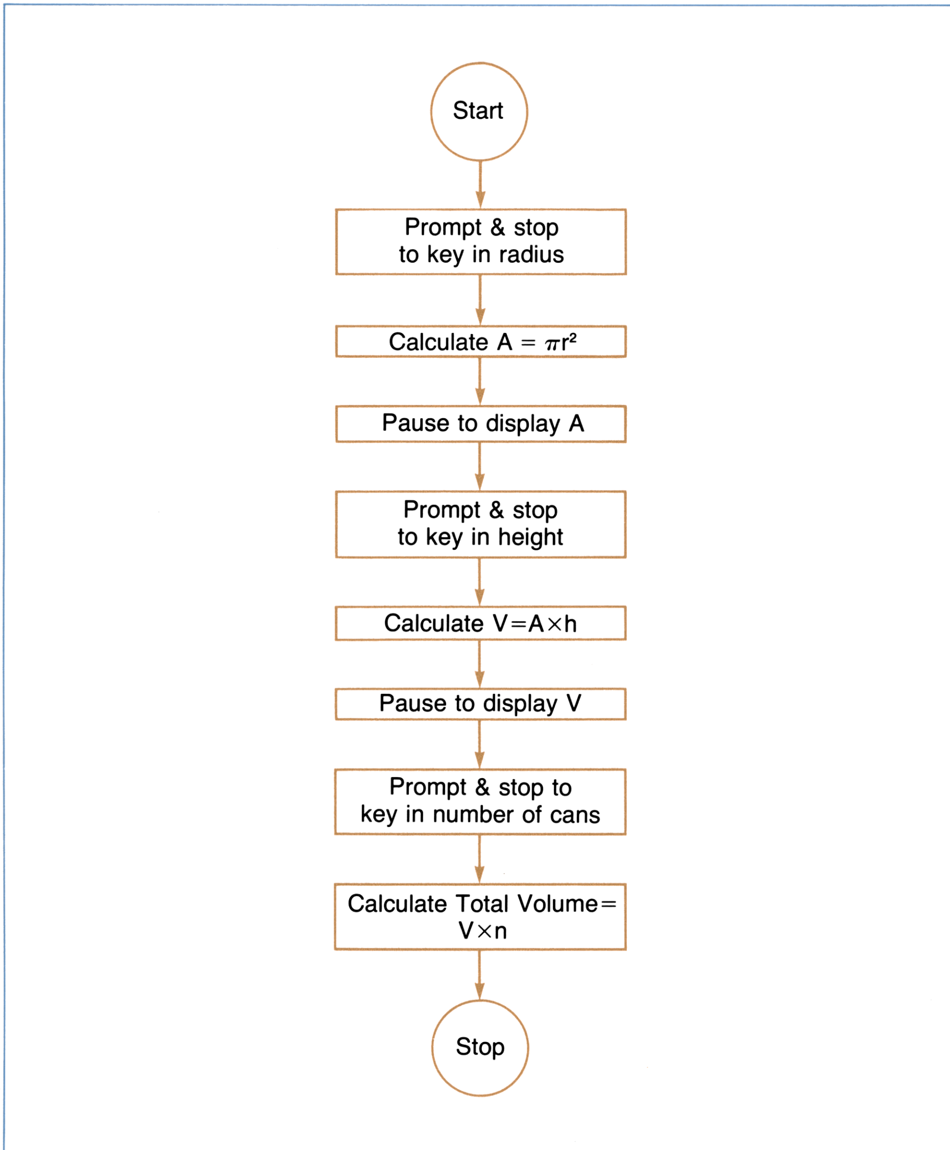
1. For several different sizes of cans, the supervisor at a canning company knows the radius r of the base of the can, the height h of the can, and n, the number of cans of that size. Write a program that will stop for the supervisor to key in the radius, the height, and the number of cans. The program should calculate the base area of one can, the volume of one can, and the total volume of all of the cans. Use **PSE** instructions to display the area and volume of the single cans before the total volume is displayed.

   Use the following flowchart to help you write and load the program. Assign the program to the **TAN** key location and 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 radii of 4.5 centimeters.

   (Answers:

   A = 314.1593 cm^2
   V = 7,853.9816 cm^3
   Total Volume = 157,079,632.7 cm^3

   A = 63.6173 cm^2
   V = 508.9380 cm^3
   Total Volume = 3,817,035.074 cm^3.)


Start
Prompt & stop to key in radius
Calculate $A = \pi r^2$
Pause to display A
Prompt & stop to key in height
Calculate $V=A\times h$
Pause to display V
Prompt & stop to key in number of cans
Calculate Total Volume= $V\times n$
Stop

Section 10

Programming with ALPHA Strings

One of the greatest utilizations of the HP-41C ALPHA capability is in programs that you write. ALPHA strings (a series of ALPHA characters) in your programs can prompt you for information, inform you of the status of a program and even label output. This section shows you how to use ALPHA strings in your programs.

Using ALPHA Strings in Your Programs

You can use ALPHA strings in many different ways in your programs, and there are certain ways that these strings change what you see in the display while a program is running.

For example, you can place an ALPHA string in a program and instruct the program to display that string with AVIEW. The ALPHA string that you input as a line in the program is placed into the ALPHA register. AVIEW then places the contents of the ALPHA register into the display. As program execution progresses, the display continues to display the string until the program clears the string from the display, or you place a new string into the display.

Any time a program places an ALPHA string into the display, that string replaces the ⌇- program execution symbol. When the program clears the display or the program is interrupted, the ⌇- returns to the display. Regardless of what is displayed, the **PRGM** annunciator is always displayed during a running program.

The maximum length of an ALPHA string on any one line in a program is 15 characters. However, using APPEND (■ K in ALPHA mode), you can construct strings of up to 24 characters. Key in the first 15 characters in the string, press APPEND and then key in the remainder of the characters. The first 15 characters will be on one line in the program, and the remainder of the characters will be on the following line. Refer to section 3 in part I for more information about APPEND.

Prompting

There are several ways to use ALPHA strings in your programs to prompt for data input. Prompts in your programs are a simple way to assure that you input the correct data value. Or you can use prompts to simply display messages.

The easiest way to use prompts is with the PROMPT function. The PROMPT instruction in a program displays the contents of the ALPHA register *and* stops program execution. Simply key in the ALPHA string as a line in the program and follow it with PROMPT. Execution will halt and the display will show the prompt string.

Another way to use prompt is to use ARCL to recall a string from a register and then use PROMPT to halt program execution and display the prompt string. This method requires you to store the ALPHA string into a register for later use as a prompt string. You can either store this string before you execute the program or you can instruct the program to store the string. Refer to section 5 of part I for more information about ARCL.

Example: The following program prompts for a number, stops for the input, then computes the common logarithm of the number. The ALPHA prompt is a line in the program and is placed into the display with PROMPT.

Keystrokes	Display	
PRGM		
■ GTO • •	00 REG 37	
■ LBL		
ALPHA CLOG ALPHA	01 LBLT CLOG	The program name, CLOG.
ALPHA NUMBER? ALPHA	02 T NUMBER?	The prompt string.
XEQ		
ALPHA PROMPT ALPHA	03 PROMPT	Displays the ALPHA register and stops for data input.
LOG	04 LOG	The common logarithm.
■ GTO • •	00 REG 34	

Find the log of 8 to see how the program works:

Keystrokes	Display	
PRGM	0.0000	Takes the HP-41C out of PRGM mode.
XEQ		
ALPHA CLOG ALPHA	NUMBER?	The prompt.
8	8_	The number.
R/S	0.9031	The log of 8.
■ CLX	0.0000	

Prompting can also be accomplished using AVIEW *(ALPHA view)* and STOP in a program. The AVIEW displays the contents of the ALPHA register and the STOP halts program execution.

Labeling Data

Data labeling can be quite useful to the output your programs produce. Labeled output leaves no doubt as to which result is displayed. Data can be labeled with ALPHA strings using ASTO, ARCL and AVIEW. To label output:

1. Key in the ALPHA string as a line in the program.
2. Recall the result to be labeled into the display with [ARCL]. Since [ARCL] adds to whatever is already in the ALPHA register, you may wish to clear the ALPHA register before you use [ARCL].
3. Then use [AVIEW] in the program to place contents of the ALPHA register into the display.

Note: Care must be used in labeling data in programs because information requiring more space than is available in the display will be scrolled off the display to the left.

Example: The following is a modification of CLOG (from above) that labels the output from the program. Begin by clearing CLOG from program memory and create a new version of the program.

Keystrokes	Display	
[XEQ]	*XEQ _ _*	
[ALPHA] CLP [ALPHA]	*CLP _*	
[ALPHA] CLOG [ALPHA]	*0.0000*	Clears CLOG from program memory.
[PRGM]		
■ [GTO] [•] [•]	*00 REG 37*	
■ [LBL]		
[ALPHA] LOG ■ 1 [ALPHA]	*01 LBL^{T}LOG1*	The new program name.
[ALPHA] NUMBER? [ALPHA]	*02^{T}NUMBER?*	The prompt for input.
[XEQ]		
[ALPHA] PROMPT [ALPHA]	*03 PROMPT*	Displays the prompt and stops for the data input.
[LOG]	*04 LOG*	The common logarithm.
[ALPHA] LOG=	*05^{T}LOG=_*	The data label.
■ [ARCL] [•] X	*06 ARCL X*	This recalls the result from the X-register and places it into the *ALPHA* register along with its current contents, *LOG=*.
■ [AVIEW]	*07 AVIEW*	This displays the contents of the ALPHA register (which is now *LOG=* and the logarithm result).
[ALPHA]		
■ [GTO] [•] [•]	*00 REG 33*	

Now run the LOG1 program to find the log of 12. Notice how the program first prompts you for the number, then labels the output.

Keystrokes	Display	
PRGM	0.0000	Takes the HP-41C out of PRGM mode. Number remains from previous example.
XEQ	XEQ _ _	
ALPHA LOG ■ 1 ALPHA	NUMBER?	The prompt for the number.
12	12	The number.
R/S	LOG=1.0792	The data label and the data.
■ CLX	0.0000	

Data labeling can also be accomplished by recalling (using ARCL) the ALPHA string from a register, and then the result from the X-register (also using ARCL).

Program Status

To detect the status of your executing program, you can place ALPHA strings in strategic places in your programs. When the string is displayed momentarily, you know exactly how far execution has progressed.

Prompting for ALPHA Strings

You can prompt for the *input* of ALPHA information just like you would for numbers. Using the AON *(ALPHA on)* and AOFF *(ALPHA off)* functions, you can even control the mode the calculator is set to when the program stops for input.

AON places the HP-41C into ALPHA mode and AOFF takes the calculator out of ALPHA mode.

Clearing the Display

To clear the contents of the *display* at any time during a running program, simply key in CLD *(clear display)* as a line in the program. This clears the display and then displays the X-register, or the ALPHA register (if the calculator is in ALPHA mode).

Using ASHF *(ALPHA Shift)*

ASHF is a handy HP-41C function that shifts the contents of the ALPHA register to the left by six characters. Manually or in a program, when you wish to store a long ALPHA string into several storage registers, ASHF makes the task simple. (Remember, each data storage register can hold up to six ALPHA characters.) When ASHF is executed, the left-most six characters in the ALPHA register are shifted off to the left and are lost. The remaining characters in the ALPHA register all shift to the left by six positions.

Here is an example of how [ASHF] can be used. The program stores a string of characters into several registers and then recalls the stored strings one at a time into the display. Begin by assigning [ASHF] to the [TAN] key for use in USER mode.

Keystrokes	Display
■ [ASN]	ASN _
[ALPHA] ASHF [ALPHA]	ASN ASHF _
[TAN]	ASN ASHF 25
	0.0000
[USER]	0.0000

Now load the program.

Keystrokes	Display	
[PRGM]		
■ [GTO] [•] [•]	00 REG 32	
■ [LBL]		
[ALPHA] SHIFTY [ALPHA]	01 LBL^TSHIFTY	
[ALPHA] SUNDAYMONDAY	UNDAYMONDAY _	
■ [ASTO] 01	03 ASTO 01	The first six characters are stored into R_{01}.
[ALPHA]		
[ASHF] ([TAN])	04 ASHF	Six characters are shifted off to the left.
[ALPHA]		
■ [ASTO] 02	05 ASTO 02	The second six characters are stored into R_{02}.
■ [CLA]	06 CLA	ALPHA register is cleared.
■ [ARCL] 01	07 ARCL 01	Recall the six characters stored into R_{01}.
■ [AVIEW]	08 AVIEW	Display the string.
[ALPHA]		
[XEQ]		
[ALPHA] PSE [ALPHA]	09 PSE	Pause.
[ALPHA] ■ [CLA]	10 CLA	ALPHA register is cleared.
■ [ARCL] 02	11 ARCL 02	Recall the six characters stored into R_{02}.
■ [AVIEW]	12 AVIEW	Display the string.
[ALPHA]		
■ [GTO] [•] [•]	00 REG 26	The end of the program.

Run the program and watch how the strings are displayed.

Keystrokes	Display
USER PRGM	0.0000
XEQ	
ALPHA SHIFTY ALPHA	SUNDAY
	MONDAY

Problem:

1. The following program computes the total price, tax, and final cost of items on a billing invoice. Rewrite the program and insert ALPHA strings and PROMPT for the quantity, unit price and tax. In addition, insert an ALPHA string to label the output of the final amount (recall the final amount from the X-register into the ALPHA register using ARCL · X. Run the program for 26 ruby rings that cost $72.90 with a tax of 7.25%; for 11 shovels that cost $7.15 with a 5% tax.

Insert these strings into the program to prompt for data: ***QUANT?*** (quantity), ***PRICE?*** (unit price), ***TAX?*** (tax rate). Store this string in storage register R_{10} (with ASTO) and recall it (with ARCL) in the program to label the output: ***TOT=$***. If you have trouble with this problem, you may wish to review this section before you continue.

(Answers: ***TOT=$***2032.82; ***TOT=$***82.58.)

01 LBL⊤BILL1	Program name.
02 STOP	Stops for input of quantity.
03 STOP	Stops for input of unit price.
04 *	Computes total price.
05 STOP	Stops for input of tax rate.
06 %	Computes tax amount.
07 +	Computes final amount.
08 END	

(This page intentionally left blank.)

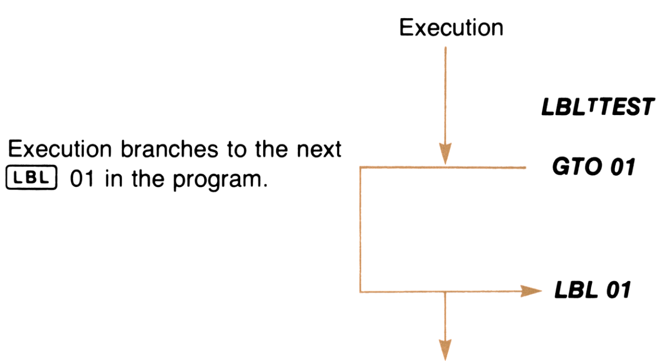
Branching and Looping

Branching and Looping

Earlier in this handbook you learned how you can use [GTO] [•] and a program line number or ALPHA label to position the calculator to a particular place in program memory. In addition, you saw how [GTO] [•] [•] positioned the calculator to the end of program memory to prepare the calculator for a new program. You can also use [GTO] *(go to label)* in your programs followed by an ALPHA or numeric label to transfer execution to any part of a program you desire.

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

Here is what a [GTO] branch would do if a program in the HP-41C looked like this:



When the program encounters the [GTO] 01 instruction, execution immediately halts and the calculator searches sequentially downward through the program for the first occurrence of a [LBL] 01. If the calculator does not find a [LBL] 01 before reaching the end of the program (an [END] instruction), the calculator starts searching from the top of the program until it finds the [LBL] 01. If the label does not exist, the HP-41C will display ***NONEXISTENT*** and the calculator will be positioned to the same line it was set to prior to beginning the search. Press [←] to clear the error.

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 HP-41C overflows).

You may wish to clear some of the programs you have recorded in program memory so that you will have room to include the problems in this and following sections. Check [CATALOG] 1 to see the names of the programs and delete the ones you don't wish to save using [CLP] *(clear program)*. You can clear any key assignments by pressing [ASN] [ALPHA] [ALPHA] and the reassigned key. Subsequent problems in the handbook assume that program memory has been cleared of all programs and no key assignments have been made.

Name the program ROOT and assign it to the [TAN] key location.

Keystrokes	Display	
[PRGM] [GTO] [•] [•]	00 REG 46	Sets the HP-41C to program mode and to the end of program memory.
[LBL]		
[ALPHA] ROOT [ALPHA]	01 LBL^{T}ROOT	The program name.
0	02 0_	
[STO] 01	03 STO 01	Stores 0 in R_{01}.
[LBL] 05	04 LBL 05	
1	05 1_	
[STO] [+] 01	06 ST+ 01	Adds 1 to the current number in R_{01}.
[RCL] 01	07 RCL 01	Recalls current number from R_{01}.
[XEQ]		
[ALPHA] PSE [ALPHA]	08 PSE	Displays current number.
[√x]	09 SQRT	Computes the square root of the number.
[XEQ]		
[ALPHA] PSE [ALPHA]	10 PSE	Displays square root of current number.
[GTO] 05	11 GTO 05	Transfers execution to the [LBL] 05 in line 4.
[GTO] [•] [•]	00 REG 43	

To run the program, first assign it to the [TAN] key location for execution in USER mode.

Keystrokes	Display	
[PRGM]	0.0000	
[ ] [ASN]	ASN _	
[ALPHA] ROOT [ALPHA]	ASN ROOT _	
[TAN]	0.0000	ROOT is assigned to the [TAN] location.
[USER]	0.0000	HP-41C is placed into USER mode.

Now, run the program:

Keystrokes	Display	
[ROOT] ([TAN])	1.0000 1.0000 2.0000 1.4142 3.0000 1.7321 4.0000 2.0000 5.0000	The program displays a table of integers and their square roots and continues until you press [R/S] from the keyboard or the calculator overflows.
[R/S]	2.2361	

How the program works: When you press [ROOT], the calculator begins executing the ROOT program starting with line 1. It executes that instruction and each subsequent instruction in order until it reaches the [GTO] 05 in line 11.

The [GTO] 05 in line 11 causes the HP-41C to begin a label search. It searches downward through the program to the [END] instruction, then starts at the beginning of the program (line 0) and searches downward until it finds the [LBL] 05 in line 4. *Notice that the address after the* [GTO] *instruction is a numeric program label, not a line number.*

Execution is transferred to the [LBL] 05 instruction in line 4 each time the calculator executes the [GTO] 05 in line 11. The calculator remains in this "loop," continually adding one to the number in storage register R_{01} and displaying the new number and its square root.

An exciting feature in the HP-41C is the calculator's ability to "remember" where most branches are located in a program. The HP-41C only has to search for most labels the first time through the program. When the program branches to that label the calculator does not have to search again! It knows where the label is located so it immediately begins execution at that line. The result is that execution time is *greatly* reduced because the calculator does not have to repeatedly search for most labels. This feature is known as compiling and is generally only found in large computer systems. For more information about how the HP-41C remembers labels, refer to appendix G.

The "infinite" loop.

```
00
01 LBLTROOT
02 0
03 STO 01
04 LBL 05
05 1
06 ST+ 01
07 RCL 01
08 PSE
09 SQRT
10 PSE
11 GTO 05
12 END
```

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 through the program and begins execution at the first specified label it encounters.

Problems

1. The following program computes $x = 2n \sin(90 \div n)$. Modify this program by placing a **LBL** 01 instruction in line 4, and these instructions at the end of the program (just before the **END**):

```
PSE
10
ST*00
GTO 01
```

The modification creates an infinite loop in the program; it now computes an infinite series of numbers that approaches the value of pi. Run the program and watch the values as they approach π. Set the calculator to FIX 9 so you can see the complete display.

```
00
01 LBLTPIFIND
02 1
03 STO 00
04 90
05 RCL 00
06 /
07 SIN
08 RCL 00
09 *
10 2
11 *
12 END
```

Insert a LBL 01 after this instruction.

Insert these instructions at the end of the program:

```
PSE
10
ST*00
GTO 01
```

Controlled Looping

The HP-41C has two powerful functions that make looping in your programs very easy. These functions are ISG *(increment and skip if greater)* and DSE *(decrement and skip if equal)*. Both functions contain internal counters that allow you to control the execution of the loop.

These two functions use a number that is interpreted in a special way to control program loops. The number is stored into any storage register (even the stack). The format of the number is:

iiiii.fffcc

where

iiiii is the current counter value,

fff is the counter test value, and

cc is the increment value.

The **iiiii** portion of the number tells the HP-41C that you wish to count the number of passes through the loop beginning with that number. If you do not specify an **iiiii** value, the HP-41C assumes you wish to begin counting at zero. An **iiiii** value can be specified as one to five digits.

The **fff** portion of the number tells the HP-41C that you wish to stop the counting at that number. The **fff** value must always be specified as a three-digit number (e.g., an **fff** value of 10 would be specified as 010). If you do not specify an **fff** value, the HP-41C assumes you wish to stop counting at zero.

The **cc** portion of the number tells the calculator how you wish to count. Current counter value **iiiii** is incremented or decremented by the increment value of **cc**. If you do not specify a **cc** value, the HP-41C assumes you wish to count by ones (**cc**=01). A **cc** value must be specified as two digits (e.g., 01, 03, 55).

Increment and Skip if Greater

Each time ISG is executed, it first increments **iiiii** by **cc**. It then tests to see if **iiiii** is greater than **fff**. If it is, then the HP-41C skips the next line in the program.

So, if you stored the number 100.20001 in storage register R_{10}, the ISG 10 instruction would begin counting at 100, would count up until the counter was greater than 200, and it would increment by 1 each time the loop was executed.

Contents of storage register R_{10} = 100.20001

Execution of ISG 10 would:

Start counting at 100.
Increment by 1.
Test to see if counter is greater than 200.

After one execution or pass through the loop, R_{10} would be 101.20001. After 10 executions or passes through the loop, R_{10} would be 110.20001. Each time ISG 10 is executed it checks to see if the counter is greater than 200. When it is greater than 200, it skips the next line of the program. You will see how skipping the next line in the program is useful in a moment.

If you execute ISG from the keyboard, it simply increments the specified register just like it would in a program, but no program lines are executed or skipped.

Decrement and Skip If Equal

Each time DSE is executed, it first decrements **iiiii** by **cc**. It then tests to see if **iiiii** is equal to (or less than) **fff**. If it is, then the HP-41C skips the next line in program memory.

So, if you stored the number 100.01001 in storage register R_{11}, the DSE instruction would begin counting down at 100, would count down until the counter was equal to (or less than) 10, and it would decrement by 1 each time the loop was executed.

Contents of storage register R_{11} = 100.01001

Execution of DSE 11 would:

Start at 100.
Decrement by 1.
Test to see if counter was equal to (or less than) 10.

Remember, in a program when the final value is obtained, the HP-41C skips the next line in the program. You will see how this is useful later.

If you execute **DSE** from the keyboard, it simply decrements the specified register just like it would in a program.

Example: Here is a program that illustrates how **ISG** works. It contains a loop that pauses to display the current value in register R_{01}, displays the square of that number, and uses **ISG** to control the number of passes through the loop and the value of the squared number. The program generates a table of squares of even numbers from 2 through 50.

Keystrokes	Display	
PRGM		
■ GTO • •	00 REG 46	
■ LBL		
ALPHA EVENS ALPHA	01 LBLTEVENS	The program name, EVENS.
2.05002	02 2.05002_	The loop control number. Beginning with 2, increments up to 50 by twos. Tests each execution to see if the counter is greater than 50.
STO 01	03 STO 01	Stores the loop control number in R_{01}.
■ LBL 01	04 LBL 01	Begins the loop.
RCL 01	05 RCL 01	Recalls the number in R_{01}.
XEQ		
ALPHA INT ALPHA	06 INT	Takes the integer portion of the number.
XEQ		
ALPHA PSE ALPHA	07 PSE	Displays the integer portion of the number.
■ x^2	08 X↑2	Squares the number.
XEQ		
ALPHA PSE ALPHA	09 PSE	Displays the square of the number.
■ ISG 01	10 ISG 01	Increments R_{01} by 2 and checks to see that the counter is not greater than the final number (50). If the counter is not greater than the final number, executes the next line. If the counter *is* greater than the final number, skips the next line in the program.
■ GTO 01	11 GTO 01	Loops back to LBL 01.
■ GTO • •	00 REG 42	

Now run the program:

Keystrokes	Display	
PRGM	0.0000	Takes the HP-41C out of PRGM mode.
XEQ		
ALPHA EVENS ALPHA	2.0000 4.0000 4.0000 16.0000 . . . 50.0000 2,500.0000	When the HP-41C begins executing the program, it first pauses to display the number, then pauses to display its square. When the loop counter increments beyond 50, the program stops.

