TEXAS INSTRUMENTS
TI-57II

OWNER'S MANUAL

English

KEY INDEX

This indexed keyboard provides a quick page reference to the description of each key.

2nd	8	INV	8	R/S	28	OFF	9	ON/C	11
x=t	38	x≥t	38	SBR	41	Dsz	44	Del	33
RST	28	GTO	37	LBL	36	BST	31	SST	31
log	18	lnx	18	1/x	16	x^2	16	$\sqrt{x}$	16
DRG▸	19	P⇌R	21	DMS-DD	20	π	10	x!	18
DRG	19	sin	19	cos	19	tan	19	y^x	18
C.t	25	Fix	16	Intg	18	Frac	18	|x|	18
x⇌t	25	EE	14	(	12	)	12	÷	11
Part	23								
STO	23	7	10	8	10	9	10	×	11
CM	23								
RCL	23	4	10	5	10	6	10	−	11
CP	29								
EXC	23	1	10	2	10	3	10	+	11
Pause	29								
LRN	27	0	10	.	10	+/−	10	=	11

CAUTION
Your TI 57-II has constant memory even when the calculator is turned off so that it remembers:

- The memory partition of the calculator
- Programs stored in memory
- Numbers stored in data memories and in the t-register

If you experience any problem, the following sequence will re-initialize the calculator for you.

ON/C ON/C 2nd Part 7 2nd CM 2nd C.t 2nd Part 1 ON/C

If you cannot get into the "LEARN" mode, change the memory partition with the following key sequence:

2nd Part m, m being the number of data memories you need (m ≤ 6)

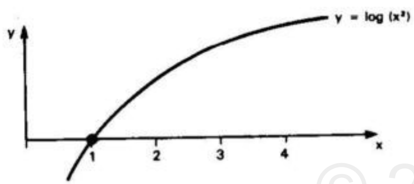
Texas Instruments reserves the right to make changes in materials and specifications without notice.

EXAMPLE

Writing a program to evaluate a function.

PROBLEM:

Plot the curve $y = \log(x^2)$ for integer values of x, starting at x = 1.



PRESS	DISPLAY		COMMENTS
ON/C ON/C	0		
2nd Part 1	48,1		Initialize the calculator
LRN 2nd CP	— —	St	Introduce the program
STO 0 x² log	31	02	inside the calculator
2nd Pause 1 +	85	05	memory
RCL 0 = RST	21	08	
LRN ON/C	0		
RST 1 R/S	0		y (0)
	0.60206		y (1)
	0.9542425		y (2)

Hold R/S down to stop the execution.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

Why a programmable calculator?

What does programming a computer mean?

Quite simply it's the action carried out in giving it a task to perform for us.

What knowledge is required in order to program?

First, we have to know how to complete the task ourselves, then understand the structure of the machine and finally write the task in the language of the machine in order that it can successfully understand and execute the task. A calculator, like a computer, is "fast" but "stupid". Everything has to be explained to it, even stages that appear evident to us, because the calculator works in a step by step logical manner.

But what is the difference between a computer and a programmable calculator?

Both have the same structure and follow the same rules of use. The main difference is essentially the size of the internal memory and the number of instructions that can be used.
Your TI 57-II offers all the necessary instructions to allow you to perform different types of programs and variations, including straight line programming, tests, loops, branching and subroutines. Therefore, it forms a good support for an introduction to programming, the limit in its memory size and in the number of instructions is in order to allow focusing on concepts and organisation. The TI 57-II is therefore a pedagogic tool.

What is a pedagogic tool?

In learning how to program we first must get acquainted with "tomorrow disciplines", but also we need to have the capacity to build "mathematical entities", then to assemble and see how they act.
Mathematics often appears to be a restricted field, dedicated to selected people, mainly because it is introduced on a too logical and formal way, neglecting other qualities such as intuition, creativity, esthetic sense...
Solving a problem requires above all a good analysis, but if the analysis does not lead to the solution, we will have to create elementary "mathematical models", that are well known, the study of their behaviour will very often dictate the way to link them to reach the correct solution.
Using a simple program will occasionally lead to an extension of the problem, opening new unexperimented fields, towards a certain "perfectionism" regarding the machine and ourselves.
The logic aspect is not altered because the calculator is managed with a strict mathematical logic. Any error is pointed out without mercy by the machine. However, as it is a machine it will detect the error but will in no

way make any comment. And that is where pupils have to concentrate to solve the problem in their program, before a solution can be achieved when previously in the written solution the error would have been discovered by the teacher.

