TEXAS INSTRUMENTS

Printed in Taiwan, R.O.C.

1060338-0103

TEXAS INSTRUMENTS

PROGRAMMABLE

TI-95 PROGRAMMING GUIDE

TEXAS INSTRUMENTS TI-95 PROGRAMMING GUIDE

Manual developed by:

The staff of Texas Instruments
Instructional Communications and
Design Communications
TI Corporate Design Center

Michael T. Keller
Chris M. Alley
David Thomas
De Warden
Joseph L. Willard

With contributions by:

Linda Ferrio
Art Hunter
Robert A. Pollan
Jacquelyn F. Quiram
Robert E. Whitsitt, II

Important

Function Reference

Use this list if you know the function you want to perform but you need a reminder of the key sequence. The page reference tells where to find additional information about the function. This list includes programming-related functions only; scientific-calculator functions are discussed in the *TI-95 Users Guide*.

General Functions	Key Sequence	Page
Learn Mode	LEARN	1-6, 1-18
Toggle Program Counter	2nd [PC]	2-6, 2-15
Clear Program	2nd [CP]	1-6, 1-18
Run Program	RUN	1-10, 1-20, 8-8, 8-33
Check Partitioning	INV 2nd [PART]	7-8, 7-14
Partition User Memory	2nd [PART]	7-9, 7-14, 7-15, 7-16
Halt	HALT	1-5, 1-20, 2-15
Break	BREAK	2-10, 2-15
Continue Execution	⟨GO⟩	2-10, 2-15
No Operation	2nd [NOP]	1-19
Pause	2nd [PAUSE]	2-13, 2-15
Restore Menu/Message	OLD	3-17, 4-21, 4-29
Assemble Program	2nd [ASM]	4-25, 4-29
Disassemble Program	INV 2nd [ASM]	4-25, 4-29
Indirect Addressing	2nd [IND]	6-3, 6-14

Program Editing Functions	Key Sequence	Page
Right	→	1-18
Left	←	1-19
Insert Instructions	2nd [INS]	1-14, 1-19
Delete Instruction	2nd [DEL]	1-15, 1-19

Alpha Functions	Key Sequence	Page
Delete Character	ALPHA ⟨DEL⟩	3-11, 3-18
Insert Characters	ALPHA ⟨INS⟩	3-11, 3-18
Right	→	3-10, 3-18
Left	←	3-10, 3-18
Set Cursor to Column	ALPHA ⟨COL⟩	3-10, 3-19
Merge Number	ALPHA ⟨MRG⟩	3-12, 3-19
Recall Alpha	ALPHA ⟨-->⟩ ⟨RCA⟩	3-9, 3-19, 3-20
Store Alpha	ALPHA ⟨-->⟩ ⟨STA⟩	3-9, 3-20
Character Code	ALPHA ⟨-->⟩ ⟨CHR⟩	3-7, 3-20
Toggle Lowercase Lock	ALPHA ⟨-->⟩ ⟨LC⟩	3-6, 3-20

(continued)

Listing Functions	Key Sequence	Page
List Program	LIST 〈PGM〉	1-12, 1-21
List Labels	LIST 〈LBL〉	4-24, 4-26
List Status	LIST 〈ST〉	2-9

Transfer Functions	Key Sequence	Page
Label Segment	2nd [LBL]	4-5, 4-26
Go To Label	2nd [GTL]	4-8, 4-26
Go To	INV 2nd [GTL]	4-10, 4-26
Subroutine Label	2nd [SBL]	4-13, 4-27
Subroutine	INV 2nd [SBL]	4-13, 4-27
Return	2nd [RTN]	4-11, 4-27
Define Function Key	2nd [DFN]	4-16, 4-28
Clear All Function Keys	2nd [DFN] CLEAR	4-23, 4-28
Clear One Function Key	2nd [DFN] *Fx* CLEAR	4-23, 4-29