Example: The island of Manhattan was sold in the year 1624 for $24.00. The program below shows how the amount would have grown each year if the original amount had been placed in a bank account drawing 6% interest compounded annually. The program prompts for the number of years and alters that number for use by DSE. The DSE is used to control the number of iterations through the loop.

Keystrokes	Display	
PRGM		
■ GTO • •	00 REG 46	
■ LBL		
ALPHA GOTHAM ALPHA	01 LBLTGOTHAM	The program name.
ALPHA YEARS? ALPHA	02TYEARS?	The ALPHA prompt.
XEQ		
ALPHA PROMPT ALPHA	03 PROMPT	Displays prompt and stops for input.
STO 00	04 STO 00	
1624	05 1624_	
STO 01	06 STO 01	
24	07 24_	
STO 02	08 STO 02	
■ LBL 01	09 LBL 01	The beginning of the loop.
RCL 02	10 RCL 02	
6	11 6_	
■ %	12 %	
STO + 02	13 ST+ 02	

Keystrokes:	Display:	
1	14 1 _	
STO + 01	15 ST+ 01	
XEQ		
ALPHA DSE ALPHA 00	16 DSE 00	The loop control number is stored in R_{00}. The counter test value (**fff**) is zero and the decrement value (**cc**) is 01. When **iiiii** reaches zero, the next line in the program is skipped. Until then, the program loops back to LBL 01.
GTO 01	17 GTO 01	The end of the loop.
RCL 01	18 RCL 01	Recalls the year.
FIX 0	19 FIX 0	
XEQ		
ALPHA PSE ALPHA	20 PSE	Pauses to display the year.
FIX 2	21 FIX 2	
RCL 02	22 RCL 02	Recalls the final amount.
GTO • •	00 REG 39	

Now run the program to find the amount in the savings account after 6 years; after 355 years. (This will take a couple of minutes to run, time enough to take a short break.)

Keystrokes	Display	
PRGM	0.0000	Takes the HP-41C out of PRGM mode.
XEQ		
ALPHA GOTHAM ALPHA	YEARS?	The program prompts and stops for input.
6 R/S	1,630 34.04	After 6 years, in 1630, the account would have been worth $34.04.
XEQ		
ALPHA GOTHAM ALPHA	YEARS?	
355 R/S	1,979 2.31 10	After 355 years, in 1979, the account would be worth about $23.1 billion.
CLX	0.00	
FIX 4	0.0000	Returns to FIX 4.

How it works: Each time you execute GOTHAM, the program prompts you for the number of years, which is stored in R_{00}. This is used by the DSE as the loop control value. The year (1624) is stored in R_{01} and the initial amount is stored in R_{02}.

Each time through the loop, 6% of the amount is computed and added to the amount in R_{02} and 1 year is added to the year in R_{01}. The DSE subtracts one from the R_{00}-register; if the value in R_{00} is not then zero, execution is transferred back to LBL 01, and the loop is executed again.

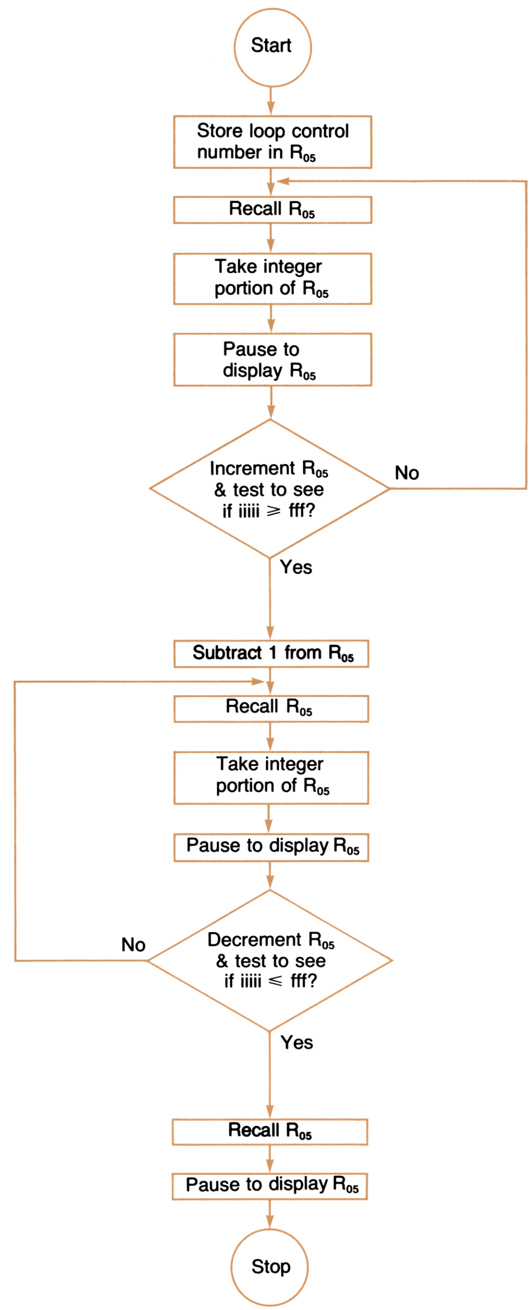
When R_{00} becomes zero, execution skips to the RCL 01 instruction in line 18. The year is then recalled and displayed (formatted in FIX 0), and the final amount is recalled and displayed (formatted in FIX 2).

Note that ISG and DSE can be used to increment and decrement *any* number that the HP-41C can display. However, the decimal portion of the control number will be affected by large numbers.

For example, the number 99,950.50055, when incremented by 55 using ISG would become 100,005.5005. The initial number was incremented by 55. But since the new number cannot be fully displayed, the decimal portion of the number was truncated. The next increment would be by 50, not 55. And when the number becomes 999,955.5005, the next number would be 1,000,005.500, thus truncating the decimal portion of the number again. Since no increment value is present, the next increment would be by 01, not 50.

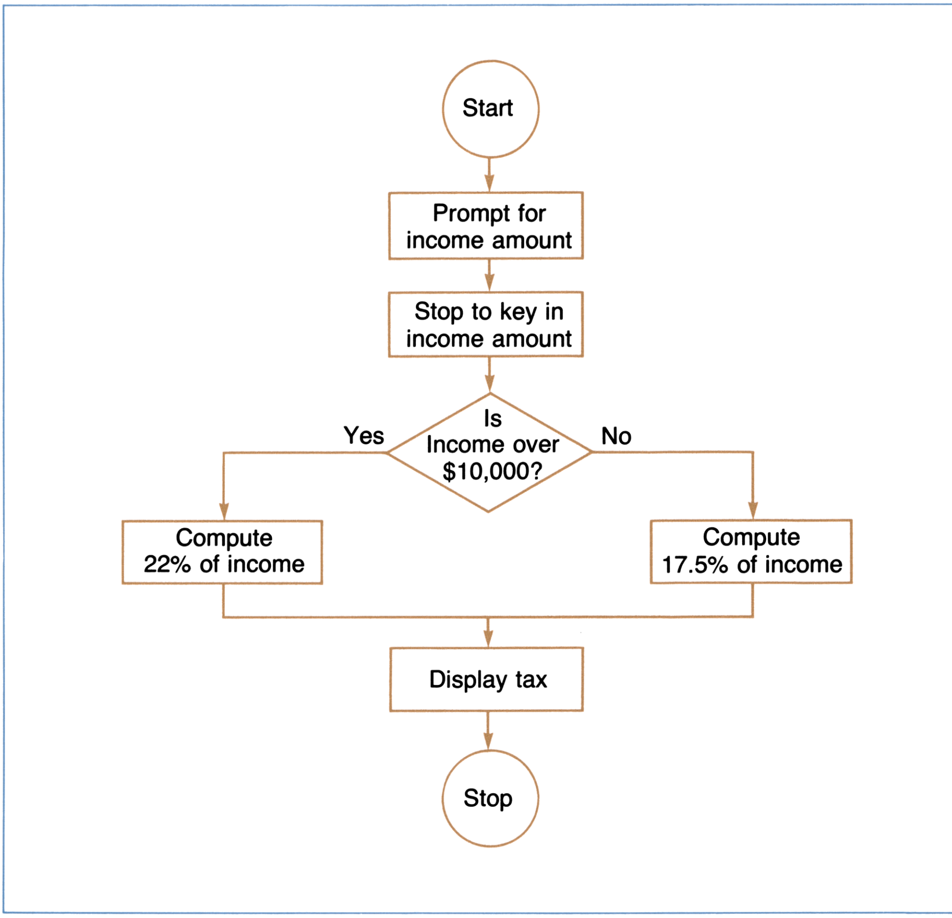
Problem:

1. Write a program that will count from zero up to a limit using the ISG function, and then, in the same program, count back down to zero using the DSE function. The program should contain two loops, the first one counting up, the second one counting down. Use the flowchart on the following page to help you.


Start
Store loop control number in R_{05}
Recall R_{05}
Take integer portion of R_{05}
Pause to display R_{05}
Increment R_{05} & test to see if iiiii $\geqslant$ fff?
No
Yes
Subtract 1 from R_{05}
Recall R_{05}
Take integer portion of R_{05}
Pause to display R_{05}
Decrement R_{05} & test to see if iiiii $\leqslant$ fff?
No
Yes
Recall R_{05}
Pause to display R_{05}
Stop

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 22%. A flowchart for the program might look like this:



The conditional operations on your HP-41C are useful as program instructions to allow your calculator to make decisions like the ones shown above. The ten conditionals available in the HP-41C are shown below.

x=y?	tests to see if the value in the X-register is equal to the value in the Y-register.
x=0?	tests to see if the value in the X-register is equal to zero.

X>Y? tests to see if the value in the X-register is greater than the value in the Y-register.

X>0? tests to see if the value in the X-register is greater than zero.

X<Y? tests to see if the value in the X-register is less than the value in the Y-register.

X<0? tests to see if the value in the X-register is less than zero.

X≤Y? tests to see if the value in the X-register is less than or equal to the value in the Y-register. Display execution form is X<=Y? .

X<=0? tests to see if the value in the X-register is less than or equal to zero.

X≠Y? tests to see if the value in the X-register is not equal to the value in the Y-register.

X≠0? tests to see if the value in the X-register is not equal to zero.

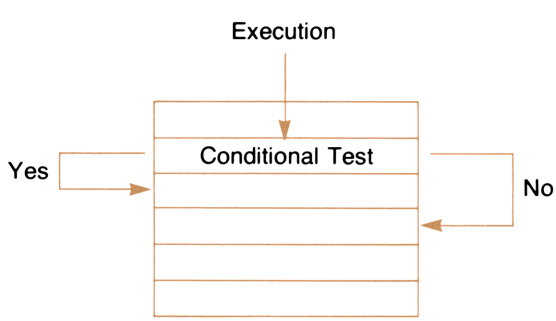
Two of these conditionals, X=Y? and X≠Y? can be used to compare ALPHA strings as well as numbers. All of the other conditionals compare only numbers. If two strings are "equal" (X=Y?), then they are *exactly* equal in length and have identical characters.

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.

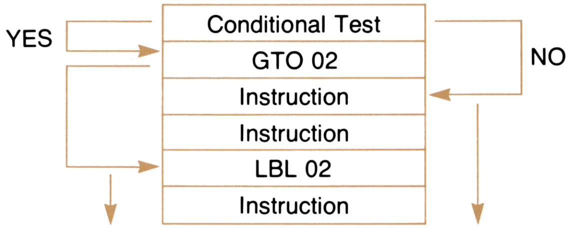
(When you execute any of these conditionals manually from the keyboard, the HP-41C displays the answer to the conditional question. If the condition is true the display shows *YES*. If the conditional is false, the display shows *NO*.)

In other words, the calculator will *do* the next line if the test is *true*. This is the "DO IF TRUE" rule.

For example:



The line immediately following the conditional test can contain any instruction. The most commonly used instruction will be a GTO instruction. This will branch program execution to another section of program memory if the conditional test is true. For example:



Now let's look at that tax accountant's problem again. For persons with incomes of more than $10,000 the program should compute a tax of 22%. For persons with income 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 tax amount.

Keystrokes	Display	
PRGM		
GTO • •	00 REG 46	
LBL		
ALPHA TAX ALPHA	01 LBL^{T}TAX	The program name.
ALPHA INCOME? ALPHA	02^{T}INCOME?	Prompts for income.
XEQ		
ALPHA PROMPT ALPHA	03 PROMPT	Displays prompt and halts execution so you can key in the income.
10000	04 10000_	
x≷y	05 X<>Y	Amount of $10,000 put in Y-register.
X>Y?	06 X>Y?	Conditional test. If income is greater than $10,000, does the next line in the program. If not, skips the next line.
GTO 02	07 GTO 02	Branch to LBL 02.
17.5	08 17.5_	Tax rate (income less than $10,000).
GTO 03	09 GTO 03	Branch to LBL 03.
LBL 02	10 LBL 02	
22	11 22_	Tax rate (incomes more than $10,000).
LBL 03	12 LBL 03	
%	13 %	Computes the tax.
GTO • •	00 REG 41	

To run TAX to compute taxes on incomes of \$38,000 and \$7,600:

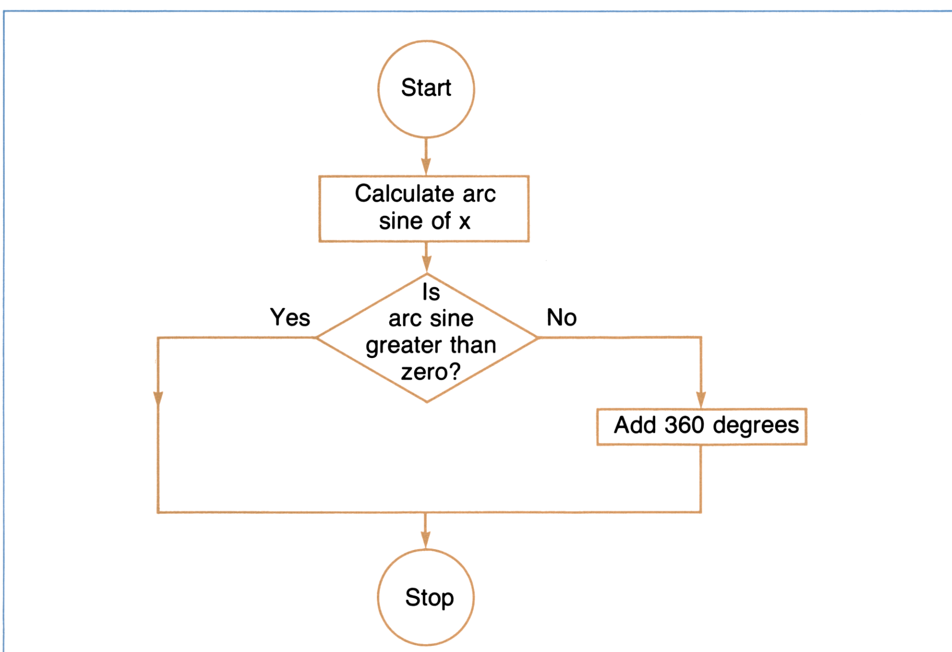
Keystrokes	Display	
PRGM	0.0000	Takes the HP-41C out of PRGM mode.
XEQ		
ALPHA TAX ALPHA	INCOME?	The prompt for the income.
38000	38,000_	
R/S	8,360.0000	The tax at 22%.
XEQ		
ALPHA TAX ALPHA	INCOME?	
7600	7,600_	
R/S	1,330.0000	The tax at 17.5%.
■ CLX	0.0000	

Problems:

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

   Run the program to find the arc sine of −0.7 of 0.5.

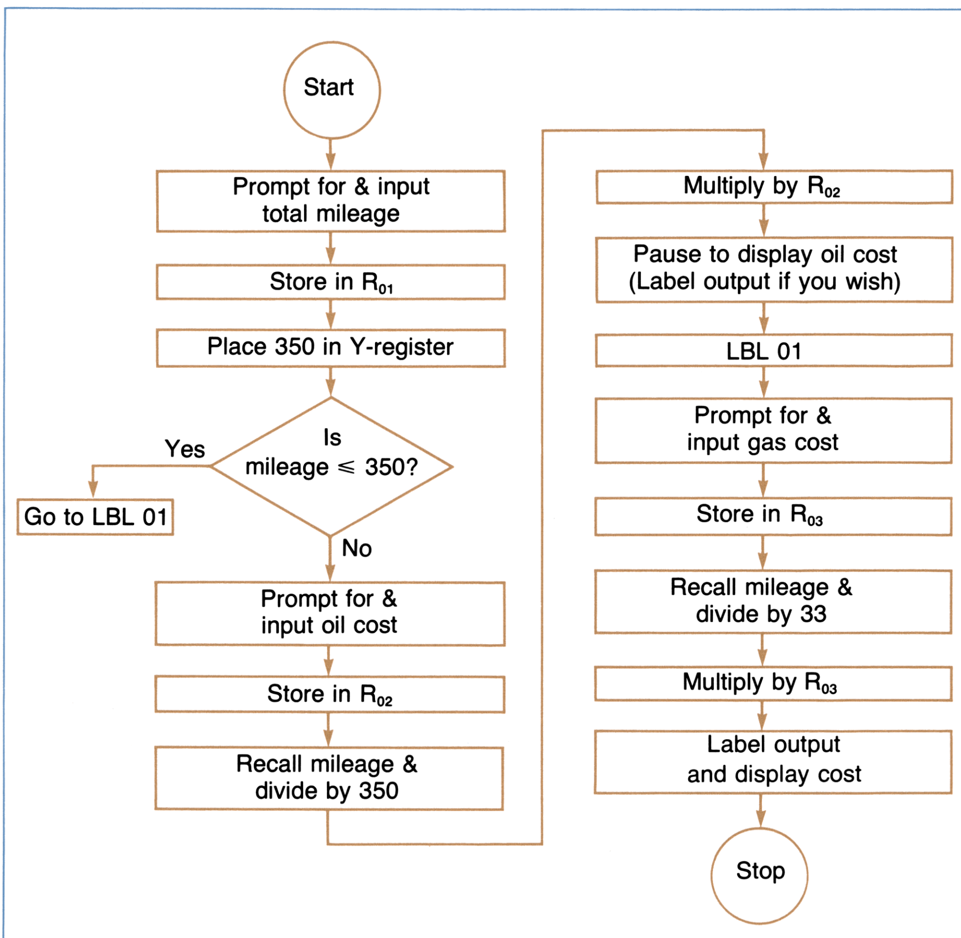
   (Answers: 315.5730; 30.0000.)



2. Write a program that will calculate the gas and oil cost for Linda Leadfoot's planned vacation. The car gets about 33 miles per gallon but uses a quart of oil every 350 miles. Use a conditional test to see if the mileage is greater than 350 miles. The following flowchart will help you write the program.

Run the program to find the oil and gas cost for Linda's proposed trip to Seattle, Washington. The round-trip is 494 miles. Oil is $0.75 per quart and gas is $0.69 per gallon.

(Answer: Oil cost is $1.06 and gas is $10.33.)



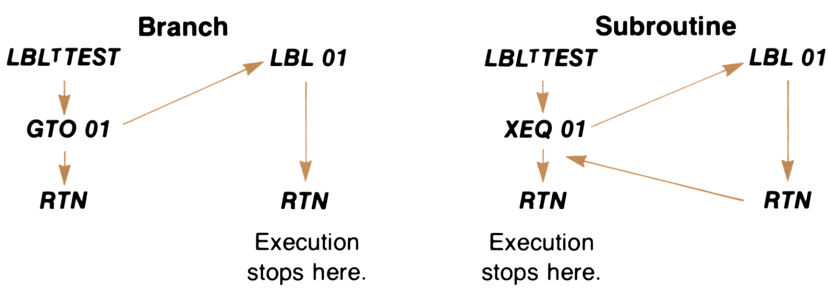
(This page intentionally left blank.)

Section 12

Subroutines

Often, a program contains a certain series of instructions that are executed several times in several places in a program. Or a program requires a set of instructions that are included in another program. These instructions can be executed by a program as a subroutine. A subroutine is selected and executed in a program by the [XEQ] *(execute)* function. Using [XEQ], you can select either ALPHA labeled or numeric labeled subroutines.

In a program, [XEQ] transfers execution to the program label specified by the [XEQ] function. After the subroutine has been executed, and the running program executes an [END] or [RTN], execution is transferred back to the main program. Execution then continues with the next instruction after the [XEQ] and sequentially down through the program. Note that a [GTO] merely transfers execution to the specified label but does not return execution to the main program. The illustration below should make clear the distinction between [GTO] and [XEQ].



In the illustration of a branch, on the left, if you ran program TEST, the program would execute instructions sequentially downward through program memory. When it encountered the [GTO] 01 instruction, it would then search for the next [LBL] 01 in the program, and continue execution until it encountered an [END] or [RTN]. At that point, execution would halt.

However, if you ran the TEST program on the right, the program would execute instructions sequentially downward through program memory until it encountered the [XEQ] 01 instruction. It would then search for the next [LBL] 01 in the program, and then continue execution there. When it encountered an [RTN], program execution would be transferred again, this time back to the main program. It would then resume with the next instruction after the [XEQ] 01.

As you can see, the only difference between a subroutine and a normal branch is the transfer of execution *after* the END or RTN . After the GTO , the next END or RTN halts a running program. After an XEQ , the next END or RTN returns execution back to the main program, where it continues until another END or RTN is encountered.

Subroutine Types and Label Searching

Basically, there are two types of subroutines that you can use in your programs. Subroutines are either *inside* the program file or *outside* the program file. Each of these types of subroutines must be terminated properly. Here are some details.

1. Numeric labels and local ALPHA labels (A through J and a through e, more about these later) are used for programs and subroutines *inside* the program file. The calculator searches for these labels *inside* the current program file only.

   Searches for numeric labels and local ALPHA labels begin at the current position in a program and progress downward through the program to the first END . If the label is not found, searching begins at the beginning of the program file and downward to where the search began. If the label is still not found, the display will show ***NONEXISTENT***.

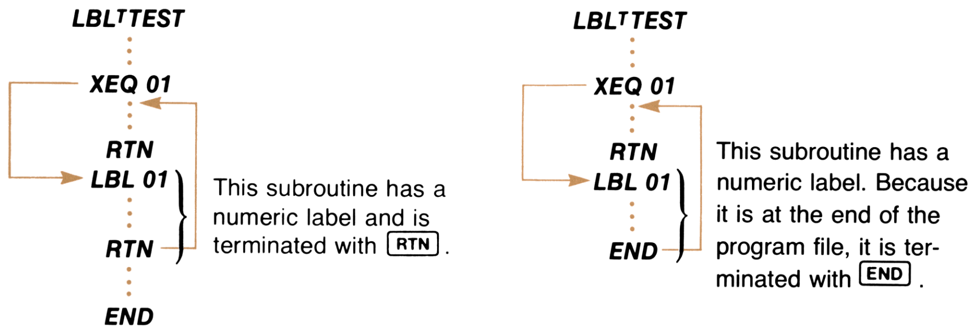
   Programs and subroutines inside program files are usually terminated with RTN . This is because the main program file that they are part of has its own beginning label and ends with an END . However, if the subroutine is at the end of the program file, the END of the program file will suffice to also end the subroutine.

2. Programs with ALPHA labels are generally used for programs and subroutines *outside* other programs. The calculator searches *all* of program memory for ALPHA labels. The ALPHA label search begins with the *last* ALPHA label in program memory and upward through all of the ALPHA labels in program memory. If the label is not found, the display will show ***NONEXISTENT***.