Programming a calculator is not only a new technique, it is also "performing mathematics in a very live and exciting way". Perhaps this is why many young people (and older people too!) have found writing programs fun, and perhaps also because they secretly enjoy a sense of revenge upon the mathematical problems posed, by not only deriving the solutions but also by understanding the concepts involved in achieving the solution.

Chapter I — Using your TI 57-II as a scientific calculator.

Introduction

Your TI 57-II is more than just a programmable calculator. You can operate it as a powerful scientific calculator, always ready to perform calculations right from the keyboard.

The TI 57-II slide-rule features include:

- **AOS™ Algebraic Operating System.**
  Comprehensive data entry with the number and decimal keys, a π Key and parentheses. Multiplication, division, addition and substraction may be used with the Algebraic Operating System (which allows the entry of most problems as they are written) with up to four operations and fifteen parentheses pending. Results may be stored in up to seven user data memories. Data may be entered and displayed in standard format (with the necessary number of decimal places) or in scientific format.

- **Mathematical and Scientific Functions.**
  Mathematical and scientific keys including reciprocal, square, square root, universal power and roots, factorial, absolute value, fractional and integer parts, logarithm in both common and natural form, and all common trigonometric functions and their inverses for angles measured in degrees, radians or grads.

- **Built-in Conversions:**
  Conversion keys for polar to rectangular coordinates, degrees/minutes/seconds to decimal degrees, angular measures units, and the reverse of each of these.

This chapter explains the features and keys listed above, including information on why each key is important as well as how it is used.

The sections of the chapter are listed below. If you are familiar with those basic functions, you may go directly to chapter 2 and get right into programming.

Section 1 — Keyboard and display basic.
Section 2 — Data entry keys.
Section 3 — Algebraic keys.
Section 4 — Memory operations.

Section 1 — Keyboard and display basics.

This section is a quick explanation of the Basics. Please keep the calculator with you in order to try each feature as it is presented.

Turn you TI 57-II on with the [ON/C] key (at the top right of the keyboard). A zero appears in the display.
If the batteries were momentarily removed or replaced, reset the calculator by pressing [ON/C], [ON/C], [2nd] [Part] [7], [2nd] [CM], [2nd] [C.t], [RST] and [ON/C]

THE KEYBOARD.
If you look at the keyboard of the calculator, you will notice that the functions are grouped by subjects (data entry, memory operations, trigonometric functions, mathematical functions, programming), in order to facilitate locating a function on the keyboard.

As the calculator has many features, some of its keys have more than one function. The symbols printed above some of the keys are second functions. To perform one of these functions, you need to press the [2nd] key and then press the key for the function you wish to execute. For example, to calculate the square root of 4.5, enter 4.5 and press [√x]. To find the factorial of 5, enter 5 and press [2nd] [x!]. Notice that pressing [2nd] once turns on a 2nd indicator in the display, indicating that the next key will operate as a second function. Pressing the [2nd] key again will eliminate the second function operation and reset the 2nd indicator.

In this book, keys for first functions are shown with black print on a white background. Keys with a black background are used to indicate second functions.

The inverse key — [INV] — also provides additional calculator functions. When [INV] is pressed, the calculator executes the inverse of the function selected next.

NOTE: in cases where you need to use both the [2nd] and [INV] keys, they may be pressed in any order.

THE DISPLAY.

The display shows up to 8 digits plus eventually two digits as an exponent. Up to 8 significant digits can be entered into the calculator at any one time (entries after the 8th digit are ignored). However the display register stores eleven digits internally for use in calculations, rounded off to 8 digits in the display.
Negative numbers are shown with a negative sign immediatly to the left of the number.
Turning the calculator off (with the [OFF] key) and back on (with the [ON/C] key) removes the number in the display and any pending calculations. The contents of the user data memories are not affected.

DISPLAY INDICATORS

Five displays indicators give you some extra information on the status of your calculator.

Special functions indicators:
2ND and INV indicators will appear whenever the [2nd] or [INV] key are pressed, to remind you that you will select an inverse or second function. To cancel the second or inverse function selection, press [2nd] or [INV] a second time.

Angular mode indicators:
Whenever the calculator is turned on, it is always in degree mode. Other angular modes are indicated by a display indicator: RAD for radian mode and GRAD for grad mode.
The angular units may be changed with the [DRG], [INV] [DRG], [2nd] [DRG▸] and [INV] [2nd] [DRG▸] keys.