Test Functions	Key Sequence	Page
If Greater Than	TESTS 〈IF>〉	5-4, 5-25
If Less Than or Equal to	TESTS INV 〈IF>〉	5-4, 5-25
If Less Than	TESTS 〈IF<〉	5-4, 5-25
If Greater Than or Equal to	TESTS INV 〈IF<〉	5-4, 5-25
If Equal to	TESTS 〈IF=〉	5-4, 5-25
If Not Equal to	TESTS INV 〈IF=〉	5-4, 5-25
Decrement and Skip on Zero	TESTS 〈DSZ〉	5-7, 5-26
Decrement and Execute on Zero	TESTS INV 〈DSZ〉	5-7, 5-26
Yes/No Response	TESTS 〈Y/N〉	5-11, 5-26

Flag Functions	Key Sequence	Page
Clear Flags	FLAGS 〈CLR〉	5-14, 5-27
Set Flag	FLAGS 〈SF〉	5-14, 5-27
Reset Flag	FLAGS 〈RF〉	5-14, 5-27
Test If Flag Set	FLAGS 〈TF〉	5-14, 5-27
Test If Flag Reset	FLAGS [INV] 〈TF〉	5-14, 5-27

File Functions	Key Sequence	Page
Load File	FILES 〈GET〉	8-10, 8-16, 8-27, 8-30
Save File	FILES 〈PUT〉	8-6, 8-12, 8-26, 8-29
Delete File	FILES 〈DF〉	8-20, 8-28, 8-32
Display Catalog	FILES 〈CAT〉	8-18, 8-28, 8-31
Clear Directory	FILES 〈-->〉 〈CD〉	8-21, 8-28, 8-32
Name Cartridge	FILES 〈-->〉 〈NAM〉	8-5, 8-27, 8-31

Input/Output Functions	Key Sequence	Page
Write Tape File	I/O 〈TAP〉 〈WRT〉	9-9, 9-22, 9-24
Read Tape File	I/O 〈TAP〉 〈RD〉	9-12, 9-22, 9-24
Verify Tape File	I/O 〈TAP〉 〈VFY〉	9-12, 9-23, 9-25
Call I/O	I/O 〈CIO〉	B-2, B-15
Key Wait	I/O 〈KW〉	B-22

System Functions	Key Sequence	Page
Store Byte	FUNC 〈SYS〉 〈STB〉	A-7
Recall Byte	FUNC 〈SYS〉 〈RCB〉	A-7
Assembly Language Subroutine	FUNC 〈SYS〉 〈SBA〉	A-15
Unformatted Mode	CONV 〈BAS〉 〈UNF〉	A-8

Table of Contents

This book describes the programming functions of the TI-95 calculator. Before using this book, you should already be familiar with the scientific-calculator functions, described in the *TI-95 User's Guide*.

Chapter 1: Working with Programs on the TI-95

Programming Keys Used in This Chapter	1-2
Introduction	1-3
Entering a Program	1-6
Running a Program	1-10
Listing a Program	1-12
Editing a Program	1-14
Reference Section	1-18

Chapter 2: Using Calculator Functions in a Program

Introduction	2-2
Using System-Menu Functions	2-6
Keyboard Commands versus Program Instructions	2-9
Making a Program Wait for Numeric Input	2-10
Pausing a Program to Display Results	2-13
Reference Section	2-15

Chapter 3: Displaying Messages

Introduction	3-2
The Alpha Menu	3-4
Creating an Alpha Message	3-5
Storing and Recalling Alpha Messages	3-8
Editing an Alpha Message	3-10
Merging Numbers with a Displayed Message	3-12
Using Alpha Messages in a Program	3-14
Reference Section	3-17

Chapter 4: Controlling the Sequence of Operations

Introduction	4-2
Before Proceeding with this Chapter	4-4
Using Program Labels	4-5
Using Go To Label	4-8
Using Go To	4-10
Using Subroutines in a Program	4-11
Programming the Function Keys	4-16
Creating a Function-Key Menu	4-18
Restoring a User-Defined Menu	4-21
Clearing Function-Key Definitions	4-22
Listing Program Labels	4-24
Speeding Up Program Execution	4-25
Reference Section	4-26