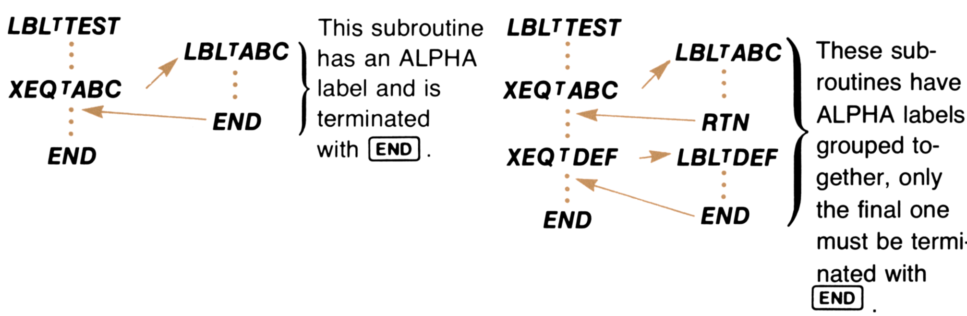
   Programs and subroutines outside program files are usually terminated with END . This is because they must stand alone as separate programs in program memory.

   Note that several subroutines or subprograms can be grouped together as a single "program." All but the final routine should be terminated with RTN instructions. The final routine should be terminated with END . In this case, each of these subroutines can be labeled with ALPHA labels.

Subroutines Inside The Program File

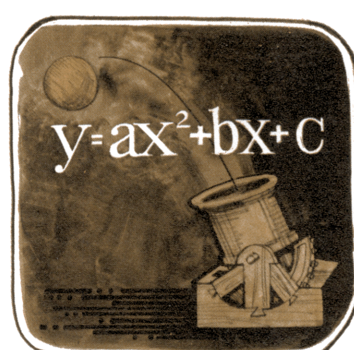


Subroutines Outside The Program File



Example: A quadratic equation is of the form $ax^2+bx+c=0$. One way to find its two roots is by using the formulas:

$$r_1 = \frac{-b+\sqrt{b^2-4ac}}{2a} \text{ and } r_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}.$$



Notice the similarity between the solutions for r_1 and r_2. The program below prompts you for the values of a, b, and c, stores those values in storage registers R_{01}, R_{02}, and R_{03}, and solves for the real roots r_1 and r_2.*

*Some values of a, b, and c may result in misleading answers because their solutions require greater than 12 digits of accuracy.

Here is a complete program for calculating the two roots of a quadratic equation:

```
00
01 LBLTQROOT
02Ta?
03 PROMPT
04 STO 01
05Tb?
06 PROMPT
07 STO 02
08Tc?
09 PROMPT
10 STO 03
11 RCL 02
12 CHS
13 RCL 02
14 X↑2
15 RCL 01
16 RCL 03
17 *
18 4
19 *
20 -
21 SQRT
22 -
23 RCL 01
24 2
25 *
26 /
27 PSE
```

These sections of program memory are identical.

```
28 RCL 02
29 CHS
30 RCL 02
31 X↑2
32 RCL 01
33 RCL 03
34 *
35 4
36 *
37 -
38 SQRT
39 +
40 RCL 01
41 2
42 *
43 /
44 PSE
45 END
```

Since the routine for calculating r_1 contains a large section that is identical to a large section in the routine for calculating r_2, you can simply create a subroutine out of the duplicated instructions. The subroutine is then executed in both the solutions for r_1 and r_2. This subroutine is *inside* the program file. Since it occurs at the end of the program file, the END for the program file also acts as the end of the subroutine.

The program with a subroutine would look like this:

```
01 LBLTQROOT
02Ta?
03 PROMPT
04 STO 01
05Tb?
06 PROMPT
07 STO 02
08Tc?
09 PROMPT
10 STO 03
11 XEQ 01
12 -
13 RCL 01
14 2
15 *
16 /
17 PSE
18 XEQ 01
19 +
20 RCL 01
21 2
22 *
23 /
24 PSE
25 RTN
26 LBL 01
27 RCL 02
28 CHS
29 RCL 02
30 X↑2
31 RCL 01
32 RCL 03
33 *
34 4
35 *
36 -
37 SQRT
38 END
```

With this version of the program, execution begins with the label in line 1 and continues until the XEQ 01 in line 11. At this point, execution is transferred to the LBL 01 in line 26; this is the beginning of the subroutine. When the END in line 38 is encountered, execution is transferred back to line 12, the − instruction. Root r_1 is displayed and the program continues.

When the XEQ 01 in line 18 is encountered, execution is transferred again to the LBL 01 in line 26. When the END in line 38 is encountered, execution transfers back to line 19 and root r_2 is displayed.

The use of the subroutine saved you seven lines of program memory!

Before you key in the program, you may wish to clear other programs from program memory. Do so by executing CLP and specifying the name of the program you wish to clear. Remember, if you are in doubt as to what is in program memory, simply list CATALOG 1.

Keystrokes	Display	
PRGM		
GTO • •	00 REG 46	
LBL		
ALPHA QROOT ALPHA	01 LBL⊤QROOT	
ALPHA a? ALPHA	02⊤a?	
XEQ		
ALPHA PROMPT ALPHA	03 PROMPT	Prompts and stops for input.
STO 01	04 STO 01	
ALPHA b? ALPHA	05⊤b?	
XEQ		
ALPHA PROMPT ALPHA	06 PROMPT	Prompts and stops for input.
STO 02	07 STO 02	
ALPHA c? ALPHA	08⊤c?	
XEQ		
ALPHA PROMPT ALPHA	09 PROMPT	Prompts and stops for input.
STO 03	10 STO 03	
XEQ 01	11 XEQ 01	
−	12 −	
RCL 01	13 RCL 01	Calculates and pauses to display r_1.
2	14 2_	
×	15 *	
÷	16 /	
XEQ		
ALPHA PSE ALPHA	17 PSE	
XEQ 01	18 XEQ 01	
+	19 +	
RCL 01	20 RCL 01	
2	21 2_	
×	22 *	Calculates and pauses to display r_2.
÷	23 /	
XEQ		
ALPHA PSE ALPHA	24 PSE	
RTN	25 RTN	Final execution stops here.
LBL 01	26 LBL 01	Beginning of the subroutine.
RCL 02	27 RCL 02	
CHS	28 CHS	
RCL 02	29 RCL 02	
x^2	30 X↑2	

Keystrokes	Display	
RCL 01	31 RCL 01	
RCL 03	32 RCL 03	
×	33 *	
4	34 4_	
×	35 *	
−	36 −	
√x	37 SQRT	End of the subroutine.
■ GTO • •	00 REG 38	

Run the QROOT program now to find the roots of the equation $x^2+x-6=0$ ($a=1$, $b=1$, $c=-6$); of $3x^2+2x-1=0$ ($a=3$, $b=2$, $c=-1$):

Keystrokes	Display	
PRGM	0.0000	Takes the HP-41C out of PRGM mode.
XEQ		
ALPHA QROOT ALPHA	a?	
1 R/S	b?	
1 R/S	c?	
6 CHS R/S	-3.0000	The first root.
	2.0000	The second root.
XEQ		
ALPHA QROOT ALPHA	a?	
3 R/S	b?	
2 R/S	c?	
1 CHS R/S	-1.0000	The first root.
	0.3333	The second root.
■ CLX	0.0000	

If the quantity b^2-4ac is a negative number, the calculator will display ***DATA ERROR*** to let you know that the program has attempted to find the square root of a negative number. The program will stop running.

Details of Subroutine Usage

Subroutines give you superb versatility in programming. A subroutine can contain a loop, or it can be executed as part of a loop. Subroutines can even be complete programs with their own ALPHA labels; separate from the program that executes it.

You can use a specific numeric label (like LBL 10) any number of times in the programs you write. When you branch to that label, the calculator finds the first occurance of that label in the current program beginning from the present location in the program. Refer to **Subroutine Types and Label Searching, page 178**, for more information.

However, note that you should use caution when using the same ALPHA label more than one time. Since the HP-41C searches *all* of program memory from the bottom up for ALPHA labels, only the last occurrence of that label in program memory will ever be found.

After the first execution of a subroutine, the HP-41C "remembers" the location of most numeric labels. Subsequent branches to those labels do *not* require the time-consuming search. Refer to appendix G for more details about label searching.

When a program is labeled with an ALPHA label, the HP-41C begins searching through all ALPHA labels beginning at the *bottom* of program memory. If the ALPHA label is not found, the display will show ***NONEXISTENT***.

Beginning with the introduction of this handbook, you have written and executed several programs that relate to the heat loss of a cylindrical water heater. These programs included HEAT, CIRCLE, and AREA. Let's now bring all of these programs together and form one master program that uses these programs to find the heat loss of the water heater. To begin, make sure all of these programs have been cleared from program memory because you will make minor changes and re-load them. Use CLP and specify the program name to clear them.

You will create three new programs: BTU, AREA, and TEMP. BTU is the master program that executes the other programs as subroutines and gives the final answer. AREA computes the area of a cylinder given its height and radius, and TEMP computes the temperature difference between the heater surface and the air around the heater. Since AREA and TEMP are *outside* the master program, they have ALPHA labels and are terminated with END instructions.

Since you will use PROMPT so many times when you input the following programs, first assign the PROMPT function to the Σ+ key for use in USER mode. Then, each time you wish to insert a PROMPT instruction in a program, simply press Σ+ in USER mode.

Keystrokes	**Display**
■ ASN	ASN _
ALPHA PROMPT ALPHA	ASN PROMPT _
Σ+	0.0000
USER	0.0000

Begin by loading the master program, BTU.

Keystrokes	Display	
PRGM		
GTO • •	00 REG 45	
LBL		
ALPHA BTU ALPHA	01 LBL^{T}BTU	The master program name.
XEQ		
ALPHA TEMP ALPHA	02 XEQ^{T}TEMP	Executes the TEMP program (to be loaded later) as a subroutine.
XEQ		
ALPHA AREA ALPHA	03 XEQ^{T}AREA	Executes the AREA program (to be loaded later) as a subroutine.
×	04 *	
.47	05 .47_	The convective heat transfer coefficient.*
×	06 *	Computes the final result.
ALPHA		
LOSS=	07^{T}LOSS=_	The final label.
ARCL • X	08 ARCL X	Recalls the answer into the ALPHA register.
AVIEW ALPHA	09 AVIEW	Displays the label and answer.
GTO • •	00 REG 40	

Now load the TEMP program.

Keystrokes	Display	
GTO • •	00 REG 40	
LBL		
ALPHA TEMP ALPHA	01 LBL^{T}TEMP	The program name.
ALPHA HEATER? ALPHA	02^{T}HEATER?	
PROMPT (Σ+)	03 PROMPT	Prompts and stops for input.
ALPHA AIR? ALPHA	04^{T}AIR?	
PROMPT (Σ+)	05 PROMPT	Prompts and stops for input.
−	06 −	Finds difference.
GTO • •	00 REG 36	

* Note that the convective heat transfer coefficient is an approximation of the actual coefficient. Care was used to find a value that resulted in acceptable values for the largest temperature span, cylinder area, cylinder position, and construction. The coefficient actually changes as all of these variables change.

Finally, load the AREA program.

Keystrokes	Display	
■ [GTO] [•] [•]	00 REG 36	
■ [LBL]		
[ALPHA] AREA [ALPHA]	01 LBLTAREA	The program name.
[ALPHA] HEIGHT? [ALPHA]	02THEIGHT?	
[PROMPT] ([Σ+])	03 PROMPT	Prompts and stops for input.
[ALPHA] RADIUS? [ALPHA]	04TRADIUS?	Prompts for data.
[PROMPT] ([Σ+])	05 PROMPT	Stops for input.
[STO] 08	06 STO 08	
■ [x^2]	07 X↑2	
■ [π]	08 PI	Computes area of top and bottom.
[×]	09 *	
2	10 2_	
[×]	11 *	
[x⇄y]	12 X<>Y	
[RCL] 08	13 RCL 08	
[×]	14 *	
■ [π]	15 PI	Computes area of the cylinder without the top and bottom.
[×]	16 *	
2	17 2_	
[×]	18 *	
[+]	19 +	Gives the total area.
■ [GTO] [•] [•]	00 REG 30	

We now have three programs in program memory that help determine the heat loss from the water heater. AREA and TEMP, however, can stand alone as independent programs and you can run these to find just the area or temperature difference. BTU, on the other hand, uses AREA and TEMP as subroutines. If those subroutines do not exist in program memory when you run BTU, the program cannot run in entirety. The calculator will search for the labels, but if they cannot be found, it will display ***NONEXISTENT*** .

Run the BTU program now to find the heat loss (BTUs per hour) from a large cylindrical water heater with a height of 17.48 feet and a radius of 4 feet. The ambient room temperature is 79 degrees Fahrenheit and the temperature of the surface of the heater is 152 degrees Fahrenheit.

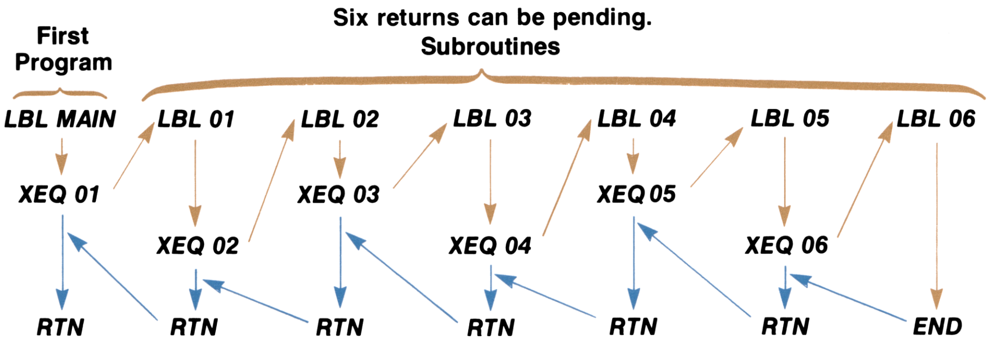
Keystrokes	Display	
PRGM	*0.0000*	Takes the HP-41C out of PRGM mode.
XEQ		
ALPHA BTU ALPHA	*HEATER?*	
152	*152 _*	
R/S	*AIR?*	
79	*79 _*	
R/S	*HEIGHT?*	
17.48	*17.48 _*	
R/S	*RADIUS?*	
4	*4 _*	
R/S	*LOSS=18,522.2975*	Btu per hour.

If you want only the temperature difference or the area, run just those programs (TEMP or AREA). Run BTU again for a water heater that is 6.2 feet high and has a radius of 1.1 feet. The room temperature is 66 degrees Fahrenheit and the temperature of the surface of the water heater is 89 degrees Fahrenheit.

Keystrokes	Display	
XEQ		
ALPHA BTU ALPHA	*HEATER?*	
89	*89 _*	
R/S	*AIR?*	
66	*66 _*	
R/S	*HEIGHT?*	
6.2	*6.2 _*	
R/S	*RADIUS?*	
1.1	*1.1 _*	
R/S	*LOSS=545.4075*	Btu per hour.
■ CLX	*0.0000*	

Subroutine Limits

A subroutine can call up another subroutine, and that subroutine can call up yet another. In fact, you can have up to six subroutine branches before returning to the first program. Subroutine branching is limited only by the number of END s or RTN s that can be held pending by the calculator. Six subroutines can be held pending at any one time in the HP-41C. The illustration below should make this more clear.



The calculator can return back to the first program from subroutines that are six deep, as shown. However, if you call up subroutines that are more than six deep, the calculator will return only six subroutines deep. For example, if you call up seven subroutines deep, when the seventh subroutine is completed, execution will transfer back only six subroutines, back to the second subroutine executed.

Naturally, the calculator can execute an [END] or [RTN] instruction as a stop any number of times. Also, if you execute any of the subroutines *manually* from the keyboard (or you press [RTN]) all *pending* [END] and [RTN] instructions are forgotten by the calculator.

Single-Line Execution of Subroutines

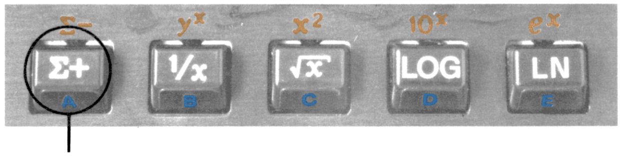
If you are executing a program one step at a time with the [SST] key in normal mode, and encounter an [XEQ] instruction, the calculator will then transfer execution to the specified subroutine. You can then execute the subroutine one line at a time with [SST]. When you encounter the [END] or [RTN] in the subroutine, execution transfers back just like a running program. You can execute programs this way, with [SST], and the HP-41C will remember up to six pending returns, as in a running program.

Local Labels

Earlier in section 7 you learned how to label or name a program with a string of ALPHA characters. There are 15 ALPHA labels on the HP-41C that have special functions that are called "local labels." These 15 labels are [LBL] A through [LBL] J and [LBL] a through [LBL] e (shifted A through E). Any time you label a portion of a program or a subroutine with one of these labels, it is a local label.

When the HP-41C is in USER mode and you press one of the keys in the top two rows (or and a top row key), the calculator immediately begins searching for the corresponding (A through J, a through e) local label *within the current program*. If the local label is not found, the calculator executes the function printed on the face of, or above, the key.

For example, when you press [Σ+] in USER mode, the calculator first searches for a [LBL] A instruction in the current program. The calculator searches downward from the current position in program memory to the end of the program. Then it begins searching at the beginning of the program and back to where the search began.



When you press [Σ+] in USER mode, the HP-41C first searches for LBL A within the current program.

If there is no [LBL] A in the current program, the calculator executes the [Σ+] function. Remember, the calculator searches only the current program for the local label, it does *not* search all of program memory.

If there *is* a [LBL] A in the current program, execution begins at that point. Using the local labels requires the calculator to be set to the portion of program memory containing the local label prior to running the program.

*When you reassign any other function to the top two row locations for execution in USER mode, the local label search is not performed for that particular reassigned location.**

Example: The following program, named SPEED, computes distance (given rate and time), rate (given distance and time), or time (given distance and rate). While in USER mode, you press A when you wish to compute a distance, B when you wish to compute a rate, and C when you wish to compute a time. The program prompts you for the required data. Since you assigned [PROMPT] to [Σ+] for USER mode operation earlier in this section, simply press [Σ+] in USER mode when you wish to load a [PROMPT].

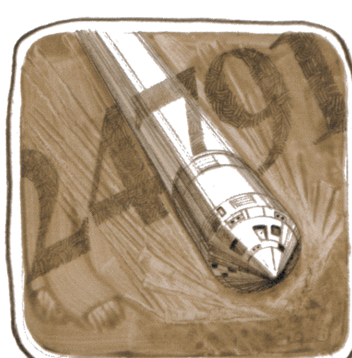
Keystrokes	Display	
[PRGM]		
■ [GTO] [•] [•]	00 REG 45	
■ [LBL]		
[ALPHA] SPEED [ALPHA]	01 LBLᵀSPEED	The main program.
[ALPHA] A,B,OR C? [ALPHA]	02ᵀA,B,OR C?	
[PROMPT] ([Σ+])	03 PROMPT	
■ [LBL]		
[ALPHA] A [ALPHA]	04 LBL A	Local label A.

* Execution of the *normal* mode functions on the top rows of keys in *USER* mode may take several seconds. The calculator must first search through the current program for the local label associated with that key. If no local label is found, the normal mode function is then executed. This is only true when no other function has been assigned to that key for USER execution. To shorten this search time, press ■ [GTO] [•] [•].

Keystrokes	Display	
ALPHA RATE? ALPHA	05^{T}RATE?	
PROMPT (Σ+)	06 PROMPT	
ALPHA TIME? ALPHA	07^{T}TIME?	
PROMPT (Σ+)	08 PROMPT	
×	09*	
■ RTN	10 RTN	End of subroutine A.
■ LBL		
ALPHA B ALPHA	11 LBL B	Local label B.
ALPHA DISTANCE? ALPHA	12^{T}DISTANCE?	
PROMPT (Σ+)	13 PROMPT	
ALPHA TIME? ALPHA	14^{T}TIME?	
PROMPT (Σ+)	15 PROMPT	
÷	16 /	
■ RTN	17 RTN	End of subroutine B.
■ LBL		
ALPHA C ALPHA	18 LBL C	Local label C.
ALPHA		
DISTANCE? ALPHA	19^{T}DISTANCE?	
PROMPT (Σ+)	20 PROMPT	
ALPHA RATE? ALPHA	21^{T}RATE?	
PROMPT (Σ+)	22 PROMPT	
÷	23 /	
■ GTO • •	00 REG 33	End of subroutine C.

Now run the program to solve the following problem:

On May 26, 1969 the Command and Service Module of Apollo X carried U.S. astronauts Stafford, Cernan, and Young at a rate of 24,791 miles per hour (the fastest speed at which any human has traveled). How far would the module travel in 2.5 hours?



$$D = RT = 24{,}791 \times 2.5$$

Before you begin, be sure that all of the upper-row keys do not have any functions assigned to them. For example, PROMPT is now assigned to the Σ+ key location. To remove the assignments:

Keystrokes	Display
■ ASN	ASN _
ALPHA ALPHA	ASN _
Σ+	00 REG 33

Now run the program. Make sure the calculator is in USER mode.

Keystrokes	Display	
PRGM	0.0000	
XEQ		
ALPHA SPEED ALPHA	A,B,OR C?	
A (Σ+)	RATE?	
24791	24,791 _	The rate.
R/S	TIME?	
2.5	2.5 _	The time.
R/S	61,977.5000	Miles in 2.5 hours.

Now run the program (local label B) to find the rate of travel of the first Antarctic continent crossing from Shackelton Base to Scott Base by way of the Pole. The crossing spanned 2,158 miles and took 99 days.

$$R = D \div T = 2{,}158 \div 99$$

Keystrokes	Display	
B (1/x)	DISTANCE?	
2158	2,158 _	The distance.
R/S	TIME?	
99	99 _	
R/S	21.7980	Miles per day.

Finally, run the program (local label C) to find the time for a tsunami (a large wave caused by a seaquake) to reach the southern shore of the Pacific Island Iwo. The wave is traveling at a constant 2.25 meters per second and is 300 meters off shore.

$$T = D \div R = 300 \div 2.25$$

Keystrokes	Display	
C (√x)	*DISTANCE?*	
300	*300_*	The distance.
R/S	*RATE?*	
2.25	*2.25_*	The rate.
R/S	*133.3333*	Seconds.

You can continue to execute the local label programs any number of times using the local label keys *without executing the main program each time*. All you do is press A (Σ+), B (1/x), or C (√x) in USER mode. But when the calculator is positioned outside of the SPEED program, pressing the local keys search only the *current* program. If they are not found, the function printed on or above the key is executed.

Problems

1. Look closely at the program for finding roots r_1 and r_2 of a quadratic equation (page 182). Can you see other instructions that could be replaced by a subroutine? (Look at lines 13 through 17 and lines 20 through 24.) 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: -3.0000, 2.0000; -1.0000, 0.3333.)

   Did you save any more lines of program memory?

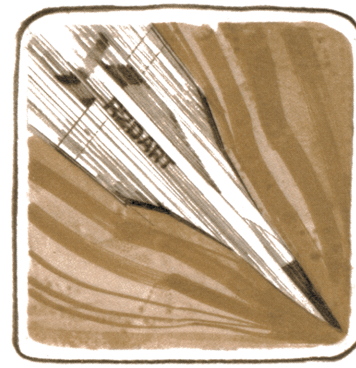
2. The surface area of a sphere can be calculated according to the equation $A=4\pi r^2$, where r is the radius. The formula for finding the volume of a sphere is $V=(4\pi r^3)\div 3$. This may also be expressed as $V=(r\times A)\div 3$.

   Create and load a program to calculate the area A of a sphere given its radius r. Name the program SAREA and include an initialization routine to prompt for the value of the radius. Then create and load a second program to calculate the volume V of a sphere, using the equation $V=(r\times A)\div 3$. Name this second program VOLUME and include an XEQ SAREA to use SAREA as a subroutine to calculate area.

   Run the two programs to find the area and volume of the planet Earth, a sphere with a radius of about 3963 miles and of the Earth's moon, a sphere with a radius of about 1080 miles.

   (Answers: Earth area = 197,359,487.5 square miles,
   Earth volume = 2.6071188×10^{11} cubic miles;
   Moon area = 14,657,414.69 square miles,
   Moon volume = 5,276,669,290 cubic miles.)