Run mode indicator.
The RUN indicator indicates that a program is running. To stop the execution and reset the indicator, hold down the [R/S] Key.

DISPLAY MESSAGES

- "CALC" appears any time your calculator is executing a function. During calculations, the indicators are displayed and the digits disappear, being replaced by the "CALC" message.

- "ERROR" appears any time that an error occurs (for example: division by zero). The list of the error conditions is given in the APPENDIX A. Whenever an error occured, press the [ON/C] key to clear the error message.

APD™ AUTOMATIC POWER DOWN

To conserve power, after 15 to 35 minutes of nonuse the calculator is automatically powered down through the APD™ feature. However, just turning it back on allows you to continue in the state the calculator was in, use the values in the user data memories, and any stored program. Any pending operations and immediate values are lost. The effect is the same as if you had pressed the [OFF] key.

NOTE: When the TI 57-II is running a user program, the Automatic Power Down is disabled in order to allow the execution of very long programs.

Section 2 – Data entry keys.

The following keys are used in entering, removing, and manipulating data to be used in subsequent calculations.

[0] - [9] – **Digit keys**

The digit keys allow any number to be entered into the display in a logical left-to-right order.

[•] – **Decimal point key**

The calculator operates with a floating decimal point which can be placed wherever needed. The decimal point is not displayed for integer numbers. A zero precedes the decimal point for numbers less than one. Zeros trailing the last significant digit on the right of a decimal point are not displayed unless the [2nd] [Fix] key has been used to fix the number of decimal places displayed.

[+/–] – **Change sign key.**

Pressing the change sign key instructs the calculator to change the sign of the displayed value. This allows the use of negative numbers in calculations.

[2nd] [π] – **PI key.**

The [2nd] [π] keys enters the value of pi to 11 significant digits, with a value of 3.1415926536, the display shows the value of pi rounded to eight digits, or 3.1415927.

[ON/C] – **Clear key.**

The [ON/C] key is also used to clear entries and operations. If an incorrect number is entered, press the [ON/C] key and re-enter the number. If an operation has just been pressed, pressing the [ON/C] key clears all pending operations and operands. Pressing the [ON/C] key twice always clears the display and all pending operations and operands. The user data memories and program registers are not affected.

[+], [–], [X], [÷], [=] – **Arithmetic keys.**

The basic arithmetic operations of addition, subtraction, multiplication and division are performed with these five keys. The equal key completes all pending operations and prepares the calculator for new calculations.
Several operations can be combined in one expression and entered into the calculator as written from left to right. The calculator has a special feature called the Algebraic Operating System to sort the operations and perform them in the correct order.

AOS™ ALGEBRAIC OPERATING SYSTEM.

The AOS™ Algebraic Operating System allows entering numbers and combined operations into the calculator in the same order in which they are written mathematically. Combined operations are performed following the universally accepted rules of the algebraic hierarchy which assigns priorities to the various mathematical operations. Without such a fixed set of rules, expressions with several operations could have more than one correct interpretation.

For example, the expression

$5 + 4 \times 3 - 2$

could have several different results. However, the rules of the algebraic hierarchy state that multiplications and divisions should be performed before additions and subtractions. Using these priorities, the calculator finds the correct solution is 15.

The above equation has to be evaluated as: $5 + (4 \times 3) - 2$.

The complete algebraic hierarchy, in descending order of priority, is:

1. Single variable function keys act immediately on the displayed value. This includes: square, square root, reciprocal, trigonometric, logarithmic, exponential, factorial, absolute value, integer, fractional part and conversion keys.
2. Universal power and roots keys: [y^x] and [INV] [y^x].
3. Multiplication and division keys.
4. Addition and subtraction keys.
5. The equals key [=] completes all pending operations.

DATA ENTRY KEYS

The keys on the right side of the calculator are positioned to help you to remember the AOS™ hierarchy.

[y^x]
[÷]
[X]
[−]
[+]
[=]

Operations with the same priority level are performed left to right.
To illustrate the Algebraic Operating System, consider the following example:
4 ÷ 5 + 7 × 3 × 4 − 2 =

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operations.
4 [÷] 5	5	The division is pending.
[+]	0.8	The division is performed as the addition as a lower priority.
7 [X] 3	3	The addition and multiplication are pending.
[X]	21	The first multiplication is completed.
4 [−]	84.8	The second multiplication, then the addition are performed. The subtraction is pending.
2 [=]	82.8	The equals sign completes the calculation.