Chapter 5: Using Tests in a Program

Introduction	5-2
Using Comparison Tests	5-4
Using DSZ Tests	5-7
Using the YES/NO Test	5-11
Using Flags	5-13
Ways to Use Tests with Transfer Instructions	5-16
Using a Test to Exit a Loop	5-22
Reference Section	5-25

Chapter 6: Using Indirect Addressing

Introduction	6-2
Indirect Data Register Operations	6-4
Indirect Transfer of Program Control	6-8
Other Indirect Operations	6-13
Reference Section	6-14

Chapter 7: Partitioning Memory

Introduction	7-2
Why Change Memory Partitions?	7-4
Effects of Partitioning	7-6
Determining the Current Partitions	7-8
Partitioning from the Keyboard	7-9
Partitioning from Within a Program	7-12
Reference Section	7-14

(continued)

Introduction to TI-95 Programming

A keystroke programming language is built into the TI-95. By using the features and capabilities of this language, you can reduce the amount of manual work required for repetitive problem-solving tasks.

What Is Keystroke Programming?

Keystroke programming, in its simplest form, consists of storing a sequence of keystrokes in the calculator's memory. The key sequence that is stored is called a **program** and the process of storing the keystrokes is called **programming**. After storing a program in memory, you can perform the key sequence by **running** the program.

You can store virtually all the functions of the TI-95 in a program.

Besides the functions used for calculations, you can include special programming functions that enable your program to perform operations such as displaying messages, repeating key sequences, and making decisions.

Why Write Your Own Programs?

The main advantages of writing a program are to save time, improve productivity, and avoid errors. Writing a program can be especially beneficial when:

- You need to solve a problem that requires an iterative process to arrive at a solution.
- You need to perform a lengthy calculation repetitively, using different data for each repetition.
- You need to change the sequence of steps in a calculation based on a condition or on the value of an intermediate result.
- You need to retain a keystroke solution for future use.

By writing a program for such tasks, you can reduce your workload to entering the numbers needed by the program. You do not need to reenter the keystrokes that are stored in the program. Your program can prompt you for the required information, control the sequence of steps, perform the calculations, and display and label the results.

Programming Features

The TI-95 has features that make it easy to write programs. Some of these features include:

- 7200 bytes of user memory, partitioned as data registers, program steps, and file space. You can change the partitioning of this memory to suit your programming needs.
- An alphabetic display that shows program instructions in a readable, mnemonic form, rather than as a series of numeric codes that you must memorize or look up to interpret.
- An alpha mode that lets you display descriptive messages from within a program. You might use such messages to label the result of a calculation, request information, or indicate errors.
- Five user-definable function keys that can be labeled in the display. You can design your programs to define the function of each key.
- Storage and retrieval functions that let you save programs and data as files for later use. You can save files in the calculator's file space, in an optional Constant Memory™ cartridge, or on cassette tape using the optional CI-7 Cassette Interface Cable. Program files that are saved in the calculator's file space or in a Constant Memory cartridge can be executed directly from the file space or cartridge.

(continued)

Using This Book

This book describes the functions used to program your calculator. The preliminary chapters cover functions basic to all programming tasks, such as entering and executing a program. Later chapters introduce more complex functions, with the final chapters discussing the most sophisticated functions of the calculator.

A reference section is included at the end of each chapter. This section can be beneficial in two ways. First, you can use it to review the information discussed in the chapter. Second, you can use it as a reference source when you want to look up details about the functions. Some of the detailed information provided in the reference sections is not repeated elsewhere. If you are already familiar with a keystroke programming language, you can use these sections to discover differences between the TI-95 and the calculator that you know.

Because each chapter builds upon the material discussed in earlier chapters, it is recommended that you read the chapters in the order presented.

The appendices provide information on manipulating the system memory of the calculator and using the input/output functions, as well as tables listing character codes, key codes, and instruction mnemonics.

Notational Conventions

Many functions are incomplete without additional identifying data following the function. This data is referred to as a **field**. For example, to store a number in memory, you must follow the STO key with the address of the register in which you want to store the number. The register address is the field.

Notational conventions allow you to see at a glance whether a field is required by a function. The following notational conventions are used to represent fields in this book.