3. Daredevil test pilot Trigo Skywalker is diving in a wingless R2DART experimental aircraft at an angle of 45 degrees and a velocity of 745 meters/second. Suddenly, at an altitude of 7460 meters, the R2 loses power and Skywalker parachutes to safety. How long after the R2 loses power does it fly before crashing? (Effects of atmospheric drag and variation in the gravitational acceleration are ignored.)



Solution: The equation describing the fall of the plane is:

$$y = -(g \div 2)t^2 - \mathbf{vt} + y_i$$

where y is the altitude. (In our problem y=0 when the plane crashes.)

g is the acceleration of gravity, 9.80665 m/s².

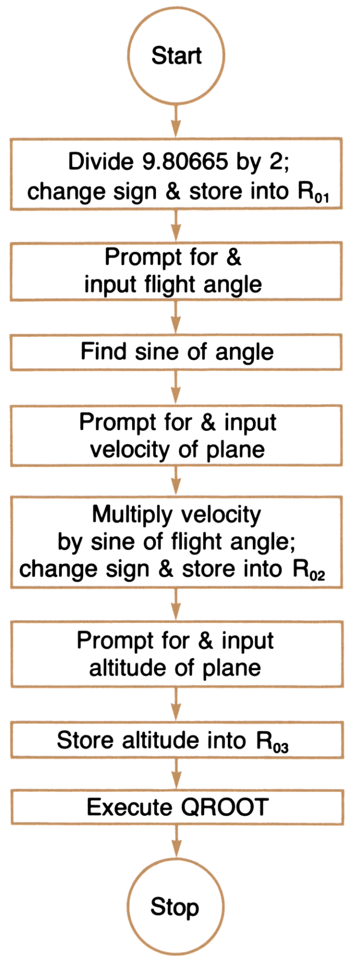
v is the vertical component of the velocity when power was lost. It is found by multiplying the velocity by the sine of the flight angle.

y_i is the initial altitude.

t is the time in flight after power failure (seconds).

(Answer: 12.6675 seconds.)

Method: Modify the QROOT program that you loaded earlier in this section (page 182) so that it no longer prompts for the input of *a*, *b*, and *c*. Write a second program, based on the following flowchart, that finds the values of *a* ($-g \div 2$), *b* ($-v$), and *c* (y_i). The *a* should be stored into R_{01}, *b* into R_{02} and *c* into R_{03}. The second program should use QROOT as a subroutine. The following flowchart will help you write the program. (Only the positive root is valid as an answer to the problem.)


Start
Divide 9.80665 by 2; change sign & store into R_{01}
Prompt for & input flight angle
Find sine of angle
Prompt for & input velocity of plane
Multiply velocity by sine of flight angle; change sign & store into R_{02}
Prompt for & input altitude of plane
Store altitude into R_{03}
Execute QROOT
Stop

(This page intentionally left blank.)

Section 13

Indirect Operations

An important feature of the HP-41C is the numerous indirect operations the calculator can perform. Any storage register in the HP-41C can be used for indirect operations. This capability greatly expands the power and utility of your HP-41C. An indirect address is selected by following a function with the shift key, , and then a register address. The function then uses the number in the specified register as an *address*. Indirect operations are most useful in programming.

For future reference, here is a complete listing of all HP-41C functions that can be used with indirect addresses:

STO nn	Store.
STO + nn	Store add (keyboard form).
STO − nn	Store subtract (keyboard form).
STO × nn	Store multiply (keyboard form).
STO ÷ nn	Store divide (keyboard form).
ST+ nn	Store add (display form).
ST− nn	Store subtract (display form).
ST× nn	Store multiply (display form).
ST+ nn	Store divide (display form).
ASTO nn	ALPHA store.
RCL nn	Recall.
ARCL nn	ALPHA recall.
VIEW nn	View register contents.
GTO nn	Go to.
XEQ nn	Execute.
FIX nn	FIX display format.
SCI nn	SCI display format.
ENG nn	ENG display format.
DSE nn	Controlled decrement loop.
ISG nn	Controlled increment loop.
TONE nn	Audible tone pitch.
ΣREG nn	Define accumulation registers.
SF nn	Set flag.
CF nn	Clear flag.

FS? ■ nn	"Flag set" test.
FC? ■ nn	"Flag clear" test.
FS?C ■ nn	"Flag set" test and clear.
FC?C ■ nn	"Flag clear" test and clear.
X<> ■ nn	Exchange X and any register.
CATALOG ■ nn	Catalog list.

To use an indirect address with a function, first store the desired register address number (the direct address) in the register you are using for indirect control. Then execute the function and press ■ and specify the indirect address. When you press ■, the HP-41C prompts you for the indirect address. Indirect addressing will become more clear as you read on in this section.

You can indirectly address any of the primary storage registers or extended storage registers (if your HP-41C has been extended with additional memory modules) currently allocated in the HP-41C Remember, you can allocate up to 63 primary storage registers on your basic HP-41C. When you extend the HP-41C memory with plug-in memory modules, the storage register capacity of the calculator is increased. All extended registers ($R_{(100)}$ through $R_{(318)}$) *require* the use of indirect addressing.

If the indirect or direct register address is outside the limits of the current allocation or the number of registers in the HP-41C, the display will show ***NONEXISTENT***. In all cases only the absolute value of the integer portion of the register address is used by the calculator.

Indirect Store and Recall

To store and recall numbers indirectly using any of the primary or extended storage registers, simply press STO or RCL, ■, and then specify the indirect address. By changing the register address number, you can change the address specified by the function.

You can easily demonstrate how indirect store and recall work by using the HP-41C manually. For example, to store the number 2.54 into R_{10} using R_{02} as an indirect address register:

Keystrokes	Display	
10 STO 02	*10.0000*	First store the desired register address (R_{10}) into the indirect address register (R_{02}).
2.54	*2.54 _*	The number.
STO ■	*STO IND _ _*	Notice how the HP-41C prompts you for the indirect address.
02	*2.5400*	The number 2.5400 is now stored into register R_{10}.

Here is what happened when you used the indirect address to store the number.

The Function

2.54 [STO] 02

The Indirect Address Register

R_{02} 10.0000

The Desired Register

R_{10} 2.5400

To recall numbers that are stored in any primary storage registers (R_{00} through R_{99}), you can simply press [RCL] and the number keys of the register address. You can also recall numbers from primary storage registers using indirect addressing, just like you did when you stored the number in the above example. Numbers in the extended storage registers ($R_{(100)}$ through $R_{(318)}$) must be stored and recalled using indirect addressing.

For example, recall the number that is stored in storage register R_{10} using register R_{05} as the indirect address register.

Keystrokes	Display	
10 [STO] 05	*10.0000*	First store the desired register address into the indirect address register.
[RCL]	*RCL IND _ _*	The HP-41C prompts you for the indirect address number.
05	*2.5400*	The number 2.5400 is recalled from storage register R_{10}.

Here is what happened when you used the indirect address to recall the number.

The Function

[RCL] 05

The Indirect Address Register

R_{05} 10.0000

The Desired Register
(Recalled into the X-register.)

R_{10} 2.5400

Storage register arithmetic is performed upon the contents of the indirectly addressed register by using [STO] [+] [■] nn, [STO] [−] [■] nn, [STO] [×] [■] nn, and [STO] [÷] [■] nn. If you do not remember how storage register arithmetic works, turn to page 74 to refresh your memory.

Now, multiply the number in R_{10} by 5280 and then store that value back into R_{10} using R_{11} as an indirect address register.

Keystrokes	Display	
10 [STO] 11	10.0000	
5280	5,280_	
[STO] [×] [■]	ST*IND __	
11	5,280.0000	The number 5280.0000 is multiplied by the number in R_{10}.
[RCL] 10	13,411.2000	The answer.

Indirect ALPHA Store and Recall

The [ASTO] *(ALPHA store)* and [ARCL] *(ALPHA recall)* functions can also be used with indirect addressing, just like with normal [STO] and [RCL]. (Remember that [ASTO] is the shifted function on the [STO] key in ALPHA mode, and [ARCL] is the shifted function on the [RCL] key in ALPHA mode.) Simply store the desired register address number into the indirect address register you choose. Then execute the function, specifying [■] and the indirect register address in response to the prompt.

For example, store the string WATER into R_{08} using R_{00} as the indirect address register.

Keystrokes	Display	
8 [STO] 00	8.0000	
[ALPHA] WATER	WATER_	
[■] [ASTO] [■]	ASTO IND __	
00	WATER	The string, WATER, is now stored in R_{08}.
[■] [CLA]		

Now recall the string using indirect addressing. (Remember, this is done in ALPHA mode.)

Keystrokes	Display	
[■] [ARCL] [■]	ARCL IND __	
00	WATER_	The string, WATER, is recalled into the ALPHA register from R_{08}.
[■] [CLA]		Clears the ALPHA register.
[ALPHA]	8.0000	Back to normal mode.

Indirect Stack and LAST X

Remember from section 5 that you can specify the stack and LAST X as register addresses by simply pressing [•] (decimal point) and X, Y, Z, T, or L (for LAST X). You can also use the stack and LAST X registers as indirect addresses by simply pressing [■] [•] and X, Y, Z, T, or L following the function. For example, to store the number 83.9701 into R_{11} using stack Z as the indirect address register:

Keystrokes	Display	
11 [STO] [•] Z	11.0000	The desired register address (R_{11}) is stored into stack register Z.
83.9701	83.9701_	The 11 is now in register T.
[STO] [■] [•] T	STO IND T	The HP-41C prompts for the stack address. You can only specify a letter (X, Y, Z, T, or L) here, the HP-41C will not accept any other inputs.
	83.9701	
[■] [CLX]	0.0000	

To recall the number that is now in R_{11} using stack Z as the indirect address register:

Keystrokes	Display
[RCL] [■] [•] Z	83.9701
[■] [CLX]	0.0000

You should remember that many functions affect the status of the automatic memory stack (e.g., pushing numbers into the stack), and that when you use the stack registers as storage registers, the normal stack operation may change the contents of those registers.

Indirect Function Control

Now that you have seen how indirect addressing works, let's progress a little and see how some of the other indirect features work in programs.

Functions requiring the input of an operating specification like [TONE] and [FIX] can use indirect addressing to specify how the function is to operate. For example, [FIX] requires a number from 0 through 9 to specify the display format. Using indirect addressing, you can store the format specification number in a register, and then use indirect addressing to complete the function ([FIX] [■] nn). Indirect control is most useful in programs you write.

Example: The following program uses two controlled loops to place a number used by the TONE *(audible tone)* function. The program counts from 0 to 9 and controls the first loop using ISG, then counts back to 0 and controls the second loop using DSE.

Keystrokes	Display	
PRGM		
GTO • •	*00 REG 46*	
LBL		
ALPHA SONG ALPHA	*01 LBL^{T}SONG*	
.009	*02 .009_*	
STO 01	*03 STO 01*	The first loop control number is stored in register R_{01}.
9	*04 9_*	
STO 02	*05 STO 02*	The second loop control number is stored in register R_{02}.
LBL 01	*06 LBL 01*	The beginning of the first loop.
XEQ		
ALPHA TONE ALPHA		
01	*07 TONE IND 01*	TONE uses R_{01} as an indirect address. The TONE function uses the number in R_{01} to control the audible tone in the HP-41C.
ISG 01	*08 ISG 01*	Add one to the loop control number in R_{01}. Test loop control number: if it is not greater than 9, execute loop again; if it is greater than 9 skip the next line.
GTO 01	*09 GTO 01*	Loop to LBL 01.
LBL 02	*10 LBL 02*	The beginning of the second loop.
XEQ		
ALPHA TONE ALPHA		
02	*11 TONE IND 02*	TONE uses R_{02} as an indirect address. The number in R_{02} controls the audible tone.

Keystrokes	Display	
XEQ		
ALPHA DSE ALPHA 02	12 DSE 02	Subtract one from the loop control number in R_{02}. Test loop control number: if it is not less than or equal to zero, execute loop again; if it is, skip the next line.
■ GTO 02	13 GTO 02	Loop to LBL 02.
■ GTO • •	00 REG 42	

Run the program now and listen to the audible tone of the HP-41C as it starts with a low pitch, works up to a high pitch, then back down to the low pitch.

Keystrokes	Display
PRGM	0.0000
XEQ	
ALPHA SONG ALPHA	9.0000

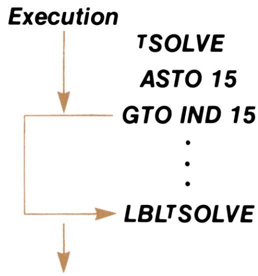
When you run the program, it executes through the first loop until the loop control number in R_{01} equals 9. The TONE function uses the loop control number in R_{01} indirectly as a specification of the TONE value. When the loop control number equals 9, the second loop begins execution until the loop control number equals 0. TONE uses the loop control number in R_{02} indirectly as the TONE specification. The second loop does not execute TONE 0.

Indirect Control of Branches and Subroutines

Like indirect addressing of storage registers, you can address routines, subroutines, even entire programs using indirect addressing.

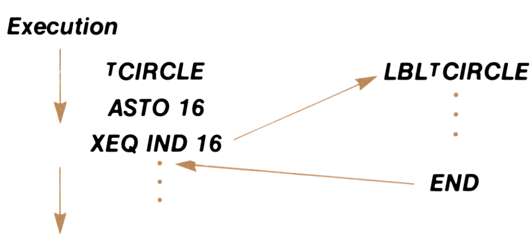
To indirectly address a subroutine with an ALPHA or numeric label (e.g., LBL TRIGO, LBL 10), use the GTO ■ nn *(go to indirect)* instruction in the program. (The calculator displays the prompt *IND* following the function name.) When the running program encounters the *GTO IND nn* instruction, the calculator searches the current program for a numeric label and all of program memory for an ALPHA label that is specified by the indirect address register. (If the label is not found, or if the label is not a legal label—e.g., the numeric label is greater than 99, the display shows *NONEXISTENT*.) Local labels (A through J, a through e) cannot be used indirectly with GTO.

As an example, with the ALPHA label SOLVE stored in register R_{15}, when the *GTO IND 15* instruction is encountered, execution is transferred to the last LBL SOLVE in memory. If the label SOLVE is found, execution resumes there. A GTO to a numeric label will *not* transfer execution out of a program file, but a GTO to an ALPHA label *will* transfer execution out of a program file (refer to section 12 for a complete discussion of label searching, branches and transferring execution).



To indirectly address routines or programs outside of the current program, you can use [XEQ] [■] nn (*execute indirect*). When the running program encounters an **XEQ IND nn** instruction, execution is transferred to the numeric or ALPHA label specified by the indirect address register. The addressed program is executed as a subroutine and control returns to the main program when execution of the subroutine is completed. For example, with the label CIRCLE stored in R_{16}, [XEQ] [■] 16 causes execution of the program defined by [LBL] CIRCLE. Local labels (A through J, a through e) cannot be used indirectly with [XEQ].

Note that only programs that you write and store into program memory and those functions contained in plug-in extensions (such as application module, or the card reader) can be executed indirectly in this manner. Standard HP-41C functions cannot be executed with [XEQ] [■].

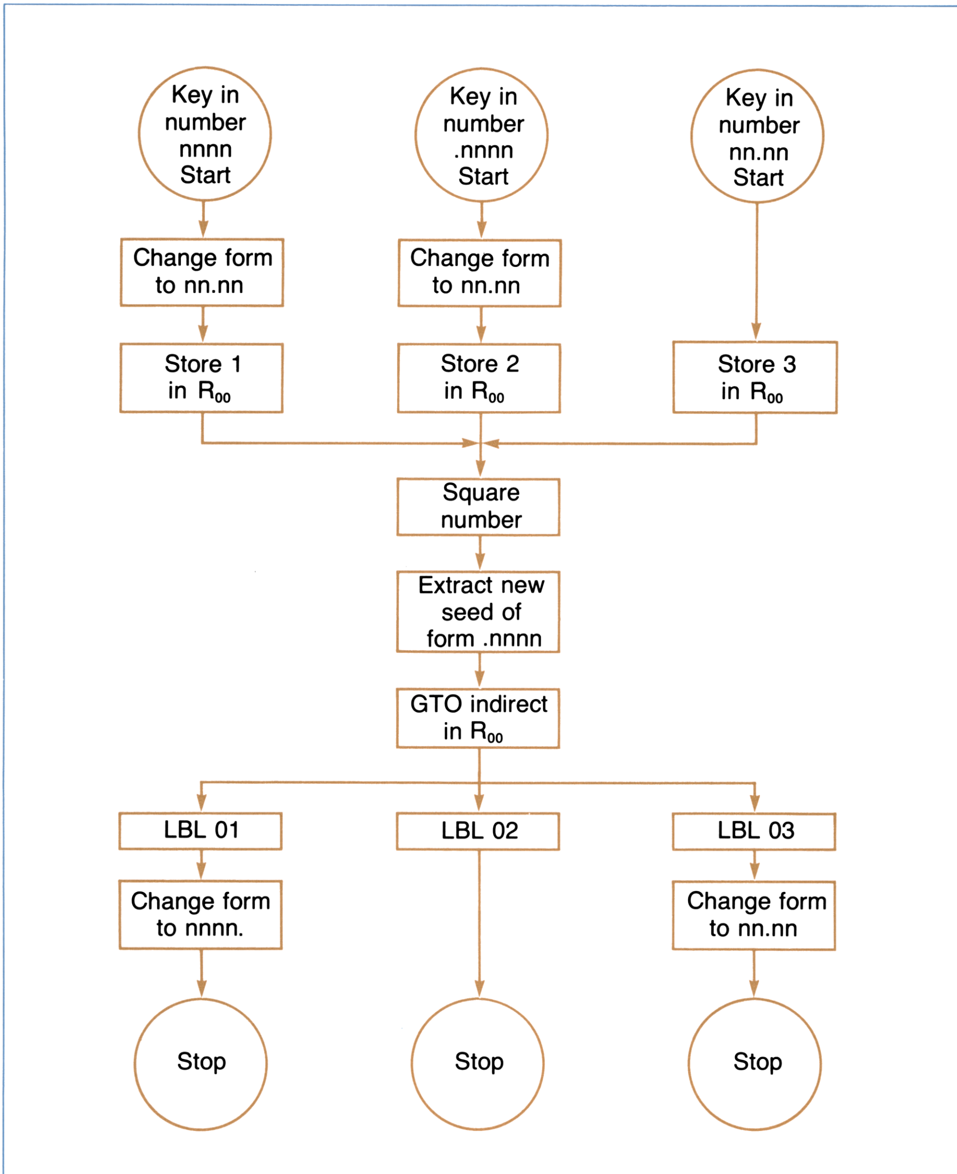


Indirect addressing works the same way with all of the functions listed on page 197.

Problems

1. One method of generating pseudorandom 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 5,182 when squared yields 26,853,124. A random number generator could then extract the four center digits, 8,531, and square that value. Continuing for several iterations through a loop would generate several random numbers. Following is a flowchart and programming hints for such a pseudorandom number generator.

   The seed is a four-digit number in the form of nn.nn, .nnnn, or nnnn. 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.



To change a seed in the form of nnnn., and .nnnn to nn.nn, you can use following keystrokes:

nnnn. to nn.nn	**.nnnn to nn.nn**
EEX 2	EEX 2
÷	×

To change the result, .nnnn, back to the input form, nnnn. or nn.nn:

.nnnn to nnnn.	**.nnnn to nn.nn**
EEX 4	EEX 2
×	×

Truncate the square to extract a new seed of the form .nnnn using:

EEX 2
×
INT
EEX 4
÷
FRC

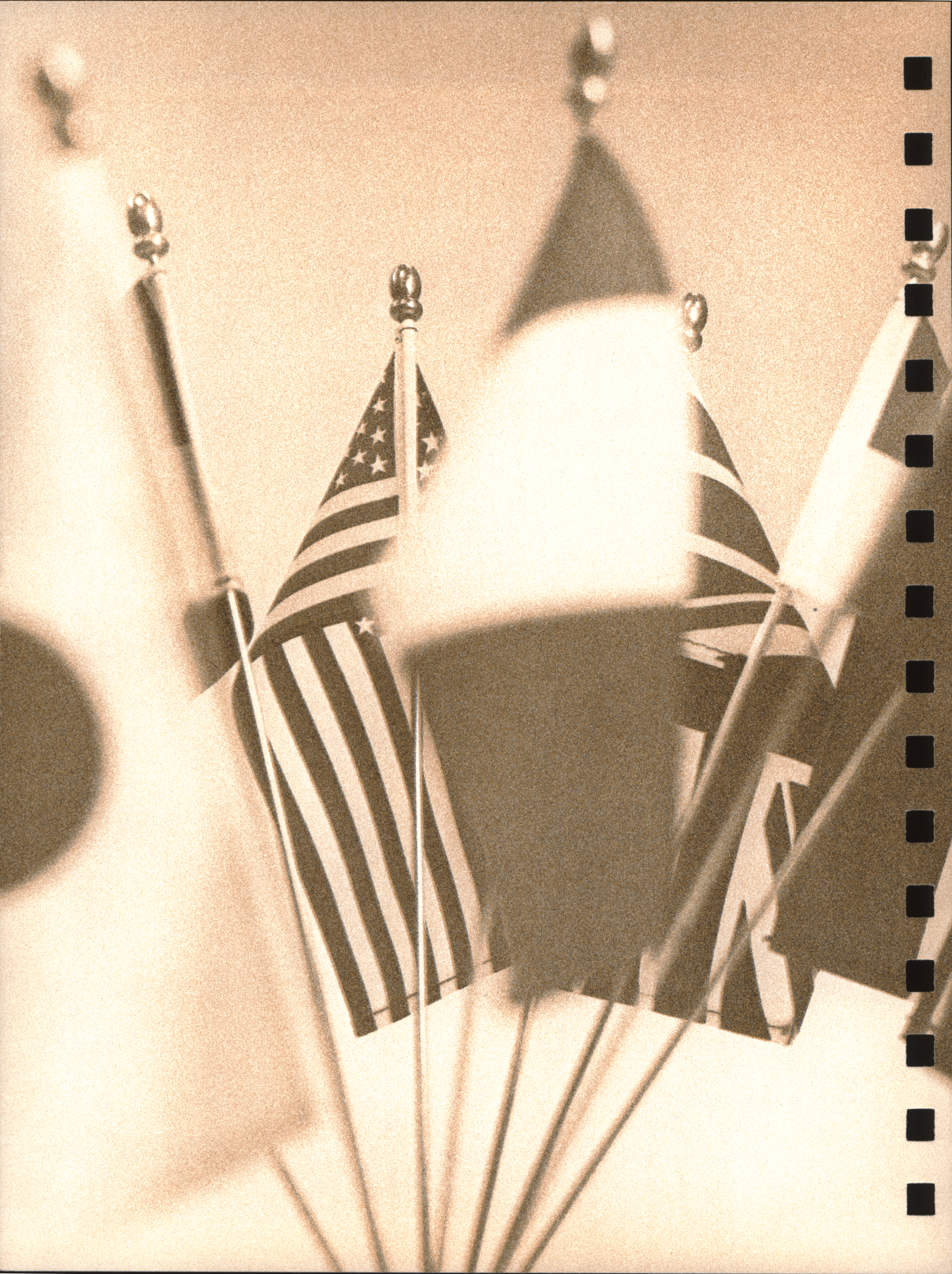
So that you can remember which form to input, you may wish to label the program with three labels, one for each form, like this: LBL NN/NN, LBL /NNNN, and LBL NNNN/. When you input a seed of the form nn.nn, you execute program NN/NN. Likewise, when you input a seed of the form .nnnn or nnnn, you execute program /NNNN or NNNN/. Use the / character in the names, not a period. Periods are not legal in ALPHA program labels.

When you key in a four-digit seed in one of the three formats and execute the associated program, an address (1, 2, or 3) is placed in the R_{00} register. This address is used by a GTO ■ 00 (go to indirect in R_{00}) to transfer program execution to the proper routine so that the new random number is seen in the same form as the original seed.

Run the program for seeds of 1191, 11.91, and .1191. The program generates a random number in the same form as the seed you keyed in. To use the random number as a new seed, continue executing the associated program.

2. Modify the random number generator program you wrote above to use XEQ indirect instead of GTO indirect for control. Run the program with the same seed numbers as above to ensure that it still runs correctly.

(This page intentionally left blank.)