[(], [)] — **Parentheses keys.**
Some calculations require specifying the exact order in which numbers and operations are to be grouped. Placing a series of numbers and operations in parentheses indicates that they are to be evaluated first instead of in the order directed by the normal algebraic hierarchy. Within each set of parentheses, the calculator operates according to the rules of algebraic hierarchy. Use the parentheses if there is any doubt about how the calculator will handle an expression.

DATA ENTRY KEYS

Example: 7 × (3 + 4) = 49

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operations.
7 [X] [(] 3 [+] 4 [)]	7	Addition result, multiplication pending
[=]	49	Result.

The open parenthesis can also be used to enter a number already in the display into a new calculation, thus supplying a missing number, as shown in the following example.

Example: 4 − (4 + 2) = −2

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operations
4 [−] [(] [+]	4	Enter the number 4. The open parenthesis followed by a [+] causes the 4 to be repeated.
2 [)] [=]	−2	Answer.

The close parenthesis does not supply a missing number. It does, however, complete the operation started with the most recent open parenthesis. If no open parenthesis is pending, the close parenthesis completes all pending operations.

There are limits to how many operations and associated numbers can be pending. As many as fifteen parentheses can be open at any one time and four operations can be pending, but only in the most complex situations will these limits be approached. If you go beyond these limits, an error message is displayed.

You may see equations or expressions written with parentheses used to indicate implied multiplication:
(2 + 1)(3 + 2) = 15. The calculator does not perform implied multiplications. You must enter the multiplication sign.
[(] 2 [+] 1 [)] [X] [(] 3 [+] 2 [)] [=]

Here is an example on using parentheses.

Example:
Evaluate $\dfrac{(8 + 9) \times -19}{(3 + 10) \div 7} = -173.92308$

In problems of this type, the calculator must evaluate the entire numerator, then divide by the entire denominator. To be sure that this takes place, add an extra set of parentheses around the numerator and demominator.

Press	Display	Comments
ON/C ON/C	0	Clear display and pending operations.
((8 + 9) X	17	(8 + 9) displayed.
19 +/-) +	-323	The value of the numerator.
((3 + 10		
) + 7)	1.8571429	The value of the denominator.
=	-173.92308	The result.

DISPLAY FORMATS.

Even though the calculator has a display and entry limit of eight digits, the internal display register holds calculated results to 11 digits for greater accuracy in following calculations. The value displayed is rounded to eight digits.

In addition to the standard eight-digit floating decimal display, there are several other display formats available to increase the versatility of the calculator.

EE – Scientific Notation Key

Many scientific and engineering calculations involve very large or small numbers which can be awkward to manipulate. Scientific notation makes these values easier to handle. Any number can be expressed in scientific notation as a base value (mantissa) times 10 raised to some power (exponent). For example, the value 1,050,000 is expressed as $1.05 \times .10^6$ in scientific notation. The sign (+ or –) of the exponent indicates where the decimal point is placed when the number is written in standard form. A positive exponent indicates that the decimal is shifted to the right to display the number is standard format, and a negative exponent indicates that it is shifted to the left. The value of the exponent gives the number of places the decimal point has to be moved. The following table shows some numbers expressed in both standard form and scientific notation.

Standard Notation	Scientific Notation
6,789	6.789×10^3
.0000000021	2.1×10^{-9}
-16,389,043	-1.6389043×10^7
8.775	8.775×10^0

Your calculator's scientific notation allows you to use numbers as small as $\pm 1 \times 10^{-99}$ and as large as $\pm 9.9999999 \times 10^{99}$. Numbers smaller than $\pm 1 \times 10^{-7}$ and larger than $\pm 9.9999999 \times 10^7$ must be entered into the calculator in scientific notation. If calculations exceed these limits, the results are automatically displayed in scientific notation.

To enter a number in scientific notation, first enter the mantissa, pressing +/- if it is negative. Press EE and "00" appears at the right of the display. Then enter the exponent, pressing +/- if it is negative. If you press a wrong digit key when entering the exponent, press the correct digits and the calculator replaces the old digits with the last digits entered.

Example: Suppose you wanted to enter 6.023×10^{23} but accidentally press the exponent digits in the reverse order.