- An *n* character represents a digit (0–9) in a field. The number of *n* characters in the notation indicates the number of digits in the field. For example, INV 2nd [GTL] *nnnn* indicates a four-digit field.
- An *X* character represents a letter in a field. Any letter from A through Z is valid for this field. For example, STO *nnn* or *X* indicates that STO can have either a three-digit numeric field or a letter field. In fields of this type, a lowercase letter is converted automatically to an uppercase letter.
- An *a* character represents an ASCII character (digit, letter, or punctuation sign) in a field. This type of field is referred to as an alphanumeric field. The number of *a* characters in the notation indicates the number of alphanumeric characters in the field. For example, 2nd [LBL] *aa* indicates a two-character alphanumeric field. Alphanumeric fields can contain both uppercase and lowercase letters.
- An *h* character represents a hexadecimal digit in a field. The hexadecimal digits are 0 through 9 and A, B, C, D, E, and F (A through F are entered with 2nd [A_H] through 2nd [F_H]). The number of *h* characters in the notation indicates the number of hexadecimal digits in the field. For example, ⟨SBA⟩ *hhh* indicates a three-digit hexadecimal field.

Chapter 1: Working with Programs on the TI-95

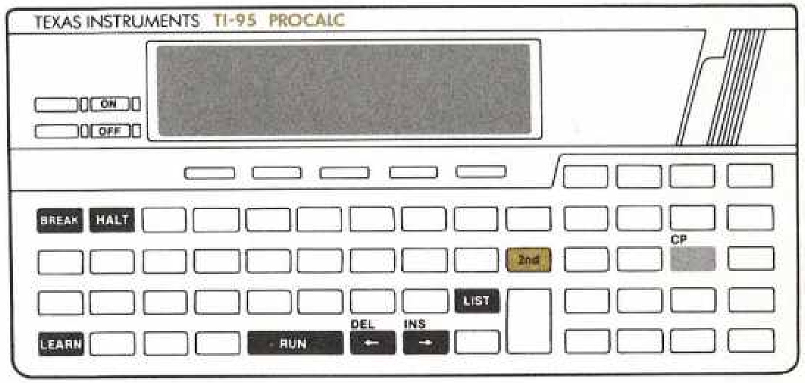
This chapter shows you how to enter, run, list, and edit a program.

Table of Contents

Programming Keys Used in This Chapter 1-2
Introduction 1-3
Entering a Program 1-6
Running a Program 1-10
Listing a Program 1-12
Editing a Program 1-14
Reference Section 1-18

Programming Keys Used in This Chapter

The keys used to enter, run, list, and edit a program are shown in the figure below. Familiarize yourself with these keys and their locations on the keyboard.



Introduction

As you enter a program, the calculator does not execute your keystrokes. Instead, it stores the keystrokes in a series of numbered memory locations called *program steps*. The number of program steps is controlled by the memory partitioning of the calculator.

How Are Programs Stored?

Each program step is identified by a unique four-digit address. The first step is program address 0000. As a general rule, each keystroke is stored in a separate step. However, there are several exceptions.

- Second functions, such as 2nd [x!], are combined into a single step. Only the actual operation, x! in this example, is stored.
- Menu functions, such as CONV ⟨MET⟩ ⟨F-C⟩, are stored as a single step. Only the actual function, ⟨F-C⟩ in this example, is stored.
- Functions that require a field, such as STO 007, are stored in two or more steps—one for the instruction and one or more for the field.
- There are two inverse function key sequences in which INV combines with the function key sequence it precedes. These two functions, SBR and GTO, are discussed in Chapter 4.

Arrangement of Program Steps

The following illustration can help you visualize the arrangement of program steps in memory. The "Address" column gives the number assigned to each program step.

Address	Program Step
0000	INV
0001	LOG
0002	+
0003	5
0004	x!
0005	=
0006	STO
0007	A
0008	HLT
0009	NOP

Any unused steps automatically contain a NOP (no operation) instruction.

What Is an Instruction?

Once a function is stored in program memory, it is referred to as a program instruction. A program instruction is a key or key sequence that forms a complete function. For example:

- [y^x] is an instruction.
- [2nd] [x!] is an instruction.
- [RCL] A is an instruction.
- [STO] [+] 007 is an instruction.
- [2nd] [FIX] 2 is an instruction.