Section 14

Flags

The HP-41C flags are an important programming tool in your calculator. A flag actually is a memory that can either be SET or CLEAR. A running program can then *test* the flag later in the program and make a decision, depending on whether the flag was set or clear.

There are 30 "user" flags (numbered 00 through 29) available in your HP-41C. In addition, there are 26 "system" flags (numbered 30 through 55) that have limited uses to you in your programs. On pages 210 and 211 are tables showing HP-41C flags and their basic capabilities. The HP-41C has six functions that allow you to manipulate the flags.

Three of the flag functions are on the normal mode keyboard. They are:

SF	*(set flag)*
CF	*(clear flag)*
FS?	*("flag set" test)*

The other flag functions are not on the keyboard, but can be assigned to the keyboard for execution in USER mode, or executed from the display (refer to section 4). These flag functions are:

FC?	*("flag clear" test)*
FS?C	*("flag set" test and clear)*
FC?C	*("flag clear" test and clear)*

When you execute one of the six flag functions, the HP-41C prompts you for the flag number (00 through 55) you wish to operate upon.

HP-41C USER FLAGS (00 THROUGH 29)

Flag Name	Flag Number	Set	Clear	Test	Status
General Purpose User Flags (11)	00 through 10	x	x	x	Always maintained by Continuous Memory.
Special Purpose User Flags (10)	11 through 20	x	x	x	Cleared each time the HP-41C is turned on.
Automatic Execution Flag (Special Purpose Flag 11)	11	x	x	x	Cleared each time the HP-41C is turned on.
Printer Enable Flag	21	x	x	x	Matches flag 55 status each time HP-41C is turned on.
Numeric Input Flag	22	x	x	x	Cleared each time the HP-41C is turned on.
ALPHA Input Flag	23	x	x	x	Cleared each time the HP-41C is turned on.
Range Error Ignore Flag	24	x	x	x	Cleared each time the HP-41C is turned on.
Error Ignore Flag	25	x	x	x	Cleared each time the HP-41C is turned on.
Audio Enable Flag	26	x	x	x	Set each time the HP-41C is turned on.
USER Mode Flag	27	x	x	x	Always maintained by Continuous Memory.
Decimal Point Flag	28	x	x	x	Always maintained by Continuous Memory.
Digit Grouping Flag	29	x	x	x	Always maintained by Continuous Memory.

HP-41C SYSTEM FLAGS (30 THROUGH 55)

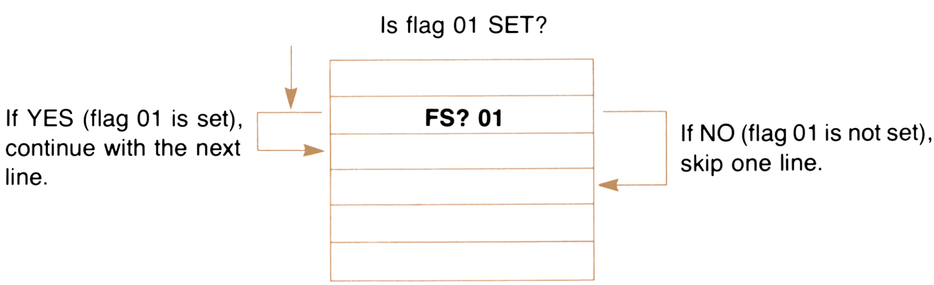
Flag Name	Flag Number	Set	Clear	Test	Status
Catalog Flag	30			x	Not applicable.
Peripheral Flags (5)	31 through 35			x	Not applicable.
Number of Digits Flags (4)	36 through 39			x	Always maintained by Continuous Memory.
Display Format Flags	FIX 40 ENG 41 SCI (refer to page 232)			x x	Always maintained by Continuous Memory.
Grads Mode Flag	42			x	Always maintained by Continuous Memory.
Radians Mode Flag	43			x	Always maintained by Continuous Memory.
Continuous On Flag	44			x	Not applicable.
System Data Entry Flag	45			x	Not applicable.
Partial Key Sequence Flag	46			x	Not applicable.
Shift Set Flag	47			x	Not applicable.
ALPHA Mode Flag	48			x	Cleared each time the HP-41C is turned on.
Low Battery Flag	49			x	Not applicable.
Message Flag	50			x	Not applicable.
SST Flag	51			x	Not applicable.
PRGM Mode Flag	52			x	Cleared each time the HP-41C is turned on.
I/O Flag	53			x	Not applicable.
Pause Flag	54			x	Not applicable.
Printer Existence Flag	55			x	Set if printer exists or clear if no printer each time the HP-41C is turned on.

To begin learning how to use the flags, set flag 00:

Keystrokes	**Display**	
SF	*SF _ _*	The HP-41C prompts you for the flag number.
00	*0.0000*	Flag 00 is now SET. Flag 00 annunciator (***0***) turns on in the display.

Flag decisions are made using the *test* flag functions (FS?, FC?, FS?C, and FC?C). Each of these functions asks a question about the status of the specified flag. In a program, if the answer to the test question is TRUE, the calculator executes the next line in the program (this is the "DO if TRUE" rule again). If the answer to the question is false, the calculator skips the next line in the program before execution continues.

For example, if you use the FS? *("flag set" test)* function to check the status of flag 01 in a program and the flag is set, the next line in the program is executed. If the flag is clear, the next line in the program is *skipped*.



Pressed from the keyboard, these flag functions will show an answer to the test question in the display. If the answer is true, the display shows ***YES***; if the answer is false, the display shows ***NO***.

Two of the flag test functions perform an additional function other than asking a question. These functions, FS?C *("flag set" test and clear)* and FC?C *("flag clear" test and clear),* also clear the specified flag in addition to testing it.

If at *any* time you are unsure as to the status of the flags, there are two ways to tell whether a flag is set or clear. (Remember, the status of some flags is maintained by the Continuous Memory of the HP-41C.)

First, and most simply, you can check the status of flags 00 through 04 by simply looking in the display at the flag display annunciator. If any of these five flags are set, the corresponding number will show in the display annunciator at the bottom of the display window.

Second, you can test the flag with FS? or FC? without changing its status. Pressed from the keyboard, these functions return a ***YES*** or ***NO*** answer to the display.

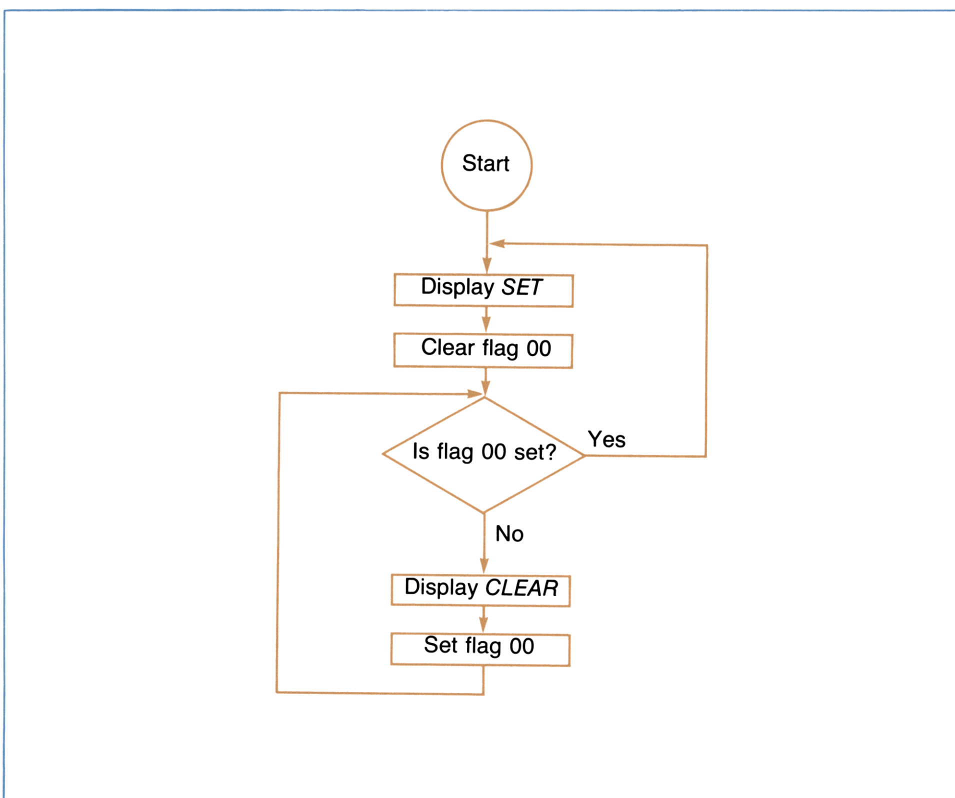
For example, if flag 00 is set and you use [FS?], the display will show *YES*. On the other hand, if flag 00 is set and you use [FC?], the display will show *NO*.

Try testing flags 00 and 01 using [FS?].

Keystrokes	Display	
[■] [FS?] 00	*YES*	Flag 00 was set in an earlier example, so the answer to the test is *YES*. Notice that the display annunciator shows *0*.
[■] [FS?] 01	*NO*	Since flag 01 is not set, the calculator returns an answer of *NO*.

Example: The following program contains an infinite loop that illustrates the operation of a flag. The program alternately displays *SET* and *CLEAR* by changing and testing the status of flag 00. A flowchart for this simple program might look like the one below.

The program assumes that flag 00 is initially set.



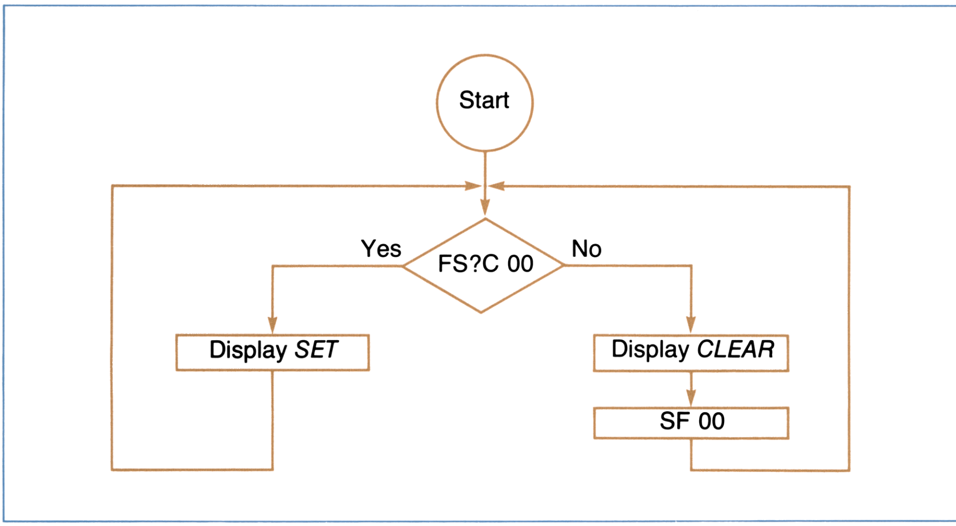
Keystrokes	Display	
PRGM		
■ GTO · ·	00 REG 46	
■ LBL		
ALPHA FLAG ALPHA	01 LBLᵀFLAG	
■ LBL 01	02 LBL 01	Display SET when flag 00 is set.
ALPHA SET	03ᵀSET	
■ AVIEW ALPHA	04 AVIEW	
XEQ		
ALPHA PSE ALPHA	05 PSE	
■ CF 00	06 CF 00	Clear flag 00.
■ LBL 02	07 LBL 02	
■ FS? 00	08 FS? 00	Test flag 00.
■ GTO 01	09 GTO 01	If the test is true, go to LBL 01. Otherwise, display CLEAR, set flag 00 and go to LBL 02.
ALPHA CLEAR	10ᵀCLEAR	
■ AVIEW ALPHA	11 AVIEW	
XEQ		
ALPHA PSE ALPHA	12 PSE	
■ SF 00	13 SF 00	
■ GTO 02	14 GTO 02	
■ GTO · ·	00 REG 41	

Now run the program.

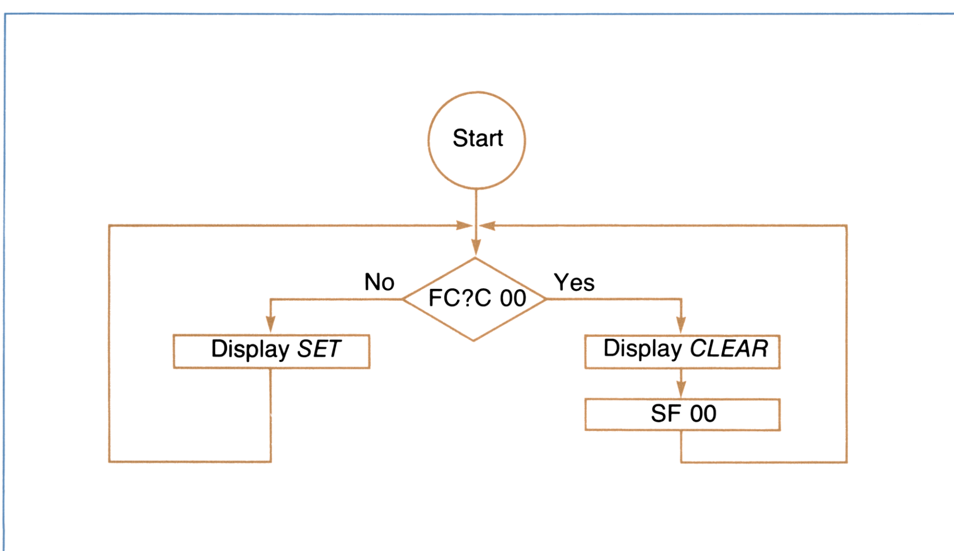
Keystrokes	Display	
PRGM	0.0000	
XEQ		
ALPHA FLAG ALPHA	SET	SET and CLEAR are displayed alternately as the flag changes status. Also notice that the annunciator for flag 00 turns on and off as the flag changes status.
	CLEAR	
	SET	
	CLEAR	
	SET	
	CLEAR	
	SET	
	.	
	.	
	.	
R/S	0.0000	

Problems:

1. Write a new program that does the same operation as the preceding program, but instead use **FS?C** to control the flag status. The following flowchart will help you set up the new program. You should be able to save two lines in program memory over the previous method.



2. Write a third program that performs the same operations as the above programs but change the flag test function again. This time use **FC?C**. Here is a new flowchart.



Flag Descriptions

Over the next few pages are descriptions of all HP-41C flags. Several examples and problems are included to help you become more familiar with how the user flags work.

General Purpose Flags (00 through 10)

The HP-41C is equipped with 11 general purpose user flags (numbered 00 through 10). You have complete control of these flags. They can be set, cleared, and tested using any of the HP-41C flag control functions. Once you set or clear one or all of these flags, that status is maintained by the Continuous Memory of the calculator, even when the HP-41C is turned on and off.

Special Purpose User Flags (11 through 20)

There are 10 special purpose flags in your HP-41C. In addition to their use as flags that you can control, flags 11 through 20 have special functions in the HP-41C. You can test, set and clear these flags using any of the flag commands discussed earlier in this section. However, under certain conditions, the calculator also controls the status of these flags.

When you are using peripheral extensions such as the printer or card reader, it is a good idea to keep in mind that the status of these flags may be altered by the calculator. Refer to the owner's handbook that belongs to the peripheral extension for specific details about flags.

All of these 10 special purpose user flags (11 through 20) are cleared each time the HP-41C is turned on.

Automatic Execution Flag

Flag 11 is one of the special purpose flags described above. Its special purpose in the HP-41C is to control program execution when the HP-41C is turned on.

When flag 11 is set, and you turn the calculator off, the HP-41C automatically begins executing the current program in program memory when you turn the HP-41C back on again. Execution begins at the instruction the calculator was positioned to before the HP-41C was turned off. In addition, the calculator sounds the audible tone before execution begins.

If flag 11 is clear and you turn the calculator off, when you turn the HP-41C back on again, the calculator turns on normally. Program instructions are not executed.

Remember that flag 11 is automatically cleared each time the calculator is turned on.

Printer Enable Flag

This flag (flag 21) is used to enable and disable printing from programs on the HP 82143A Printer. You can set, clear and test this flag just like any other of the general or special purpose flags described above.

When flag 21 is clear, printing by programs is suppressed.

On the other hand, if flag 21 is set, printing by programs is enabled.

Flag 21 has no effect on print functions executed from the keyboard. Execution of any printer function while the printer is not plugged in results in the ***NONEXISTENT*** display.

The status of this flag is set to match the status of flag 55 (the printer existence flag) each time the HP-41C is turned on. (Flags 21 and 55 are both set if the printer is present and clear if not.)

Data Entry Flags

There are two flags in the HP-41C that are used to detect keyboard data input: the numeric input flag (22) and the ALPHA input flag (23).

Flag 22 is used to detect numeric data input. The HP-41C automatically sets flag 22 when numeric data is entered from the keyboard.

Flag 23 is similar to flag 22 except that it is used to detect ALPHA data input. The calculator sets flag 23 when ALPHA data is entered from the keyboard.

Both flags 22 and 23 are cleared each time you turn the calculator on.

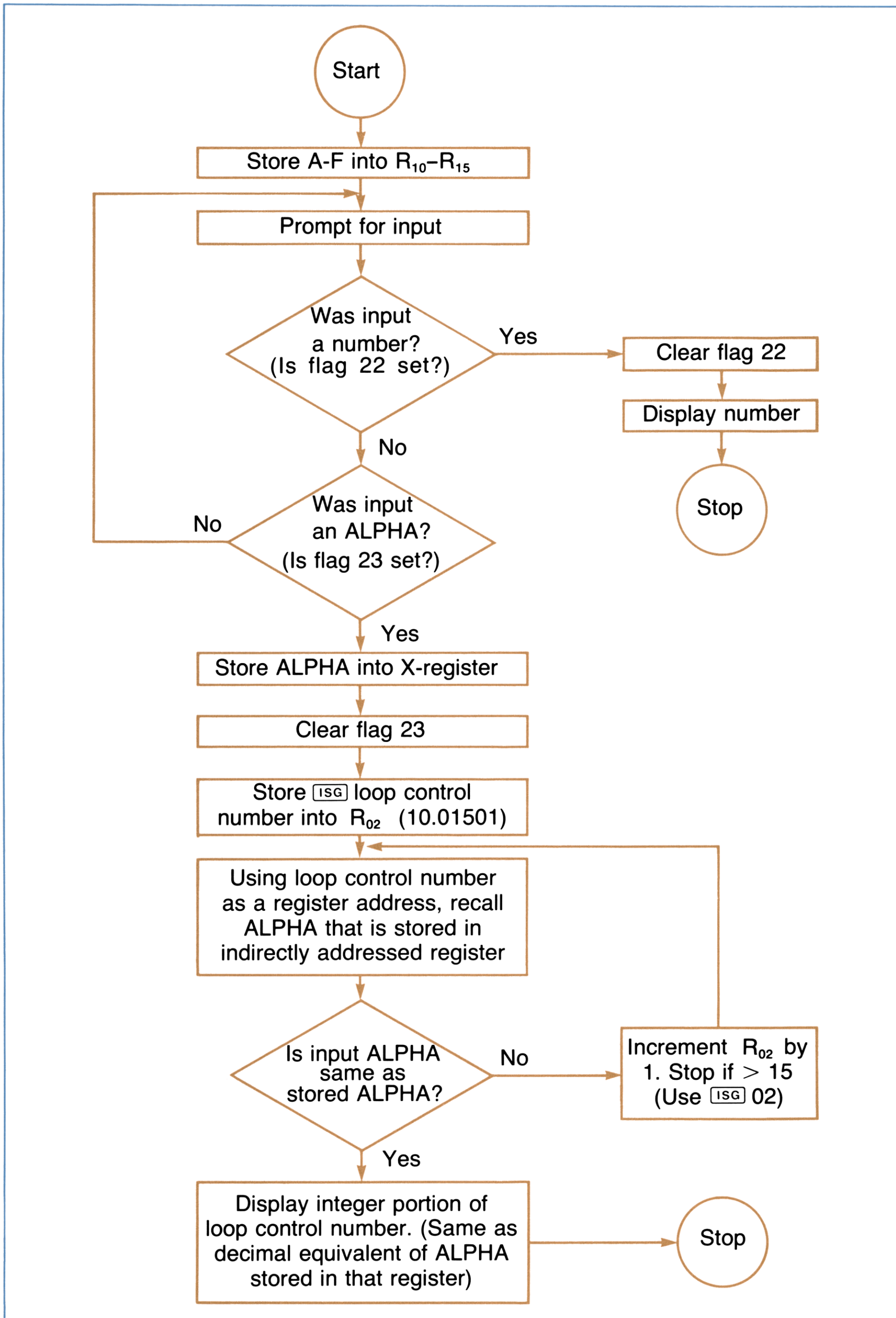
Example: Computer programming student Jill Bitter is a little confused about how to use hexadecimal numbers (base 16). Her teacher suggests that she write a program on her HP-41C to convert hexadecimal numbers to decimal numbers. Jill's first program is below. It converts a single-digit hexadecimal number to decimal.

Hexadecimal/Decimal Equivalents

Hexadecimal	Decimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Jill's program initializes itself by storing the letters A through F in storage registers R_{10} through R_{15}. The program then uses the storage register number to assign a value to the hexadecimal letter that is input.

Here is a flowchart that will help you understand how the program uses the data input flags to determine whether numeric data or ALPHA data was input.


Start
Store A-F into R_{10}–R_{15}
Prompt for input
Was input a number? (Is flag 22 set?)
Yes
Clear flag 22
Display number
Stop
No
Was input an ALPHA? (Is flag 23 set?)
No
Yes
Store ALPHA into X-register
Clear flag 23
Store ISG loop control number into R_{02} (10.01501)
Using loop control number as a register address, recall ALPHA that is stored in indirectly addressed register
Is input ALPHA same as stored ALPHA?
No
Increment R_{02} by 1. Stop if > 15 (Use ISG 02)
Yes
Display integer portion of loop control number. (Same as decimal equivalent of ALPHA stored in that register)
Stop

Keystrokes	Display	
[PRGM]		
■ [GTO] [•] [•]	00 REG 46	
■ [LBL]		
[ALPHA] HEX [ALPHA]	01 LBLTHEX	This initializes the program by storing A through F in R_{10} through R_{15}, respectively.
[ALPHA] A	02TA_	
■ [ASTO] 10	03 ASTO 10	
B	04TB_	
■ [ASTO] 11	05 ASTO 11	
C	06TC_	
■ [ASTO] 12	07 ASTO 12	
D	08TD	
■ [ASTO] 13	09 ASTO 13	
E	10TE_	
■ [ASTO] 14	11 ASTO 14	
F	12TF_	
■ [ASTO] 15 [ALPHA]	13 ASTO 15	
■ [LBL] 01	14 LBL 01	
[ALPHA] INPUT? [ALPHA]	15TINPUT?	
[XEQ]		
[ALPHA] PROMPT [ALPHA]	16 PROMPT	Prompts and stops for input.
[XEQ]		
[ALPHA] FS?C [ALPHA]		
22	17 FS?C 22	Was the input a number?...
■ [RTN]	18 RTN	...yes, displays number and stops.
[XEQ]		
[ALPHA] FC?C [ALPHA]		
23	19 FC?C 23	Was the input an ALPHA?...
■ [GTO] 01	20 GTO 01	...yes, goes to [LBL] 01.
[ALPHA] ■ [ASTO] [•] X [ALPHA]	21 ASTO X	No, places the input into the X-register.
10.01501	22 10.01501_	
[STO] 02	23 STO 02	Stores the loop control number into R_{02}.
■ [LBL] 02	24 LBL 02	
[R↓]	25 RDN	
[RCL] ■ 02	26 RCL IND 02	Recalls the letter in the indirectly addressed register.

Keystrokes	Display	
■ [x=y?]	27 X=Y?	Is input same as stored letter?...
■ [GTO] 03	28 GTO 03	...yes, goes to [LBL] 03.
■ [ISG] 02	29 ISG 02	Increments R_{02}...
■ [GTO] 02	30 GTO 02	...goes to [LBL] 04 if number is less than or equal to 15, and...
■ [RTN]	31 RTN	...stops if it is greater than 15.
■ [LBL] 03	32 LBL 03	
[RCL] 02	33 RCL 02	Displays the integer portion of the loop control number. It is the same as the decimal value of the letter stored in that direct address.
[XEQ]		
[ALPHA] INT [ALPHA]	34 INT	
■ [GTO] [•] [•]	00 REG 35	

Now assign the program to the [Σ+] key for execution in USER mode.