Press	Display	Comments
ON/C ON/C	0	Clear display and pending operations.
6.023 EE 32	6.023 32	The exponent digits are reversed.
3	6.023 23	The new entry shifts the exponents and corrects the error.

Regardless of how a mantissa is entered in scientific notation, the calculator normalizes the number, displaying a single digit to the left of the decimal point, when any function or operation key is pressed.

After pressing the EE key, all results are displayed in scientific notation. To remove the scientific notation format or convert a number to standard form, press INV EE. Scientific notation is also removed by ON/C or turning the calculator off and back on. If the number displayed is outside the range $\pm 1 \times 10^{-7}$ to $\pm 9.9999999 \times 10^7$, the calculator returns to the standard format only when a calculated result or entry is in the displayable range.

Example: Enter 32.5×10^4 in scientific notation and change it to standard notation.

Press	Display	Comments
ON/C ON/C	0	Clear display and pending operations
32.5 EE 4	32.5 04	Entry
=	3.25 05	Scientific notation
INV EE	325000	Standard notation

Data entered in standard form may be mixed with data in scientific notation for quicker calculations. The calculator converts the standard numbers and displays the results in scientific notation.

Example: $3.2 \times 10^3 + 12575.321 = 15775.321$

Press	Display	Comments
ON/C ON/C	0	Clear display and pending operations
3.2 EE 3	3.2 03	Enter first number
+ 12575.321	12575.321	Add second number
=	1.5775321 04	Result in scientific notation
INV EE	15775.321	Convert result to standard notation

[2nd] [Fix] n — Fix Decimal Key

In some calculations you may wish to display a fixed number of digits following the decimal point in standard, or scientific notation. Pressing [2nd] [Fix] n directs the calculator to round the display to n decimal places. The internal display register still retains the full 11 digits accuracy for use in subsequent calculations.

Fixed decimal format can be used in conjunction with scientific notation. In that case, [2nd] [Fix] n sets the number of decimal places displayed in the mantissa.

If the calculator is in the fixed-decimal format and a calculated result exceeds $\pm 9.9999999 \times 10^7$ or goes below $\pm 1 \times 10^{-7}$, the display automatically converts to scientific notation and the fixed-decimal format is ignored. The display returns to the fixed-decimal format when scientific notation is no longer necessary.

Reset the calculator to the floating decimal point with [INV] [2nd] [Fix], [2nd] [Fix] [8], [2nd] [Fix] [9], or by turning the calculator off and back on.

Example:

Press	Display
[ON/C] [ON/C]	0.
[2nd] [π]	3.1415927
[2nd] [Fix] 2	3.14
[2nd] [Fix] 4	3.1416
[INV] [2nd] [Fix]	3.1415927

Section 3 — Algebraic keys.

In this section you will get acquainted will the powerful algebraic functions included in your TI 57-II .

NOTE: Limits on the range and accuracy of these keys are discussed in the appendix. Values out of the range cause the display of an ERROR message.

[1/x], [x²], [√x] — Reciprocal, Square and Square root keys.

These three keys act immediately on the number in the display (x), and do not affect pending calculations.

[1/x] divides the displayed number into one.
[x²] calculates the square of the displayed number, multiplying it by itself.
[√x] calculates the square root of the number in the display. The square root of a number (x) is labeled $\sqrt{x}$ and is such that $\sqrt{x}$ times $\sqrt{x}$ equals x. The number in the display must be positive.

[yˣ], [INV] [yˣ] — Universal Power and Root Keys

[yˣ] is the universal power key. It raises any positive number to any power. To use this key:

- Enter the number to be raised to a power ("y").
- Press [yˣ].
- Enter the power ("x").
- Press [=]

Example: Calculate $3.1897^{4.7343}$

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operation.
3.1897 [yˣ]	3.1897	"y" value.
4.7343	4.7343	"x" value.
[=]	242.60674	Result: "y^x".

[INV] [yˣ] is the universal root sequence. It allows you to take any root of any positive number ($\sqrt[x]{y}$). To use this key:

- Enter the number to take the root of ("y")
- Press [INV] [yˣ]
- Enter the root to be taken ("x")
- Press [=]

Example: Calculate $\sqrt[3.871]{21.496}$

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operations
21.496 [INV] [yˣ]	21.496	"y" value
3.871	3.871	"x" value
[=]	2.2089685	Result $\sqrt[x]{y}$

NOTE: See appendix B for accuracy information.