What Is a Mnemonic?

The symbol displayed by the calculator when you store a function in program memory is called a **mnemonic**. The mnemonic displayed for primary and second functions generally corresponds to the symbol shown on the keyboard. For some functions, the mnemonic is an abbreviated form of the key symbol.

Some examples of the mnemonics displayed for calculator functions are listed below.

Key Sequence	Mnemonic
[CLEAR]	CLR
[×]	*
[÷]	/
[+/−]	+/−
[x~t]	x~t
[2nd] [nPr]	nPr
[INCR] A	INC A
[STO] [+] 007	ST+ 007

A complete list of the mnemonics of the calculator is given in Appendix C.

How Are Programs Executed?

When you run a program, the calculator sets an internal pointer called a **program counter** to the first program step. It then executes the instruction stored at that location. After the proper action is accomplished, the program counter advances to the next instruction.

The calculator continues to advance the program counter and execute instructions until one of the following events occurs.

- You stop the program by manually pressing the [BREAK] or [HALT] key.
- The program is stopped by an instruction stored in the program, such as [BREAK] or [HALT].
- The program counter reaches the end of program memory, which causes an error condition that stops execution. (Certain other errors also cause program execution to stop. See Chapter 5 for details.)

Entering a Program

The learn mode of the calculator enables you to enter a sequence of keystrokes into program memory. Once entered, a program remains in memory until you clear it or replace it with another program.

Activating the Learn Mode

Before you can enter a program, the calculator must be in the learn mode. To activate the learn mode:

1. Press LEARN.

   The following screen is displayed.

```
SHOW LOCATION:
1st  PC   END        ESC
```

⟨1st⟩	Positions the cursor to the first step in program memory.
⟨PC⟩	Positions the cursor to the step specified by the current setting of the program counter.
⟨END⟩	Positions the cursor to the last instruction stored in program memory.
⟨ESC⟩	Clears the learn mode menu.

2. Select the option for the program location you want to display. The calculator displays the program counter (PC) on the second line of the display. The PC value corresponds to the address of the step marked by the cursor.

Clearing the Program Memory

The key sequence 2nd [CP] clears all keystrokes from program memory. When you clear program memory, the calculator fills all program steps with NOP (no operation) instructions. To prevent the accidental clearing of a program, the clear program function works only in the learn mode.

Entering the Program

After selecting the learn mode, enter your program by pressing the desired sequence of keys. When a key (or key sequence) is pressed, the calculator enters that function into program memory and displays the mnemonic that represents the function.

As you enter your program, consider the following points.

- Enter the instructions in the order that you want them executed. (It is possible to change the order in which instructions execute by using transfer instructions. Transfer instructions are discussed in Chapter 4.)
- Include every instruction that you want executed. If you omit an instruction, your program will probably not run as intended.
- End your program with a HALT instruction. The HALT function instructs the calculator to stop program execution and return to keyboard operation.
- If you make a mistake in entering a program, you can correct it by using the editing keys discussed later in this chapter.

Exiting the Learn Mode

After entering the keystrokes that make up your program, you must exit the learn mode before you can run the program.

To exit the learn mode, press LEARN again.

When the calculator exits the learn mode, it displays the contents of the numeric display register. Any alpha message that was in the display when the learn mode was entered initially is not redisplayed. (Alpha messages that you have created can be recalled, as explained in Chapter 3.)

(continued)

Exiting the Learn Mode (Continued)

Exiting the learn mode does not affect the contents of program memory or change the location of the program counter. The program counter remains where it was last positioned, until you perform an operation that changes it.

Example

To illustrate the process of programming the calculator, write a program to calculate the cube of a number. To make this calculation manually, you would:

- Enter the number to be cubed.
- Press [yˣ] to specify the power function.
- Press 3 to specify the power.
- Press [=] to complete the calculation and display the result.

The last three actions in the list are the actions that you want to put into the program. Do not put the number to be cubed into the program, or you will have to change the program every time you want to cube a different number.