Keystrokes	Display
[PRGM]	0.0000
■ [ASN]	ASN _
[ALPHA] HEX [ALPHA]	ASN HEX _
[Σ+]	ASN HEX 11
	0.0000

Run HEX in USER mode to convert the following single-digit hexadecimal integers to their decimal equivalents: 1, B, 9, F.

Keystrokes	Display	
[USER]	0.0000	
[HEX] ([Σ+])	INPUT?	
1 [R/S]	1.0000	The decimal equivalent of hexadecimal 1.
[HEX]	INPUT?	
[ALPHA] B [ALPHA]		
[R/S]	11.0000	Hexadecimal B equals decimal 11.
[HEX]	INPUT?	
9 [R/S]	9.0000	
[HEX]	INPUT?	
[ALPHA] F [ALPHA]		
[R/S]	15.0000	

Range Error and Error Ignore Flags

Two flags in the HP-41C can be used to control how the calculator reacts to range errors (overflows and underflows) and all operating errors. Flag 24 is the range error ignore flag and flag 25 is the error ignore flag. These flags provide excellent error detection and handling in your programs.

Flags 24 and 25 are both cleared each time you turn the calculator on.

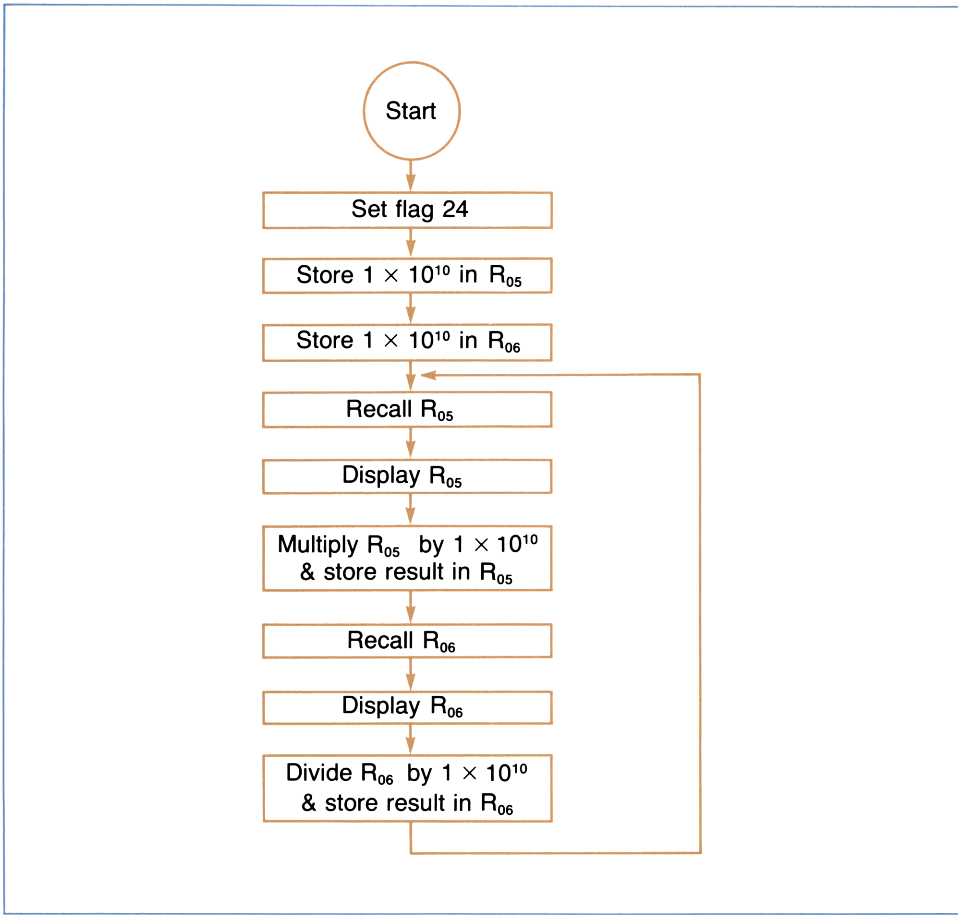
Range Errors

Remember from part I of this handbook that any calculation that exceeds the computation or storage range of the calculator is an error (except in statistics calculations). Normally, when such a calculation is attempted, the HP-41C immediately displays **OUT OF RANGE** and the error-causing function is *not* executed. Flag 24 allows you to ignore these out-of-range errors.

If flag 24 is set, then the HP-41C places $\pm 9.999999999 \times 10^{99}$ into the affected register and execution continues. Note that the range error ignore flag is *not* cleared when the error occurs. Since flag 24 is cleared (automatically) only when you turn the calculator on, you only need to set it one time at the beginning of the program. All subsequent range errors will be ignored by the calculator.

Specifically, a range error is an overflow where a number is generated that exceeds $\pm 9.999999999 \times 10^{99}$. Underflows (numbers closer to zero than $\pm 1 \times 10^{-99}$) do not cause the **OUT OF RANGE** message to be displayed. Zeros are placed into the affected register. Other range errors that can be ignored by flag 24 are listed in appendix E.

For example, the following program demonstrates how flag 24 works. An infinite loop in the program begins with 1×10^{10} and alternately multiplies and divides that number by 1×10^{10}. Each time through the loop, the result from the previous multiply is multiplied by 1×10^{10}, and the result from the previous divide is divided by 1×10^{10}. You can watch as the displayed numbers approach the overflow ($9.999999999 \times 10^{99}$) and the underflow ($0.000000000 \times 10^{00}$). Since flag 24 is set, the overflow error does not cause the program to stop.



Keystrokes	**Display**
PRGM	
GTO . .	*00 REG 46*
LBL	
ALPHA FLOW ALPHA	*01 LBLᵀFLOW*
SF 24	*02 SF 24*
EEX 10	*03 1 E 10*
STO 05	*04 STO 05*
STO 06	*05 STO 06*
LBL 01	*06 LBL 01*
RCL 05	*07 RCL 05*
XEQ	
ALPHA PSE ALPHA	*08 PSE*
EEX 10	*09 1 E 10*

Keystrokes	Display
STO × 05	10 ST*05
RCL 06	11 RCL 06
XEQ	
ALPHA PSE ALPHA	12 PSE
EEX 10	13 1 E 10
STO ÷ 06	14 ST/ 06
■ GTO 01	15 GTO 01
■ GTO • •	00 REG 41

Run the program and watch the numbers as they approach the range over- and underflows.

Keystrokes	Display	
PRGM ■ CLX	0.0000	
XEQ		
ALPHA FLOW ALPHA	1.0000 10	
	0.0000 10	
	1.0000 20	
	1.0000	
	1.0000 30	
	1.0000 -10	
	1.0000 40	
	1.0000 -20	
	1.0000 50	
	1.0000 -30	
	1.0000 60	
	1.0000 -40	
	1.0000 70	
	1.0000 -50	
	1.0000 80	
	1.0000 -60	
	1.0000 90	
	1.0000 -70	
	9.9999 99	The overflow is ignored.
	1.0000 -80	
	9.9999 99	
	1.0000 -90	
	9.9999 99	
	0.0000 00	The underflow.
	9.9999 99	
	.	
	.	
	.	
R/S	0.0000 00	Press and hold R/S to stop the program.

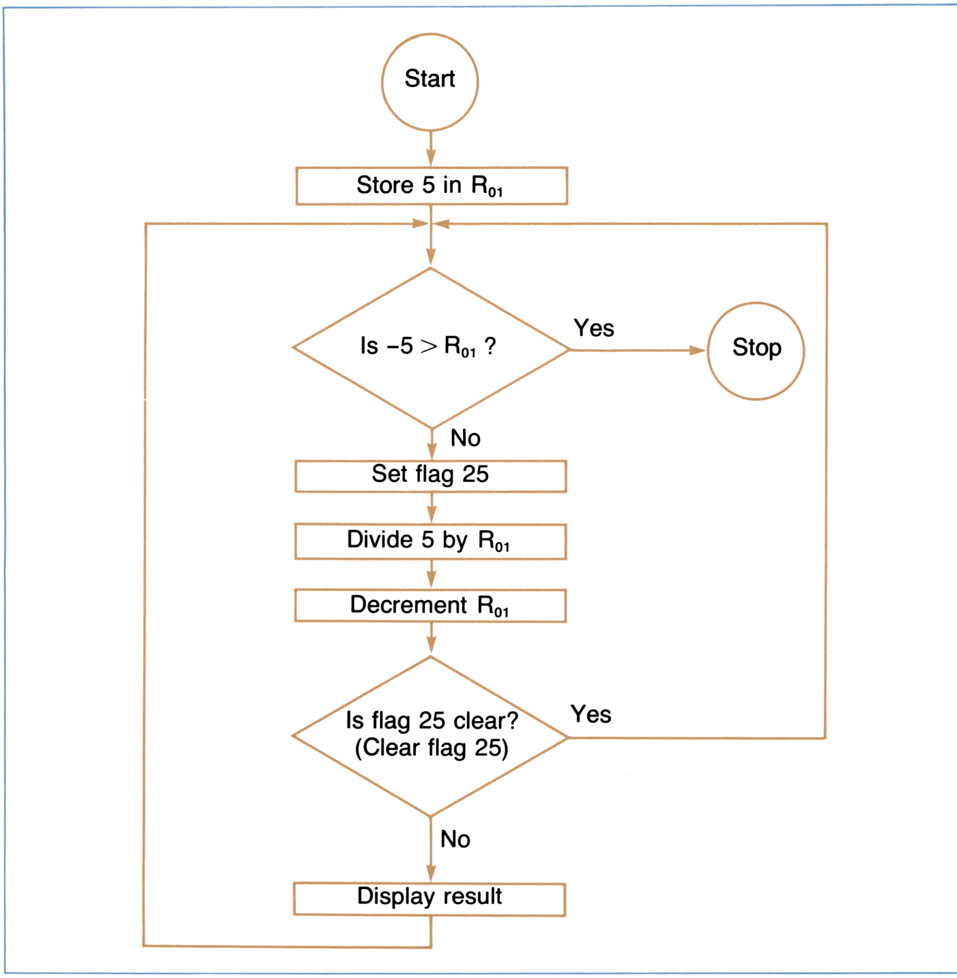
Errors

Normally, the HP-41C halts execution and displays ***DATA ERROR*** when any improper operation (like division by zero) is performed. The HP-41C also halts execution and displays ***OUT OF RANGE*** when a range error occurs. When you set flag 25, however, the HP-41C will ignore a single improper operation. The operation is not performed but execution continues.

Note that when the improper operation is attempted, flag 25 is automatically cleared. Because the HP-41C clears flag 25 when an improper operation is attempted, it is a good idea to set the flag just prior to the line where you suspect an error might occur. You can also test the flag immediately after the suspect line. This allows you to prevent bad data from interrupting your program.

Range errors can be controlled by either flag 24 (range error ignore flag) or flag 25 (the error ignore flag) because range errors are also treated as errors. Flag 24 allows you to continue execution indefinitely when a range error occurs, and flag 25 allows you to detect a range error and take corrective action.

Example: The following program counts from 5 down to –5 and divides 5 by the count number. When the count reaches 0, normally a division by zero would cause the program to stop execution. However, this program uses flag 25 to detect the division by zero and branch around the bad data value, continuing with –1. Here is a flowchart illustrating the program.



Keystrokes	Display
PRGM	
GTO • •	00 REG 46
LBL	
ALPHA ERROR ALPHA	01 LBL^{T}ERROR
5	02 5_
STO 01	03 STO 01
LBL 01	04 LBL 01
RCL 01	05 RCL 01
5 CHS	06 −5_
X>Y?	07 X>Y?
RTN	08 RTN

Keystrokes	Display
5	09 5_
RCL 01	10 RCL 01
■ SF 25	11 SF 25
÷	12 /
1	13 1_
STO − 01	14 ST − 01
XEQ	
ALPHA FC?C ALPHA 25	15 FC?C 25
■ GTO 01	16 GTO 01
x⇄y	17 X<>Y
XEQ	
ALPHA PSE ALPHA	18 PSE
■ GTO 01	19 GTO 01
■ GTO • •	00 REG 41

Now run the program. Notice how the division by zero never appears.

Keystrokes	Display	
PRGM	0.0000	
XEQ		
ALPHA ERROR ALPHA	1.0000	
	1.2500	
	1.6667	
	2.5000	
	5.0000	
	−5.0000	
	−2.5000	
	−1.6667	
	−1.2500	
	−1.0000	
	−5.0000	The program halts showing −5.0000

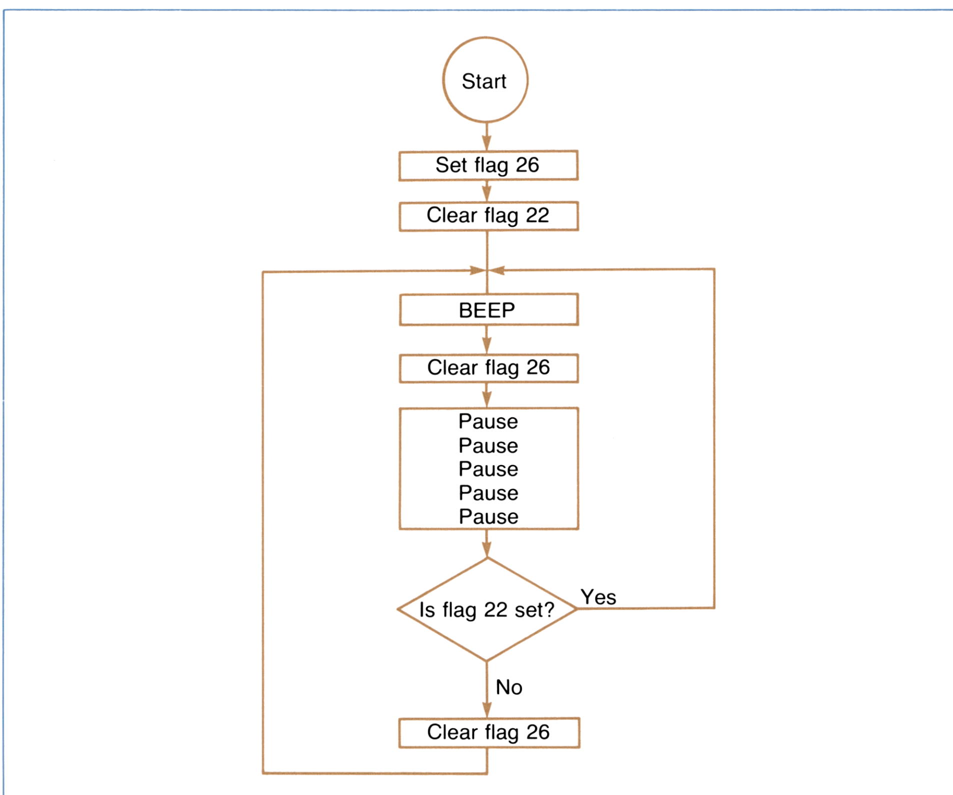
Audio Enable Flag

Flag 26 is used to control the audible tone in the HP-41C. When flag 26 is set, the HP-41C audible tone will operate. When you clear flag 26, the audible tone will not operate.

You can set, clear and test flag 26, just like any of the general or special purpose flags. But you should remember that this flag also controls the operation of the audible tone. Flag 26 is the only user flag that is automatically set (so that the audible tone functions) each time the HP-41C is turned on.

Example: DeDe Daldre has a program in her HP-41C that helps her keep track of her reading speed. In her job with Dul Publishing, she works as a proofreader and she has found that she must proof and mark corrections on one complete line every five seconds in order to keep up with her daily quota of 5760 lines.

At the end of each line, without lifting her eyes from the page, she presses any number key (usually 0). If more than about five seconds go by without her pressing a number key again, the program sounds the audible tone. By placing **PSE** *(pause)* instructions in the program, in combination with the other program instructions, the HP-41C can time approximately the five seconds required in this program. The following flowchart will help you understand the flow of the program. The numeric data entry flag (flag 22) is used to detect the press of a number key, and the audio enable flag (flag 26) is used to control the audible tone.



Before you begin, assign the PSE function to the √x key location so that you can input PSE at the press of a single key in USER mode.

Keystrokes	Display
ASN	ASN_
ALPHA PSE ALPHA	ASN PSE_
√x	0.0000

Now load the program.

Keystrokes	Display	
PRGM		
GTO • •	00 REG 45	
LBL		
ALPHA PROOF ALPHA	01 LBLTPROOF	
CF 26	02 CF 26	Disables the audible tone (flag 26 is initially set).
CF 22	03 CF 22	
LBL 01	04 LBL 01	
BEEP	05 BEEP	Sounds alarm if flag 26 is set, not if flag 26 is clear.
CF 26	06 CF 26	Clears flag 26.
USER		
PSE (√x)	07 PSE	Timing routine.
PSE (√x)	08 PSE	
PSE (√x)	09 PSE	
PSE (√x)	10 PSE	
PSE (√x)	11 PSE	
USER		
XEQ		
ALPHA FS?C ALPHA 22	12 FS?C 22	Tests and clears flag 22.
GTO 01	13 GTO 01	If flag 22 is set (data has been entered), goes to LBL 01.
SF 26	14 SF 26	If flag 22 is clear (data has not been entered), sets flag 26 …
GTO 01	15 GTO 01	… then goes to LBL 01.
GTO • •	00 REG 40	

Now run the program to see if you can keep up with DeDe's proofreading speed. Remember to look at every word in the line before you press the number key. If you miss the timing on a line, go to the next line.

Keystrokes	Display	
PRGM	0.0000	
XEQ		
ALPHA PROOF ALPHA		When program execution begins, PROOF begins timing you for about five seconds. If you press a number key before the timing is over, the tone will not sound. But if you take too long to press the number key, the tone will sound and the program will immediately begin timing again.
R/S	0.0000	Press R/S to halt execution.

USER Mode Flag

This flag (flag 27) is used to place the calculator into and out of USER mode. When flag 27 is set, the HP-41C is placed into USER mode. When cleared, the HP-41C is taken out of USER mode.

You can set, clear and test flag 27 just like a general purpose flag, but keep in mind that this flag also controls USER mode.

The status of flag 27, whether clear or set, is maintained at all times by Continuous Memory, even when the calculator is turned off and on.

Number Display Control Flags

Two flags, the decimal point flag (28) and the digit grouping flag (29) are used to control how numbers appear in the HP-41C display.

The decimal point flag (flag 28) controls the *radix mark* and the *separator mark* in a number. A radix mark is the divider between the integer portion of a number and the fractional portion of a number. The separator mark is the separator between groups of digits in a large number.

In Europe, and many other international locations, the radix mark is the comma and the separator mark is the decimal point. So numbers appear like this: 1.234.567,01. In the U.S. the radix mark is the decimal point and the separator mark is the comma. Numbers appear like this: 1,234,567.01. The decimal point flag (28) allows you to use the radix mark and the separator mark with which you are most accustomed.

When flag 28 is set, the decimal point is the radix and the comma is the separator. Numbers appear like this: 1,234,567.01.

When flag 28 is clear, the comma is the radix and decimal point is the separator. Numbers appear like this: 1.234.567,01.

You can set, clear and test flag 28 just like the general purpose flags. The status of the decimal point flag (28) is preserved at all times. Flag 28 is initially set (decimal point is the radix, comma is the separator).

The second flag that controls how numbers appear in the HP-41C display is the digit grouping flag (flag 29). It controls whether or not a separator is used.

Regardless of which separator mark is specified (refer to flag 28 description, above), you can control whether or not a separator appears in the display. If your preference is for separators, you can specify them. If your preference is no separators, then you can turn them off.

When flag 29 is set, groups of three digits in the integer portion of the number are separated like this: 1,234,567.01 or 1.234.567,01.

When flag 29 is clear, numbers are not separated, like this 1234567.01 or 1234567,01.

The status of flag 29 is preserved at all times. The initial status is set, so numbers will appear like this: 1,234,567.01.

In FIX 0 when both flag 28 and flag 29 are clear, no radix will appear. At any time if there is only one symbol showing in a number, it is always the radix.

HP-41C System Flags

Flags 30 through 55 are all used by the HP-41C system to control the calculator's internal operation. Some have little value to you, and all can only be tested. Following is a listing and short description of each system flag.

Catalog Flag (flag 30). As with all of the system flags, flag 30 can be tested only. It is used internally for the operation of the catalog feature and always tests clear for you.

Peripheral Flags (flags 31 through 35). These flags are used internally for the operation of certain peripheral extensions.

Number of Digits (flags 36 through 39). These four flags, in combination, are used internally to set the number of displayed decimal digits in FIX, SCI and ENG display formats. The number of decimal digits is determined by the following chart.

	Flag status (clear = 0, set = 1)			
No. of digits	36	37	38	39
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Display Format Flags (FIX = flag 40, ENG = flag 41). When flag 40 is set, the HP-41C is in FIX display format (when flag 40 is set, flag 41 is always clear). If flag 41 is set, the calculator is is ENG display format (when flag 41 is set, flag 40 is always clear). When both flags 40 and 41 are clear, the calculator is in SCI format. The number of digits displayed is determined by flags 36 through 39.

Grads Mode Flag (flag 42). If flag 42 is set, the calculator is in GRAD mode (when flag 42 is set, flag 43 is clear).

Radians Mode Flag (flag 43). If flag 43 is set, the HP-41C is in RAD mode (when flag 43 is set, flag 42 is clear).

Continuous On Flag (flag 44). Flag 44 controls whether the HP-41C is in continuous on power mode or not. When set, the HP-41C is in continuous on power mode. When clear, the calculator automatically turns off after 10 minutes of inactivity.

System Data Entry Flag (flag 45). This flag is used internally by the HP-41C in data entry. It always tests clear for you.

Partial Key Sequence (flag 46). This flag is used internally by the HP-41C in function execution. It always tests clear for you.

Shift Set Flag (flag 47). Flag 47 is used internally in shifted operations and always tests clear for you.

ALPHA Mode Flag (flag 48). This flag is used for ALPHA mode control. When the HP-41C is in ALPHA mode, flag 48 is set, when not, flag 48 is clear.

Low Battery Flag (flag 49). The low battery flag is used to indicate low battery power. When set, power is low. When clear, power is sufficient. Refer to Batteries, appendix B, for battery replacement instructions. Remember that when battery power is low, the BAT annunciator in the display appears.

Message Flag (flag 50). When set, the display contains some message. When clear, the display contains the default display (ALPHA- or X-register).

SST Flag (flag 51). Flag 51 is used internally for single-line program execution and always tests clear for you.

PRGM Mode Flag (flag 52). Flag 52 is used to control PRGM mode. It always tests clear for you.

I/O Flag (flag 53). This flag is used to determine if some peripheral extension is ready for I/O. When set, the extension is ready. When clear the device is not ready for I/O activity.

Pause Flag (flag 54). When flag 54 is set, a user program PSE is in progress. When clear, a pause is not in progress.