Remember that [yˣ] and [INV] [yˣ] have the higher priority in the AOS™ hierarchy. Consider the following example:

$4 + 6 \times 2^3 \div 7 = 10.857143$

Press	Display	Comments
[ON/C] [ON/C]	0	
[4] [+]	4	
[6] [X]	6	Addition is pending.
[2] [yˣ]	2	Addition and multiplication are pending
[3] [÷]	48	[yˣ] and multiplication are executed.
[7] [=]	10.857143	[=] completes the calculation.

[lnx], [log], [INV] [lnx], [INV] [log] — Logarithm and antilogarithm keys.
Logarithms are mathematical functions used in a variety of technical and theoretical calculations. In addition, they form an important part of many mathematical "models" of natural phenomena. The logarithm keys give immediate access to the "log" of any number without having to locate it in a table.

[lnx] — The natural Logarithm key
Immediately displays the natural logarithm (base e = 2.7182818) of the displayed number which must be greater than zero.

[log] — The Common Logarithm key
Immediately displays the common logarithm (base 10) of the number in the display (which must be greater than zero).

[INV] [lnx] — e^x Key Sequence
Raises e to the power of the number in the display (natural antilogarithm).

[INV] [log] — 10^x Key Sequence
Raises 10 to the power of the displayed number (common antilogarithm).

Example: Calculate log 15.32 ln 203.451, $e^{-.69315}$, 10^{π}.

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operations.
15.32 [log]	1.1852588	
203.451 [lnx]	5.3154252	
.69315 [+/-] [INV] [lnx]	0.4999986	
[2nd] [π] [INV] [log]	1385.4557	

[2nd] [x!] — Factorial Key Sequence.
The factorial of any positive integer number (x) is written x!, and is equal to: 1 × 2 × 3 × ...x. By definition 0! is equal to 1
[2nd] [x!] acts on the number in the display and calculates the factorial of this number, which must be any positive integer less than 70.

[2nd] [|x|], [2nd] [Intg], [2nd] [Frac] — Number Portion Keys.
[2nd] [|x|] calculates and displays the absolute value of the number in the display. The absolute value of a number is the magnitude of the number, regardless of its sign. Thus the result of [2nd] [|x|] is always a positive number.

[2nd] [Intg] truncates the number in the display register by discarding its decimal part.

NOTE: For negative numbers, you need to subtract 1 to the displayed result to follow the mathematical definition which states that the integer part of a number is equal to the next integer number less or equal than the argument. Thus the integer part of −4.5 is −5, when your calculator will give you −4.

[2nd] [Frac] displays the fractional part of the number in the display register and discards the integer part.

NOTE: The [2nd] [Intg] and [2nd] [Frac] keys operate on the 11 internal digits in the display register, not the 8 digits shown in the display. This means that when [2nd] [Intg] is pressed and 4.9999999999 is in the display register internally (which rounds to a value of 5 in the display), that 4 will be the integer that remains in the display. Similarily [2nd] [Frac] will give a display of 1, with the actual value being .9999999999.

[DRG], [INV] [DRG], [2nd] [DRG▸], [INV] [2nd] [DRG▸] — Degree, radian, and grad keys.
The calculator handles a variety of calculations involving angles, such as the trigonometric functions and polar/rectangular conversions. When performing these calculations, select any one of the three common units for angular measure.

Degrees are each equal to 1÷360 of a circle. A right angle equals 90°.

Radians are each equal to 1÷2 π of a circle. A right angle equals π ÷2. radians

Grads are each equal to 1÷400 of a circle. A right angle equals 100 grads.

The calculator is always in degree mode when it is turned on. Pressing [DRG] changes it to radian mode, indicated by RAD in the display. Pressing [DRG] again changes it to grad mode, indicated by GRAD in the display. Pressing [DRG] again returns the calculator to degree mode. You may also go through the modes in reverse order — from degrees to grads to radians and back to degrees — by pressing the [INV] [DRG] key.

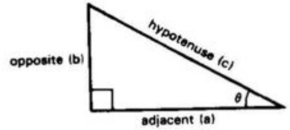
The [2nd] [DRG▸] key changes the mode displayed, and additionally converts the number in the display to the new units. Thus 90 in the degree mode followed by [2nd] [DRG▸] changes the mode to radians and the display to 1.5707963 (π÷2). Pressing [2nd] [DRG▸] again changes the mode to grads and the display to 100. You may also go through the modes and values in reverse order—from degrees to grads to radians and back to degrees—by pressing the [INV] [2nd] [DRG▸] key.