You write this program by entering the learn mode and duplicating the keystrokes in the list. The program design assumes that the number to be cubed is already in the display when the program is started.

Example (Continued)

To write the program, duplicate the keystrokes shown below.

Procedure	Press	Display
Activate learn mode	[LEARN]	SHOW LOCATION:
Display first step	⟨1st⟩	
Clear program memory	[2nd] [CP]	
Enter function	[yˣ]	y^x
Specify power	3	y^x 3
Calculate result	[=]	y^x 3=
Stop execution	[HALT]	y^x 3= HLT
Exit learn mode *	[LEARN]	0.

*The value displayed is the value that was in the numeric display register before you activated the learn mode.

The program is stored in the calculator. The procedure for running the program is described on the following pages.

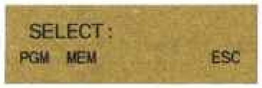
Running a Program

The advantage of entering a program into memory is that you can perform the same key sequence as many times as needed, without having to re-enter the program keystrokes.

Procedure

To run the program currently stored in program memory:

1. Be sure the calculator is not in the learn mode.
2. Press [RUN] to display the following menu.


SELECT:
PGM MEM ESC

⟨PGM⟩	Runs the program in program memory, starting at program step 0000.
⟨MEM⟩	Enables you to run a program you have saved in the file space. (File space is discussed in chapter 8, "File Operations.")

Note: If an optional software cartridge or 8K Constant Memory ™ cartridge is installed, the name of the cartridge is displayed in the menu. See the "File Operations" chapter for details on running a program in a Constant Memory cartridge.

3. If your program requires you to enter a number into the display before starting the program, enter the number.
4. Select the menu option for the program you want to run.

   The calculator runs the program. The **RUN** indicator is displayed while the program is running.

Example

Using the program written on page 1–9, calculate the cubes of 5 and 3.

Procedure	Press	Display
Display the menu	[RUN]	SELECT:
Calculate 5^3	5 ⟨PGM⟩	125.
Calculate 3^3	3 ⟨PGM⟩	27.

You only need to press [RUN] when ⟨PGM⟩ is not showing above the [F1] key. As long as ⟨PGM⟩ is visible, the program can be executed by pressing [F1].

Listing a Program

You can list the program currently in program memory by using the [LIST] ‹PGM› function. If you have a printer connected to the calculator, [LIST] ‹PGM› also prints the listing. The calculator displays the listing in groups. The address of the first instruction in the group is shown in the left side of the display.

Listing the Program

To list the program currently in memory:

1. Be sure the calculator is not in the learn mode. Then press [LIST] to display the following menu.

```
LIST:
REG  PGM  LBL  ST
```

2. Press ‹PGM› to display the following menu.

```
START LISTING AT
1st  PC
```

‹1st›	Begins at the first step in program memory.
‹PC›	Begins at the address specified by the current setting of the program counter.

3. Select the point at which you want to begin the listing. Unless you pause or stop the listing as explained in the following sections, the calculator lists through the last step in the program.

Controlling the Speed of the Listing

If you do not have a printer connected, the calculator pauses for one second before displaying the next group of program steps.

However, you can use the [→] key to control the speed of the listing.

- ► To pause the listing indefinitely, hold down the [→] key.
- ► To advance through the listing without the one-second pause, repeatedly press and release the [→] key.

Stopping the Listing

To stop a program listing before it has finished, press and hold the [BREAK] or [HALT] key until the display returns to the list menu.

Example

List the program that you entered on page 1-9. Before starting the listing, be sure the calculator is not in the learn mode.

Procedure	Press	Display
Select program listing	[LIST] ‹PGM›	START LISTING AT
Begin at first step	‹1st›	0000 y^x 3= HLT
Listing finishes		LIST:

Editing a Program

You can edit a program by inserting, deleting, or replacing instructions. The calculator must be in the learn mode before you can edit the program. All changes occur at the location of the cursor. To move toward the beginning or end of a program, press [←] or [→], respectively. Both keys repeat when held down for approximately one second.