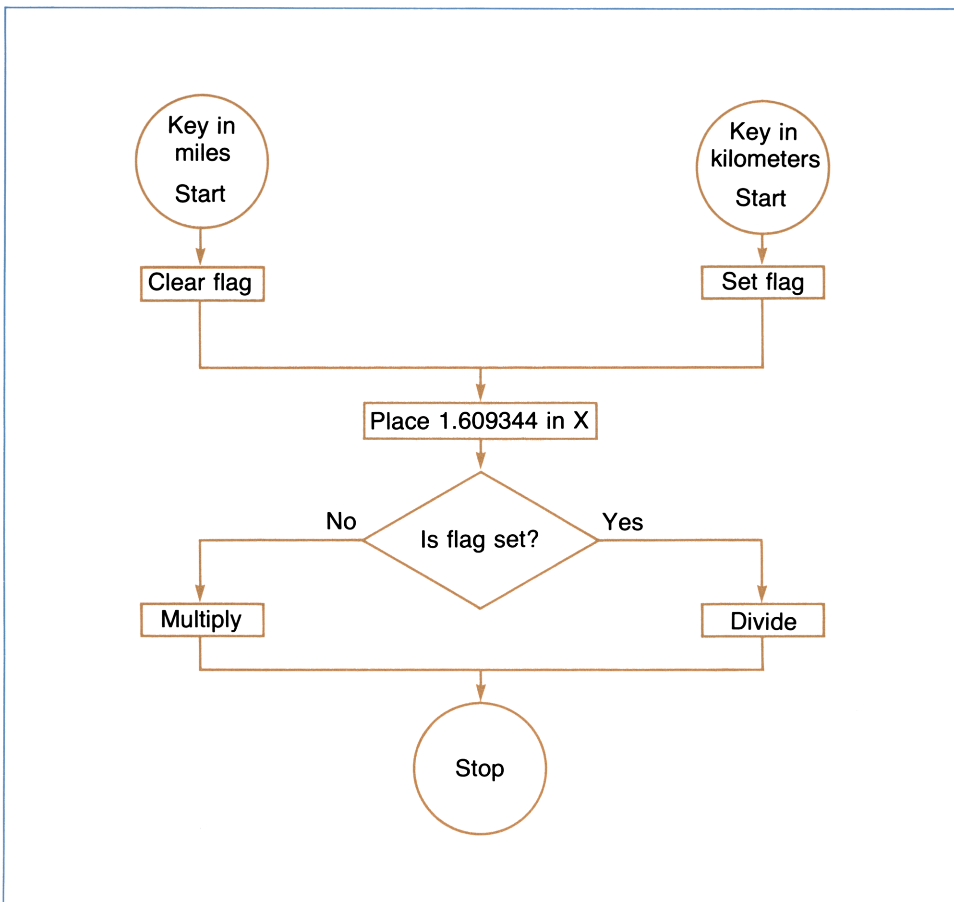
Printer Existence Flag (flag 55). This is used to indicate if the standard HP-41C printer is attached to the HP-41C. When set, a printer is attached. When clear, no printer is present. Flag 55 works in conjunction with the printer enable flag (flag 21).

Problems:

1. One mile is equal to 1.609344 kilometers. Use the flowchart below to create and load a program that will permit you to key in distances in either miles (LBL MILE) or kilometers (LBL KILO). Using a flag and a subroutine, either multiply or divide to convert from one unit of measure to the other. (Hint: 1/x × yields the same as ÷.)

   Run the program to convert 187,000 miles to kilometers; to convert 1.2701 kilometers to miles.

   (Answers: 300,947.3280 kilometers; 0.7892 miles.)



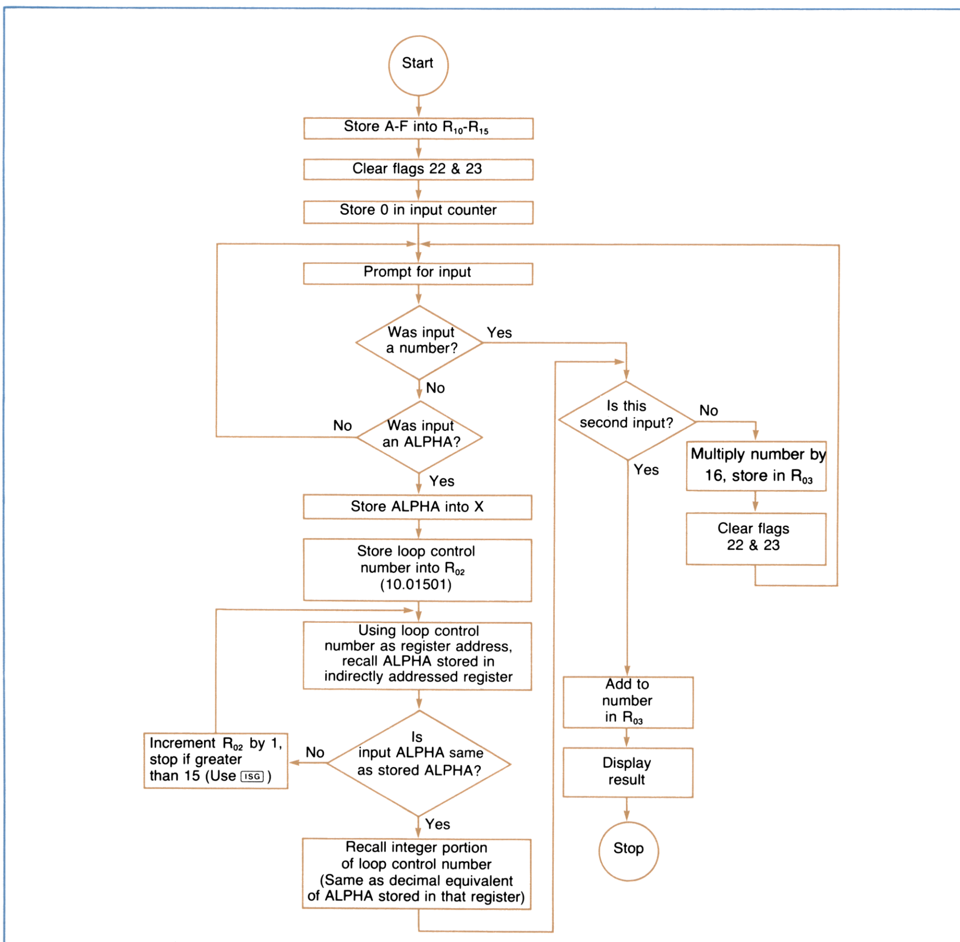
2. Rewrite the timing program you input on page 229 so that it counts the number of times the flag was set (successful timings). Store that number in a register so you can check the total later.

3. Rewrite the timing program above so that it also counts the number of times the flag was cleared (unsuccessful timings). Again, store that number in a register for later reference.

4. The example on page 220 converts single-digit hexadecimal numbers to their decimal equivalents. Using the following flowchart, and the concepts in the example problem, write a new program that converts two-digit hexadecimals to decimals. A solution to this problem is given following the flowchart. Before looking at the solution, try writing your own program from the flowchart.

   Run the program and convert 4F, 2B, 13, AA to decimal equivalents. The program prompts you for one digit of the number at a time (e.g., to convert 4F, when the program prompts you, first key in 4 [R/S], then [ALPHA] F [ALPHA] [R/S]).

   (Answers: 79; 43; 19; 170)



Here is a solution to problem 4.

```
00
01 LBLTHEX
02TA
03 ASTO 10
04TB
05 ASTO 11
06TC
07 ASTO 12
08TD
09 ASTO 13
10TE
11 ASTO 14
12TF
13 ASTO 15
14 0
15 STO 00
16 LBL 01
17 1
18 ST+ 00
19 CF 22
20 CF 23
21TINPUT?
22 PROMPT
23 FS? 22
24 GTO 02
25 FS? 23
26 GTO 04
27 GTO 01
28 LBL 04
29 ASTO X
30 10.01501
31 STO 02
32 LBL 05
33 RDN
34 RCL IND 02
35 X=Y?
36 GTO 06
37 ISG 02
38 GTO 05
39 RTN
40 LBL 06
41 RCL 02
42 INT
43 LBL 02
44 1
45 RCL 00
46 X>Y?
47 GTO 03
48 RDN
49 RDN
50 16
51 *
52 STO 03
53 GTO 01
54 LBL 03
55 RDN
56 RDN
57 ST+ 03
58 RCL 03
59 END
```

Congratulations!

You have just completed the *HP-41C Owner's Handbook and Programming Guide*. You have certainly noticed that programming on the HP-41C is simple, and even fun. Yet the capability of the system is astounding. Your programming expertise will increase as you continue to use your HP-41C. And you will find it an easy matter to completely customize your HP-41C.

The appendices following this section will provide you with more specific information about the HP-41C.

(This page intentionally left blank.)

Appendix A

Accessories

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 accessories for the HP-41C system.

Standard Accessories

Your HP-41C Comes Complete With These Standard Accessories:

	HP Part Number
Four Size N Batteries (ready to be installed).	—
HP-41C Owner's Handbook and Programming Guide.	00041-90001
HP-41C Quick Reference Guide.	00041-90002
HP-41C Application Book.	00041-90018
One blank Keyboard Overlay.	—
Soft Carrying Case.	—
One Module/Overlay Holder	—
One set of Function Labels	—
One pre-printed Keyboard Overlay	—

HP-41C Optional Accessories

HP 82106A 64-Register Memory Modules.	82106A
Application Modules. (Refer to the HP-41C Accessory Brochure for titles.)	—
HP 82143A Thermal Printer.	82143A
HP 82104A Card Reader.	82104A
AC Adapter.	
Printer Paper. (For HP 82143 Printer.)	82045A
Magnetic Cards. (For HP 82104A Card Reader.)	
40 Blank Cards With Holder	00097-13141
120 Blank Cards With Holders	00097-13143
1000 Blank Cards	00097-13206

To order additional standard or optional accessories, or system extensions for your HP-41C, or for information about new optional accessories and extensions, see your nearest dealer or fill out an Accessory Order Form and return it with check, money order, Master Charge or VISA numbers to:

HEWLETT-PACKARD
CORVALLIS DIVISION
P.O. BOX 3499
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.

Appendix B

Maintenance and Service

Your Hewlett-Packard Calculator

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

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

Hewlett-Packard owner's handbooks are carefully prepared by professional writers and have won international awards for writing excellence.

Calculator Care

Designed to be durable and dependable, your HP-41C requires virtually no attention to ensure proper operation. All you need to do is:

1. Replace the batteries when the ***BAT*** annunciator in the display appears (refer to Batteries).
2. Make sure that you keep the caps on the input/output receptacles (ports) in place whenever a module or other plug-in accessory is not plugged into a port. These caps prevent the contacts inside the ports from becoming contaminated, which could lead to improper operation.

> **CAUTION**
>
> Do not insert your fingers or any objects other than an HP module or plug-in accessory into any port. To do so could alter the Continuous Memory or could even damage the port or the calculator.

Temperature Specifications

■ Operating:	0° to 45° C	32° to 113° F
■ Storage:	−20° to 65° C	−4° to 149° F

Plug-In Extensions

CAUTION

Always turn the HP-41C off before inserting or removing any plug-in extensions or accessories. Failure to turn the HP-41C off could damage both the calculator and the accessory.

All plug-in extensions should be handled with care.

1. Keep the contact area free of obstructions. Should the contacts become dirty, carefully brush or blow the dirt out of the contact area. Do not use any liquid to clean the contacts or extensions.
2. Store the extensions in a clean dry place. Do not place plug-in extensions in a pocket unless they are protected in their case. Static electricity could damage the extension.
3. Always turn the HP-41C off before inserting or removing any plug-in extension. Failure to do so could damage both the calculator and the extension.

Batteries

Because your HP-41C uses so little power, disposable batteries will provide many hours of calculator operation. The total number of operating hours depends upon how fresh the batteries were when you purchased and installed them, and how much you use peripherals. When you use peripherals that draw power from the HP-41C batteries (such as the HP 82104A Card Reader or the HP 82153A Optical Wand), total battery life will be reduced considerably. If the *BAT* (low power) annunciator turns on (or the display shows *LOW BATTERY*) while a peripheral is in use, turn the HP-41C and the peripheral off, disconnect the peripheral from the HP-41C, and turn the calculator back on again. The batteries will then power the calculator without peripherals for a significant amount of time before the *BAT* annunciator turns on again. If you use peripherals frequently, we recommend that you power your HP-41C with an HP 82120A Rechargeable Battery Pack. Refer to the instruction sheet for the rechargeable battery pack for installation and use instructions.

Everready E90	Mallory MN9100	UCAR E90
National AM5(s)	Panasonic AM5(s)	VARTA 7245

Disposable batteries should be installed as described under Replacing the Batteries. Use only the following alkaline batteries or the HP 82120A Rechargeable Battery Pack in your HP-41C:

These batteries, like those originally supplied with your HP-41C, are not rechargeable.

WARNING

Do not attempt to recharge the batteries. Do not store batteries near a source of high heat or dispose of them in fire. Doing so may cause them to leak or explode.

Replacing the Batteries

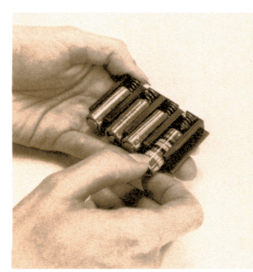
The Continuous Memory of your HP-41C will normally be preserved for about 30 to 60 seconds while the batteries are out of the calculator. However, you must turn the calculator off *before* removing the batteries in order to preserve Continuous Memory. This gives you ample time to replace the batteries with new ones. Leaving batteries out of the calculator for extended periods will result in loss of the information in Continuous Memory.

To replace the batteries, use the following procedure (you may want to read through the entire battery replacement procedure prior to replacing the batteries):

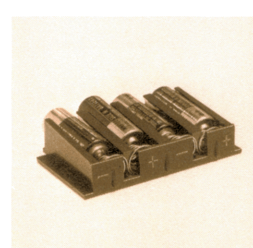
1. Turn the calculator off.

2. Turn the calculator over in your hand and push up on the lip on the battery holder as shown in the photograph.

3. Remove the batteries from the battery holder, making sure you do not mix them up with the new batteries.

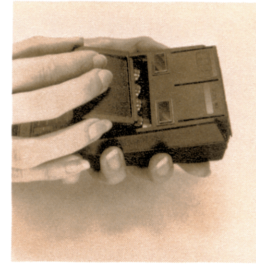


4. Look at the polarity marks on the end of the battery holder. It shows how the batteries should be inserted into the battery holder. Insert the new batteries, and *carefully note the position of each battery*. If any of the batteries are inserted wrong, the calculator will not turn on.



5. Insert the battery holder into the calculator such that the exposed ends of the batteries are pointing toward the input/output ports.

6. Push the upper edge of the battery holder into the calculator until it goes no further. Then snap the lower edge of the holder into place.



If any of the batteries are inserted incorrectly, the calculator may not turn on. If, when you insert the new batteries the calculator fails to turn on, immediately remove the battery holder and check the position of the batteries. The calculator cannot be damaged by inserting the batteries wrong; it simply will not function.

Service

Using state-of-the-art technology, the HP-41C Continuous Memory circuits operate continuously—even while the calculator is turned off. Because these circuits are always drawing very low power from the batteries, they are susceptible to disruption at all times. Disruption can be caused by *inserting or removing plug-in modules or peripherals while the power is turned on; electrostatic discharge to the unit; strong magnetic fields; plugging devices into the HP-41C that are not supported by Hewlett-Packard for use with the HP-41C;* or other conditions that can traumatize the calculator.

Of course, all causes of disruption should be avoided, but should disruption occur, the most common symptom is a loss of keyboard control of the calculator. The HP-41C has been designed to allow recovery from these conditions. The procedure for resetting the calculator is to simply remove the battery pack and replace it again immediately. This will reset the HP-41C without causing a **MEMORY LOST** condition (unless the trauma itself was great enough to cause a **MEMORY LOST** condition). After several attempts, if this procedure fails to reset the calculator, work through the service procedure below.

If the display blanks out, or the calculator will not respond to keystrokes, do the following:

1. Ensure that the batteries are fresh, are properly installed, and that the battery contacts are not dirty.
2. Turn the calculator off then back on. If the calculator does not respond, continue on to step 3.
3. While holding down the ⬅ key, turn the calculator on. This is a "master clear" and the entire calculator will be cleared. If the calculator does not respond, continue on to step 4.
4. Remove the batteries and let the continuous memory in the calculator discharge over night. When you reinstall the batteries and turn the calculator on, if the display shows **MEMORY LOST**, you know that the calculator has been cleared.
5. If the calculator still does not respond, service is required (refer to Limited One-Year Warranty).

Repair Policy

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

Limited One-Year Warranty

What We Will Do

The HP-41C and its accessories *(except software, the batteries, and damage caused by the batteries)* are warranted by Hewlett-Packard against defects in materials and workmanship for one year from the date of original purchase. If you sell your calculator or give it as a gift, the warranty is automatically transferred to the new owner and remains in effect for the original one-year period. During the warranty period we will repair or, at our option, replace at no charge a product that proves to be defective provided that you return the product, shipped prepaid, to a Hewlett-Packard repair center.

What Is Not Covered

The batteries or damage caused by the batteries are not covered by this warranty. However, certain battery manufacturers may arrange for the repair of the calculator if it is damaged by the batteries. Contact the battery manufacturer first if your calculator has been damaged by the batteries.

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

Hewlett-Packard shall have no obligation to modify or update products once sold.

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

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

Warranty Information Toll-Free Number

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

How to Obtain Repair Service

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

The Hewlett-Packard United States repair center for handheld and portable printing calculators is located in Corvallis, Oregon. The mailing address is:

HEWLETT-PACKARD COMPANY
CORVALLIS DIVISION SERVICE DEPT.
P. O. Box 999/1000 N.E. CIRCLE BLVD.
CORVALLIS, OREGON 97330

Note: Not all Hewlett-Packard repair centers offer service for all models of HP calculators. However, if you bought your calculator from an authorized Hewlett-Packard dealer, you can be sure that service is available in the country where you bought your calculator. A list of repair centers for other countries may be obtained by writing to the above address.

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

HEWLETT-PACKARD COMPANY
SERVICE DEPT.
1000 N.E. CIRCLE BOULEVARD
CORVALLIS, OREGON 97330
U.S.A.

All shipping, reimportation and duty arrangements are your responsibility.

Shipping Instructions

Do not return any batteries in or with the calculator. Please refer to Battery Damage on page 245.

Should your HP-41C require service, the calculator should be returned with the following items:

1. A completed Service Card, including a description of the problem.
2. A sales slip or other proof of purchase (if the one-year warranty period has not expired).
3. Whether the unit is under warranty or not, it is your responsibility to pay shipping charges for delivery to the Hewlett-Packard repair center.

4. After warranty repairs are completed, the repair center returns the unit with postage prepaid.
5. On out-of-warranty repairs, unit will not be repaired until payment method has been established. (Refer to HP-41C System Service Card.)

The calculator, Service Card, and (if required) the proof of purchase should be packaged in its original shipping case or other *adequate protective packaging* to prevent in-transit damage. Such damage is not covered by the one-year limited warranty; Hewlett-Packard suggests that you insure the shipment to the repair center. The packaged calculator should be shipped to the address shown on the Service Card.

Battery Damage

Do not return any batteries in or with the calculator. The batteries or damage caused by the batteries are not covered by the one-year limited warranty.

If your HP-41C is damaged by battery leakage you should first contact the battery manufacturer for warranty information. Some battery manufacturers may repair the calculator if it has been damaged by leaking batteries. If the battery manufacturer warrants against battery damage, you should deal directly with that manufacturer for repairs. If the battery manufacturer does not warranty against battery damage, you should send the calculator to the Hewlett-Packard for repair. Whether the calculator is under warranty or not, there will be a charge for repairs made by Hewlett-Packard when the calculator has been damaged by the batteries. To avoid this charge, contact the battery manufacturer first when your calculator has been damaged by the batteries.

Programming and Applications

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

HEWLETT-PACKARD
CORVALLIS DIVISION CUSTOMER SUPPORT
1000 N.E. CIRCLE BOULEVARD
CORVALLIS, OR 97330

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

"All information stated within is submitted to Hewlett-Packard Company without any confidentiality or obligation. I am submitting this information freely to Hewlett-Packard for their disposition."

Further Information

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

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

Appendix C

Stack Lift Conditions and Termination of Keyboard Entry

Your HP-41C has been designed to operate in a natural, friendly manner. As you have seen while you worked through this handbook, you are seldom required to think about the operation of the automatic memory stack or the display—you simply work through calculations in the same way you would with pencil and paper, performing one operation at a time. There may be occasions, however, such as when you are creating a program, when you want to know the effect of a certain operation upon the display of the stack.

Termination of Digit and ALPHA String Entry

Except for those operations used for digit entry ([•], [CHS], [EEX], [←], [USER], [ALPHA], [■]), all operations in the HP-41C 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. The new number will be written over the number in the X-register. However, depending on the particular operation, the stack may first be "lifted" so that the contents of the X-register are copied into the Y-register before the new number is keyed into the X-register.

ALPHA entry is terminated by all other functions except [ARCL]. To continue building an ALPHA string after ALPHA entry is terminated, simply press [■] [APPEND].

Stack Lift

Operations in the HP-41C are of three types with respect to their effect on the stack. Most operations *enable* the stack lift. A few operations *disable* the stack lift, and a few others are *neutral*.

Enabling Operations

All operations on the HP-41C other than those listed below (under Disabling Operations and Neutral Operations) *enable* the stack lift. If you key in a number immediately following an enabling operation, the stack is lifted and the number is entered into the display.

Disabling Operations

If you key in a number immediately following a disabling operation, the stack is *not* lifted. Therefore, the contents of the X-register are not copied into the Y-register before the new number is keyed into the X-register. The disabling operations are:

[ENTER↑] [CLX] [Σ+] [Σ−]

Neutral Operations

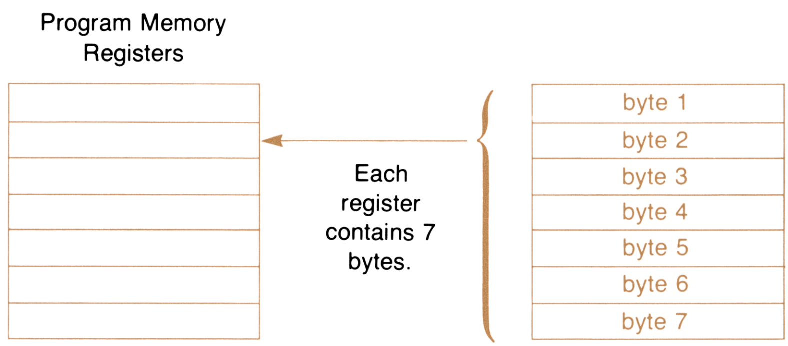
Neutral operations are those that do not alter the status of the stack lift, so that whether or not the stack is lifted depends upon the previous operation. Note that [CHS] and [EEX] are neutral only during digit entry. When pressed any other time, [CHS] and [EEX] enable the stack lift. The neutral operations are:

[PRGM] [ALPHA] [ON] [CHS] [EEX] [USER] [←] [■]

Appendix D

Program Memory Storage Requirements and LAST X Operations

Program memory in the HP-41C is structured in registers. Each register can hold up to seven lines of program instructions. In other words, each register in program memory is divided into seven parts. One of these parts is called a *byte* of program memory.



Most operations on the HP-41C require only one byte of program memory for storage as a line in a program, but some require two or even more bytes. For your reference, the number of bytes required for storing each programmable HP-41C function is listed in the table beginning below. The operations are listed in alphabetical order of their name.

Note that ALPHA characters require one byte each plus an additional byte for the string when stored in program memory. So the string CIRCLE would require seven bytes of program memory. Each digit in a number requires one byte when stored in program memory. The decimal point in a number also requires one byte. The number 28.741 would require six bytes of program memory.

Also indicated for each operation is whether the contents of the X-register are copied into the LAST X register before the operation is performed.