Inserting an Instruction

Pressing [2nd] [INS] displays the **INS** indicator and places the calculator in the insert mode. In the insert mode, the instructions you enter are inserted into the program at the position of the cursor. The calculator remains in the insert condition until you cancel it by pressing [←], [→], [2nd] [DEL], or [LEARN]. When you insert an instruction, the instruction at the current position of the cursor and all instructions that follow are shifted toward the end of program memory.

Example

The following key sequence changes the program entered on page 1–9 so that it divides the cubed value by 2. The current program is y^x 3= HLT. Insert "/2" before the "=" instruction.

Procedure	Press	Display
Activate learn mode	[LEARN] ⟨1st⟩	y^x
Position cursor on =	[→] [→]	y^x 3=
Prepare for insert	[2nd] [INS]	y^x 3=
Insert instructions	[÷] 2	y^x 3/2=
Exit learn mode	[LEARN]	0.

Running the Example

Test your editing changes by running the new version of the program.

Procedure	Press	Display
Display **RUN** menu	[RUN]	SELECT:
Enter the number	5	5
Display result	⟨PGM⟩	62.5

Deleting an Instruction

To delete the instruction at the current position of the cursor, press [2nd] [DEL]. If the instruction has a field, the field is also deleted. All instructions following the deleted instruction are moved toward the beginning of program memory.

Example

Delete the "/2" instructions from the sample program. This restores the program to its original form, which calculates the cube of a number.

Procedure	Press	Display
Activate learn mode	[LEARN] ⟨1st⟩	y^x
Move cursor to /	[→] [→]	y^x 3/
Delete /	[2nd] [DEL]	y^x 32
Delete 2	[2nd] [DEL]	y^x 3=
Exit learn mode	[LEARN]	0.

Running the Example

Test the program.

Procedure	Press	Display
Display **RUN** menu	[RUN]	SELECT:
Enter the number	5	5
Display result	⟨PGM⟩	125.

(continued)

Replacing an Instruction

To replace an instruction, just "write over" the old instruction. For example, to replace a / with a *, position the cursor on the / symbol and press [×]. Because / and * each occupy one program step, no further change is necessary.

Watch that you do not leave an unwanted [INV] preceding the new instruction. Also consider whether the new instruction requires more steps than the instruction it replaces. If more steps are required, use the insert function to enter the extra keystrokes.

To replace the field of an instruction, you must reenter the entire instruction. For example, to replace [STO] A with [STO] B, place the cursor on **STO** and press [STO] B. The calculator does not permit you to position the cursor over a field, a feature that prevents accidental changes to the field.

If you replace an instruction that has a field with an instruction that occupies fewer program steps, the calculator stores NOP instructions in the extra steps. For example, if you replace [STO] A (which occupies two steps) with [=], the calculator will place a **NOP** in the second step.

When you enter the first character or digit of a field, the calculator reserves enough program steps for the entire field. If you know that there is not enough space to enter the entire field without writing over existing instructions that you want to save, press [2nd] [INS] before you begin entering the field.

Example

Change the sample program so that it raises the entered number to the fifth power instead of cubing it. The current program is **y^x 3= HLT**. Replace the "3" with a "5."

Procedure	Press	Display
Activate learn mode	[LEARN] ⟨1st⟩	y^x
Position cursor on 3	[→]	y^x 3
Replace 3 with 5	5	y^x 5=
Exit learn mode	[LEARN]	0.

Running the Example

Now test the program.

Procedure	Press	Display
Display **RUN** menu	[RUN]	SELECT:
Enter the number	5	5
Display result	⟨PGM⟩	3125.

Reference Section

Use this section as a source of reference information on entering, running, listing, and editing a program.

Learn Mode

LEARN—Displays the learn mode menu or exits the learn mode, depending upon the calculator's state. When the learn mode is not active, pressing LEARN displays the learn mode menu. When the learn mode is active, pressing LEARN leaves the learn mode. Entering and exiting the learn mode clears the definitions (if any) of the function keys but does not clear the numeric display register.

LEARN ⟨1st⟩—Sets the program counter to step 0000 and enters the learn mode at the beginning of program memory.

LEARN ⟨PC⟩—Enters the learn mode at the current address of the program counter.