Function	Storage Requirement (Bytes)	Saves x in LAST X
ABS	1	Yes
Σ-	1	Yes
Σ+	1	Yes
+	1	Yes
ADV	1	No

Function	Storage Requirement (Bytes)	Saves x in LAST X
ALPHA strings, n characters long (1 byte each character plus 1 byte for the string).	$n + 1$	No
AOFF	1	No
AON	1	No
ARCL	2	No
ASHF	1	No
Assignments.	*	No
ASTO	2	No
AVIEW	1	No
10^x, 10↑X	1	Yes
e^x, E↑X	1	Yes
COS^{-1}, ACOS	1	Yes
SIN^{-1}, ASIN	1	Yes
TAN^{-1}, ATAN	1	Yes
BEEP	1	No
CHS	1	No
CLRG	1	No
CLA	1	No
CLD	1	No
CF	2	No
CLST	1	No
CLΣ	1	No
CLx, CLX	1	No
COS	1	Yes
DEC	1	Yes
DSE	2	No
DEG	1	No
D-R	1	Yes
÷	1	Yes
END	3	No
ENG	2	No
EEX	1	No
ENTER↑	1	No
x≷y, X<>Y	1	No
X<>	2	No
XEQ (ALPHA, add 1 byte for each ALPHA in name.)	2	No
XEQ (indirect)	2	No
XEQ (numeric)	3	No
y^x, Y↑X	1	Yes
E↑X-1	1	Yes

Function	Storage Requirement (Bytes)	Saves x in LAST X
FACT	1	Yes
FIX	2	No
FC?	2	No
FC?C	2	No
FS?	2	No
FS?C	2	No
FRC	1	Yes
GTO (00 through 14)	2	No
GTO (15 through 99)	3	No
GTO (ALPHA, add 1 byte for each ALPHA in name.)	2	No
GTO (indirect)	2	No
GRAD	1	No
HMS	1	Yes
HMS+	1	Yes
HMS−	1	Yes
HR	1	Yes
ISG	2	No
INT	1	Yes
LBL (00 through 14)	1	No
LBL (15 through 99)	2	No
LBL (ALPHA, add 1 byte for each ALPHA in name.)	4	No
LOG	1	Yes
LN	1	Yes
LN1+X	1	Yes
LASTX, LASTX	1	No
MEAN	1	Yes
MOD	1	Yes
×	1	Yes
OCT	1	Yes
PSE	1	No
%	1	Yes
%CH	1	Yes
π, PI	1	No
P−R	1	Yes
PROMPT	1	No
OFF	1	No
RAD	1	No
R−D	1	Yes
RCL (00 through 15)	1	No
RCL (16 through 99)	2	No

Function	Storage Requirement (Bytes)	Saves x in LAST X
RCL (indirect)	2	No
1/x, 1/X	1	Yes
R-P	1	Yes
RTN	1	No
R↓, RDN	1	No
R↑	1	No
RND	1	Yes
SF	2	No
SCI	2	No
SIGN	1	Yes
SIN	1	Yes
x^2, X↑2	1	Yes
$\sqrt{x}$, SQRT	1	Yes
SDEV	1	Yes
ΣREG	2	No
STOP	1	No
STO (00 through 15)	1	No
STO (16 through 99)	2	No
STO (indirect)	2	No
STO +	2	No
STO ÷	2	No
STO ×	2	No
STO −	2	No
−	1	Yes
TAN	1	Yes
TONE	2	No
VIEW	2	No
x=y?, X=Y?	1	No
x=0?, X=0?	1	No
x>y?, X>Y?	1	No
X>0?	1	No
X<Y?	1	No
X<0?	1	No
x≤y?, X<=Y?	1	No
X<=0?	1	No
X≠Y?	1	No
X≠0?	1	No
0 through 9	1	No
•	1	No

* Assignments of standard HP-41C functions to key locations consume one register (seven bytes) for each odd-numbered assignment made. For example, the first assignment made consumes one register, the second assignment consumes no additional space, the third assignment consumes another full register, the fourth consumes no additional space, and so on. Assignments of programs that you have written and stored into program memory do not require any additional space; the assignment is stored with that program's label.

(This page intentionally left blank.)

Appendix E

Messages and Errors

Following is a listing of all messages and errors that you might see in the HP-41C display.

Display	Meaning
ALPHA DATA	**The HP-41C attempted to perform a numeric operation, such as addition or subtraction, on non numeric data, or on an ALPHA string.**
DATA ERROR	The HP-41C attempted to perform a meaningless operation. These errors are:

[÷]	where $x=0$.
[y^x]	where $y=0$ and $x \leq 0$, or 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 $x \leq 0$.
[LN1+X]	where $x \leq -1$.
[COS^{-1}]	where $\|x\| > 1$.
[SIN^{-1}]	where $\|x\| > 1$.
[STO] [÷]	where $x=0$.
[TONE]	where $\|x\| \geq 10$ or $x < 0$.
[MEAN]	where $n=0$.
[OCT]	where $\|x\| > 1073741823$ (decimal), or x is non-integer.
[DEC]	where x contains an ALPHA, 8 or 9, or x is non-integer.
[%CH]	where $y=0$.
[FIX], [SCI], [ENG]	where absolute value of digits is ≥ 10 or is non-integer.
[FACT]	where $x < 0$ or x is non-integer.

MEMORY LOST	The continuous memory of the calculator has been cleared.
NONEXISTENT	The HP-41C has attempted to use a register that does not exist or is not currently allocated as a storage register.

An attempt was made to [ASN] or [XEQ] a function that does not exist.

An attempt was made to [ASN], [GTO], or [XEQ] an ALPHA or numeric label that does not exist.

An attempt was made to [GTO] a line number that does not exist.

An attempt was made to execute a specific print function when the printer was not connected to the system.

NULL Keystroke was nullified by holding the key down for longer than about a half second.

PRIVATE Refer to the owner's handbook provided with the HP 82104A Card Reader.

An attempt was made to view a private program.

OUT OF RANGE A number has exceeded the computational or storage capability of the HP-41C.

Overflow = ±9.999999999 99

[SDEV] where the standard deviation of x$\left(S_x=\sqrt{M/[n(n-1)]}\right)$ or y $\left(S_y=\sqrt{N/[n(n-1)]}\right)$ results in division by zero or the square root of a negative number. $\left(M=n\Sigma x^2-(\Sigma x)^2;\ N=n\Sigma y^2-(\Sigma y)^2.\right)$

[FACT] where $x>69$.

PACKING Program memory is being packed.

TRY AGAIN As a result of a packing operation, the last keystroke sequence must be repeated. This could be an [XEQ], [ASN], [GTO] [•] [•], or when an attempt is made to insert an instruction into a program.

YES The answer to flag test when test is true. Also the answer to conditionals when relationship between x and 0 or y is true.

NO The answer to flag test when test is false. Also the answer to conditionals when relationship between x and 0 or y is false.

RAM An attempt was made to [COPY] a program in RAM (random access memory—a memory module, or internal memory) to RAM.

ROM An attempt was made to [DEL], [CLP], [←], or insert into a program that is currently in ROM (read only memory—an application module).

Appendix F

HP-41C Extensions

The system capabilities of the HP-41C can be greatly expanded by connecting it to one or more peripheral devices. Available as system extensions, these devices enable you to customize your computational system to suit your particular requirements. You can supplement the standard features of the basic HP-41C with:

Memory Modules for increased program and data storage capacity.

Magnetic card input and output.

Printer output.

Extensive applications libraries.

Input and output through other peripheral devices.

Four input/output (I/O) ports are provided on the top of the HP-41C for interfacing with these devices. A detailed description of capabilities and operation is provided with each device. But to give you a feel for the remarkable power you can achieve by adding to your basic HP-41C calculator, lets look briefly at some of the devices available.

CAUTION

Always turn the HP-41C off before inserting or removing any plug-in extensions or accessories! Failure to turn the HP-41C off could damage both the extension and the calculator.

HP 82106A Memory Module

With up to 63 registers of program memory or 63 registers of data storage, or any combination, the basic HP-41C can swallow a whale of a computational task. But suppose your application requires even more program or data storage capacity. To meet your needs, Hewlett-Packard developed the HP 82106A Memory Module. Each module contains an additional 64 registers that can be allocated as program memory or storage registers, or any combination. You can add four memory modules to your HP-41C system, providing you with a whopping 319 registers. (That's 1000 to 2000 lines of program memory.)

Just as the internal memory of the HP-41C, the additional memory contained in each memory module can be allocated in various combinations between program and data storage. And all of the additional memory, just like the internal memory of the HP-41C, is Continuous Memory. As long as the memory module is plugged into the HP-41C, its contents are preserved for your later use, even while the HP-41C is turned off.

HP 82104A Card Reader

The HP-41C is so easy to program—and the resulting programs so powerful and versatile—that you'll undoubtedly be inspired to write specialized programs for later use. When your programming output exceeds the sizeable capacity of the continuous memory in the HP-41C—or the even larger capacity with optional memory modules—you can permanently store your programs on magnetic cards using the HP 82104A Card Reader.

The HP-41C allows you to specify a single program you wish to record from its continuous memory onto a magnetic card. Each card can contain up to 32 registers of program instructions or 32 storage registers. A program or group of registers need not be limited in length to the capacity of a single card, though; it can be segmented among as many cards as necessary. You don't have to figure out whether more than one card is required for reading and writing; the HP-41C does that for you automatically, then *tells* you by displaying a message.

The HP 82104A Card Reader will even record any key reassignments that are made to run the recorded programs. So all you do is set the HP-41C to USER mode, read in the card or cards and begin. And if you would like to assure the "privacy" and "security" of your recorded programs, you can instruct the card reader to record a card so that the program on that card can only be executed and *not* viewed or altered (under normal operating circumstances).

With an HP-41C and the HP 82104A Card Reader, you are not limited to reading programs or data on magnetic cards that you have recorded yourself. The HP-41C has been specifically designed to accept a program or data on a magnetic card recorded on an HP-67 or HP-97. This will allow you to utilize the vast number of specialized programs available from the HP-67/HP-97 User's Library.

HP 82143A Printer

For a permanent record of calculation results, or for assistance in checking or editing long programs, you can connect an HP 82143A Printer to your HP-41C. Powered by its own set of batteries, it prints alphanumeric characters quietly and efficiently.

The printer can also be set to provide you automatically with valuable diagnostic information when creating or running a program. You can obtain a printed record of the program line number and function name when creating a program. And when executing a program or series of manual keystrokes, the printer can provide a record of the numbers keyed in, functions performed, and answers calculated.

Hewlett-Packard Application Modules

If you're a specialist interested in preprogrammed solutions for problems in a specific field, an HP application module can greatly enhance the usefulness of your HP-41C. Available in a variety of disciplines, HP application modules each contain a number of professionally developed programs. These modules quickly transform your HP-41C into a special-purpose machine designed to solve complex problems in your field at the touch of a few keys.

Up to four application modules can be plugged into the I/O ports on the HP-41C. While a module is plugged in, the names of all programs contained in the module can be displayed by pressing CATALOG 2.

Appendix G

Advanced Programming and Operation

There are several features on the HP-41C that offer significant power and convenience in the operation of the calculator. As you become more interested in the HP-41C and how it works, you may wish to know more specifically how some features work.

Label Searching

Earlier in this handbook it was mentioned that the HP-41C could remember the location of most labels in program memory. More specifically, the HP-41C has been designed to remember the location of *all* labels depending on their location in a program and how they are used. The calculator can only remember a numeric label location *after* the first execution of that label. Subsequent branches to that label are much faster because the HP-41C does not need to search (in most cases).

Labels 00 through 14 are called short form labels. They use only a single byte in program memory (there are seven bytes per register). When a program branches to LBL 00 through LBL 14 using a GTO instruction, the calculator can remember the location of these labels if they are located 112 bytes before or after the GTO instruction. If the short form label is beyond 112 bytes from the GTO, the calculator must search sequentially for that label. So if you are concerned about the speed of execution, you should examine your program and determine the location of branches and corresponding labels.

Labels 15 through 99, on the other hand, are not short form labels. They require two bytes in program memory. However, the location of these labels is always remembered by the calculator, *regardless* of their location in a program.

The location of all numeric labels (LBL 00 through LBL 99) is remembered by the calculator when the program branches using XEQ.

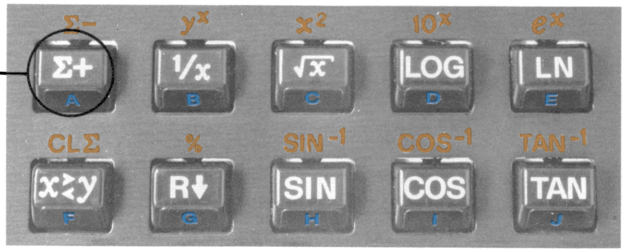
The HP-41C handles branches to ALPHA labels in a unique way. As soon as an ALPHA label is keyed into a program, the calculator records that label and its location in such a way that each ALPHA label knows where the next ALPHA label is located. A GTO or XEQ of an ALPHA label then causes the HP-41C to search from ALPHA label to ALPHA label for the ALPHA name. The HP-41C then branches to the corresponding location in program memory. The ALPHA label search is from the bottom-most program in program memory to the top-most program. The result is a search of the last programs first. This ALPHA label search scheme increases the speed of execution by decreasing search time.

Key Mapping

Another unique feature that you may have discovered is the correspondence between the top two rows of keys and the numbers 01 through 10. This feature lets you key in a two-digit label, address or function parameter using a single keystroke.

For example, when you press XEQ and the $\Sigma+$ key, the calculator interprets that as XEQ 01. The $\Sigma+$ key corresponds to the number 01.

01



01 through 05

06 through 10

So, when you execute a function that requires a two-digit address or parameter, you can simply press the key that corresponds to the desired number.

Here are some more examples:

GTO SIN	=	GTO 08
LBL LN	=	LBL 05
XEQ $x \gtrless y$	=	XEQ 06
STO 1/x	=	STO 02
RCL $\Sigma+$	=	RCL 01

Note that if you press one of the top two row keys to specify a number for a function requiring only a single digit input, only the right-most digit is used by the function. For example:

FIX TAN	=	FIX 0
ENG $\Sigma+$	=	ENG 1

The COPY Function

COPY is used to copy a program from an application module into program memory. With the application module in place and the desired program name in mind, execute COPY and spell out the program name. This will copy the specified program into program memory.

However, there are a few things that must be considered before you attempt a COPY . The application program on the application module must be able to fit into program memory. If it does not you will not be able to execute a successful COPY . Here is what happens when you execute COPY and specify a program name:

1. The calculator first searches for the specified name. If it is not found (it is misspelled, or the application module is not in place) the display will show ***NONEXISTENT***
2. The HP-41C then determines the length of the specified program.
3. The size of unused program memory is determined.
4. If the unused portion of program memory is large enough to accept the entire application program, the program is copied into program memory.
5. In the event that there is not enough room in program memory to hold the entire application program, the HP-41C will pack program memory (packing is explained in section 8). You will momentarily see ***PACKING*** in the display.
6. The calculator will then ask you to reenter the COPY function with the ***TRY AGAIN*** display.
7. If the unused portion of program memory is now large enough to hold the entire application program, the program will be copied into program memory. If the unused portion of program memory is still not large enough to hold the application program, the calculator will again pack (***PACKING***) and ask you to ***TRY AGAIN***
8. At this point, you should clear program instructions out of program memory to make room for the application program. If you continue to execute COPY when there is not enough room in program memory to hold the desired program, the HP-41C will continue to pack program memory while displaying ***PACKING***, and ask you to ***TRY AGAIN***

Attempting to COPY a program from program memory to another location in program memory will result in the ***RAM*** message (***RAM*** means Random Access Memory—these are the storage registers that you can store data and program instructions into). Attempting to DEL , CLP , ←, or insert into a program that is currently in an application module will result in the ***ROM*** display (***ROM*** means Read Only Memory—this is what application modules are stored in).

You may copy the application module program to which the calculator is presently positioned by not specifying a program name. For example, COPY ALPHA ALPHA copies the application module program that the calculator is currently positioned to into program memory.

Index

D

E

R

S

T

Function Index

All HP-41C functions can be recorded as instructions in program memory except those indicated. Functions with one name for keyboard execution and a second name for display execution are shown adjacent to each other in the columns below (e.g., [√x] on the keyboard and SQRT in the display). Unless otherwise noted, all of the following functions can be executed from the display and reassigned. Functions unique to the ALPHA mode keyboard are marked with *. Refer to the ALPHA mode keyboard on page 19 or on the back of the calculator. Note that only the functions on the normal or ALPHA mode keyboard are shown as key functions, even though all HP-41C functions may be assigned to the keyboard (except those indicated). To execute a function from the display, press [XEQ] [ALPHA], key in the alpha characters shown, then press [ALPHA].

Function: Display Execution	Function: Keyboard Execution	Description
	[■]	Shift key **(page 17, 18).** Not programmable or assignable.
+	[+]	Addition operator **(page 25).**
−	[−]	Subtraction operator **(page 25).**
×	[×]	Multiplication operator **(page 25).**
/	[÷]	Division operator **(page 25).**
1/X	[1/x]	Reciprocal **(page 80).**
10↑X	[10^x]	Common antilogarithm **(page 96).**
ABS		Absolute value **(page 78).**
ACOS	[COS^{-1}]	Arc cosine **(page 86).**
ADV		Advance paper if printer is in system **(page 105).**
AOFF	[ALPHA] mode key	ALPHA mode off **(page 154).** The ALPHA mode *key* **(page 18)** is not programmable or assignable.
AON	[ALPHA] mode key	ALPHA mode on **(page 154).** The ALPHA mode *key* **(page 18)** is not programmable or assignable.
	[APPEND] *	Append ALPHA display **(page 40, 151).** Not assignable, not executable.
ARCL	[ARCL] *	ALPHA recall. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 70).**
ASHF		ALPHA shift left **(page 154).**
ASIN	[SIN^{-1}]	Arc sine **(page 86).**

ASN	ASN	Assign. Requires function name and key location input **(page 61, 115).** Not programmable.
ASTO	ASTO *	ALPHA store. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 70).**
ATAN	TAN^{-1}	Arc tangent **(page 86).**
AVIEW	AVIEW *	ALPHA view **(page 72, 151).**
BEEP	BEEP	Beeper **(page 104).**
BST	BST	Back step **(page 132).** Not programmable.
CAT	CATALOG	Catalog list. Requires 1-number input. or indirect address **(page 60, 140).** Not programmable.
CF	CF	Clear program flag. Requires 2-digit, indirect 2-digit, or indirect stack address **(page 209).**
CHS	CHS	Change sign **(page 77).**
CLA	CLA *	Clear ALPHA register **(page 41).**
CLD		Clear display **(page 154).**
CLP		Clear program. Requires program name input **(page 119).** Not programmable.
CLRG		Clear all storage registers **(page 73).**
CLΣ	CLΣ	Clear statistics registers **(page 99).**
CLST		Clear automatic memory stack **(page 47).**
CLX	CL X/A	Clear X-register **(page 42).**
COPY		Copy (download or copy). Requires ALPHA program name input **(page 260).** Not programmable.
	← *	Correction key **(page 22, 42, 136).** Not assignable, not programmable.
COS	COS	Cosine **(page 86).**
D – R		Degrees to radians conversion **(page 87).**
DEC		Octal to decimal conversion **(page 105).**
DEG		Degrees mode **(page 85).**
DEL		Delete program memory lines. Requires 3-number input **(page 137).** Not programmable.
DSE		Decrement and skip if equal. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 164).**
	EEX	Enter exponent **(page 21, 77).**
END		End of program **(page 118, 178).**
ENG	ENG	Engineering notation display. Requires 1-digit, indirect 2-digit, or indirect stack address **(page 34).**
ENTER↑	ENTER↑	Enter number in X-register into Y-register **(page 46).**
E↑X	e^x	Natural antilogarithm **(page 96).**

E↑X–1		Natural antilogarithm for arguments close to zero **(page 96).**
FACT		Factorial **(page 81).**
FC?		"Flag clear" test. Requires 2-digit, indirect 2-digit, or indirect stack address **(page 209).**
FC?C		"Flag clear" test and clear. Requires 2-digit, indirect 2-digit, or indirect stack address **(page 209).**
FIX	[FIX]	Fixed point display. Requires 1-digit, indirect 2-digit, or indirect stack address **(page 32).**
FRC		Fractional portion of number **(page 79).**
FS?		"Flag set" test. Requires 2-digit, indirect 2-digit, or indirect stack address **(page 209).**
FS?C		"Flag set" test and clear. Requires 2-digit, indirect 2-digit, or indirect stack address **(page 209).**
GRAD		Grads mode **(page 85).**
GTO	[GTO]	Go to. Requires 2-digit label, ALPHA program name, indirect 2-digit, or indirect stack address **(page 159).**
GTO.	[GTO] [•]	Go to line number. Requires 3-number input or ALPHA label **(page 128, 134).** Not assignable, not programmable.
GTO..	[GTO] [•] [•]	Go to end of program memory and prepare calculator for new program **(page 112).** Not programmable, not assignable.
HMS		Decimal hours to hours, minutes, seconds conversion **(page 88).**
HMS+		Hours, minutes, seconds addition **(page 90).**
HMS–		Hours, minutes, seconds subtraction **(page 90).**
HR		Hours, minutes, seconds to decimal hours conversion **(page 88).**
INT		Integer portion of number **(page 79).**
ISG	[ISG]	Increment and skip if greater. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 164).**
LASTX	[LASTX]	Recalls LAST X register contents to X-register **(page 52).**
LBL	[LBL]	Label program. Requires 2-number label or ALPHA program name **(page 110).**
LN	[LN]	Natural logarithm **(page 96).**
LN1+X		Natural logarithm for arguments close to one **(page 96).**
LOG	[LOG]	Common logarithm **(page 96).**
MEAN		Mean **(page 101).**
MOD		Modulo (remainder) operator **(page 79).**
OCT		Decimal to octal conversion **(page 105).**

OFF		Power off **(page 106).**
	[ON]	Power on/off key **(page 15).** Not programmable or assignable. Refer to ON and OFF.
ON		Power on (continuous) function **(page 106).** Not programmable.
P - R	[P-R]	Polar to rectangular conversion **(page 94).**
PACK		Pack program memory **(page 141).** Not programmable.
%	[%]	Percent **(page 83).**
%CH		Percent of change **(page 84).**
PI	[π]	Pi **(page 82).**
	[PRGM] mode key	Program mode key **(page 106, 112).** Not programmable.
PROMPT		Prompt **(page 151).**
PSE		Pause **(page 147).**
R↑		Roll up **(page 44).**
R - D		Radians to degrees conversion **(page 87).**
R - P	[R-P]	Rectangular to polar conversion **(page 93).**
	[R/S]	Run/stop. Stops running program or starts a stopped program **(page 145, 147).**
RAD		Radians mode **(page 85).**
RCL	[RCL]	Recall. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 68).**
RDN	[R↓]	Roll down **(page 44).**
RND		Round **(page 78).**
RTN	[RTN]	Return **(page 177, 178).**
SCI	[SCI]	Scientific notation display. Requires 1-digit, indirect 2-digit, or indirect stack address **(page 33).**
SDEV		Standard deviation **(page 101).**
SF	[SF]	Set program flag. Requires 2-digit, indirect 2-digit, or indirect stack address **(page 209).**
Σ+	[Σ+]	Accumulations for statistics **(page 99).**
Σ−	[Σ−]	Accumulation correction **(page 103).**
Σ***REG***		Statistical register block specification. Requires 2-digit, indirect 2-digit, or indirect stack address **(page 99).**
SIGN		Sign, unary of *x* **(page 85).**
SIN	[SIN]	Sine **(page 86).**
SIZE		Size of register configuration (allocation). Requires 3-number input **(page 73, 117).** Not programmable.
SQRT	[$\sqrt{x}$]	Square root **(page 81).**

SST	SST	Single step **(page 129, 132).** Not programmable.
ST+	STO +	Storage register addition. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 74).**
ST−	STO −	Storage register division. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 74).**
ST*	STO ×	Storage register multiplication. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 74).**
ST/	STO ÷	Storage register division. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 74).**
STO	STO	Store. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 68).**
STOP	(R/S)	Stops program execution **(page 145).**
TAN	TAN	Tangent **(page 86).**
TONE		Tone of beeper. Requires 2-digit, indirect 2-digit, or indirect stack address **(page 104).**
	USER	USER mode key **(page 63).** Not programmable or assignable.
VIEW	VIEW	View register contents. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 72).**
X=0?	x=0?	X equal to 0 conditional test **(page 170).**
X≠0?		X not equal to 0 conditional test **(page 171).**
X<0?		X less than 0 conditional test **(page 171).**
X<=0?		X less than or equal to 0 conditional test **(page 171).**
X>0?		X greater than 0 conditional test **(page 171).**
X=Y?	x=y?	X equal to Y conditional test **(page 170).**
X≠Y?		X not equal to Y conditional test **(page 171).**
X<Y?		X less than Y conditional test **(page 171).**
X<=Y?	x≤y?	X less than or equal to Y conditional test **(page 171).**
X>Y?	x>y?	X greater than Y conditional test **(page 171).**
X<>		Exchange contents of X-register with any other register. Requires 2-digit, stack, indirect 2-digit, or indirect stack address **(page 105).**
X<>Y	x⇄y	Exchange X- and Y-registers **(page 44).**
XEQ	XEQ	Execute. Requires program or function name, label number or indirect address **(page 57, 114, 177).**
X↑2	x^2	Square **(page 82).**
Y↑X	y^x	Exponential **(page 97).**

HEWLETT PACKARD

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

00041-90001 Rev. B 8/79

Printed in U.S.A.