LEARN ⟨END⟩—Sets the program counter to the last instruction in the program and enters the learn mode at that location.

Clear Program

2nd [CP]—Clears a program from memory. Clear program only operates when the calculator is in learn mode.

→

→—In the learn mode, the → key increments the program counter to the beginning of the next instruction, which is displayed in full, including any field. The key repeats if held down longer than one second.

Outside of the learn mode, the → key controls the speed of listing functions. Holding down the → key pauses the listing; repeatedly pressing it accelerates the listing. For details on the use of → in the alpha mode, refer to Chapter 3 of this guide.

←

←—In the learn mode, the ← key decrements the program counter to the beginning of the previous instruction. The key repeats if held down longer than one second.

Outside of the learn mode, the ← key removes the last digit of a numeric entry. For details on the use of ← in the alpha mode, refer to Chapter 3 of this guide.

Insert

2nd [INS]—Places the calculator in insert mode, permitting you to insert program instructions at the current location of the program counter (used only in the learn mode). Pressing 2nd [DEL], ←, →, LEARN, or OFF cancels the insert mode.

The instruction (if any) in the highest-numbered program step in program memory is lost each time you insert an instruction.

Delete

2nd [DEL]—Deletes the instruction, including any field, stored at the current location of the program counter (used only in the learn mode).

No Operation

2nd [NOP]—Enters a no operation instruction into program memory. This function is valuable when you want to eliminate an instruction from a program without altering the addresses of any instructions in the program. No operation instructions are ignored during program execution.

A NOP instruction appears as a space character when the cursor is positioned over it in the learn mode.

(continued)

Halt

[HALT]—Stops program execution without affecting the definitions (if any) of the function keys.

You can use [HALT] from the keyboard to stop a running program or a listing function. You can use a [HALT] instruction in a program to stop execution at the point in the program where the halt instruction is placed.

The instruction mnemonic for [HALT] is HLT.

Run

[RUN]—As a keyboard command, [RUN] displays the program execution menu. As a program instruction, [RUN] enables an executing program to identify and execute another program. For details on the use of [RUN] as a program instruction, refer to page 8-34 of this guide.

[RUN] ‹PGM›—Runs the program currently stored in program memory, beginning with the first program instruction.

[RUN] ‹MEM›—Runs a program stored in the calculator's file space. For details on the use of this function, refer to page 8-8 of this guide.

[RUN] ‹Directory›—Runs a program stored in a software cartridge or Constant Memory cartridge. (A directory name appears in the menu only when a cartridge is installed in the calculator.) For details on the use of this function, refer to page 8-8 of this guide.

Note: Running a program does not clear calculations in progress. Therefore, if you decide to run a program before you complete a calculation, press [CLEAR] first to clear any pending numeric operations. You should also press [CLEAR] if you decide to rerun a program that was stopped in the middle of a calculation. To be certain that no pending numeric operations will affect your program's calculations, you can include a [CLEAR] instruction at the beginning of the program.

List

[LIST]—Displays the list menu. Although all listing functions can be entered as program instructions, they are used primarily as keyboard commands. (Pressing [LIST] clears the numeric display register.)

Listings are normally displayed at a one-second rate, but they can be paused or accelerated using the [→] key. Listings stop automatically when the last item has been listed but can be cancelled manually by holding down the [BREAK] or [HALT] key.

If a printer is connected to the calculator, all listings are printed and displayed. The speed of the listing is determined by the printing rate of the printer.

[LIST] ‹REG›—Lists the contents of the data registers, starting with the register address in the numeric display register.

The list registers function is represented by the mnemonic LR in program memory.

[LIST] ‹PGM›—Lists the contents of program memory. When this key sequence is executed as a keyboard command, you are prompted to specify whether you want the listing to start at the beginning of program memory or at the current location of the program counter. When this function is executed in a program, the listing always starts at program address 0000.

The list program function is represented by the mnemonic LP in program memory.

[LIST] ‹LBL›—Lists labels used in program memory. For details on the operation of this function, refer to page 4-24 of this guide.

[LIST] ‹ST›—Lists status information about the calculator. For details on the operation of this function, refer to page 2-9 of this guide.