[sin], [cos], [tan], [INV] [sin], [INV] [cos], [INV] [cos] — Trigonometric keys.
The trigonometric keys [sin], [cos] and [tan] calculate the sine, cosine and tangent of the angle in the display, with the angle measured in the units selected with the [DRG], [INV] [DRG], [2nd] [DRG▸], or [INV] [2nd] [DRG▸] keys. The trigonometric functions relate the angles and sides of a right triangle as shown below.

$\sin\theta = \frac{b}{c}$

$\cos\theta = \frac{a}{c}$

$\tan\theta = \frac{b}{a}$



The inverse functions of the trigonometric keys give the angle, in the units selected, whose sine, cosine or tangent is in the display. [INV] [sin] calculates the arcsine ($\sin^{-1}$), [INV] [cos] calculates the arccosine ($\cos^{-1}$), and [INV] [tan] calculates the arctangent ($\tan^{-1}$).

Note the following ranges from the use of the inverse trigonometric functions.

Arc Function	Ranges of Resultant Angle
arcsin x	0 to 90°, π ÷2 radians, or 100G
arcsin −x	0 to 90°, − π÷2 radians, or −100G
arccos x	0 to 90°, π ÷2 radians, or 100G
arccos −x	90° to 180°, π ÷2 to π radians, or 100G to 200G
arctan x	0 to 90°, π ÷2 radians, or 100G
arctan −x	0 to 90°, −π ÷2 radians, or −100G

[2nd] [DMS-DD], [INV] [2nd] [DMS-DD] — Degrees/Minutes/Seconds to decimal degrees conversion keys.

There are two ways of representing an angle in degrees. One method is to use the decimal degree format DD.dd. Here DD represents the integer portion of the angle and dd represents the fraction portion written as a decimal. Up to 8 digits may be entered.

The second method is to use the degree/minute/second format DD.MMSSss. Again DD represents the whole angle. MM represents minutes and SS denotes seconds. For greater accuracy, fractional seconds may be entered in the ss position. The decimal point separates degrees from minutes.

To convert from the degree/minutes/second format to degrees, enter the angle in the display as DD.MMSSss and press [2nd] [DMS-DD]. Pressing [INV] [2nd] [DMS-DD] reverses the conversion process and converts decimal degrees to degrees, minutes, and seconds. Two digits should always be entered for minutes and two for seconds. Trailing zeros need not be entered.

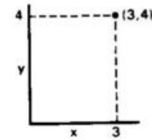
Example: Convert 3°1′30.456″ to decimal degrees and back.

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operations.
3.0130456 [2nd] [DMS-DD]	3.0251267	Answer in decimal degrees
[INV] [2nd] [DMS-DD]	3.0130456	Answer returned to degrees/minutes/seconds.

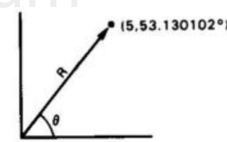
The same process can be used to convert hours, minutes, and seconds to decimal hours and vice versa.

[2nd] [P⇄R], [INV] [2nd] [P⇄R] — Polar/Rectangular conversion keys

The rectangular coordinate system describes where points are placed on a grid with a pair of numbers. The first, the x-coordinate, describes the distance of the point from the y-axis, which is a vertical line. The second, the y-coordinate, describes the distance of the point from the x-axis, which is a horizontal line. The following shows the point described in rectangular coordinates as (3,4).



The polar system of coordinates describes a point in terms of a line drawn from a centre to the point. It also uses a pair of numbers. The first number is the length of the line, labeled R. The second is the number of degrees the line is from horizontal, labeled theta (θ). The following shows the same point, but described as (5,53.130102°).



The conversion from polar to rectangular coordinates and back involves some detailed arithmetic. Fortunately, the calculator can perform these calculations.

- To convert from polar to rectangular coordinates, follow these steps:

Enter the R value.

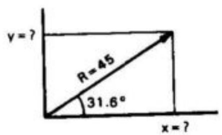
Press [x⇄t]
Enter the θ value
Press [2nd] [P⇄R]

The y- coordinate is displayed.
Press [x⇌t] to get the x-coordinate in the display.

- To convert from rectangular to polar coordinates, follow these steps:
  Enter the x-coordinate
  Press [x⇌t]
  Enter the y-coordinate
  Press [INV] [2nd] [P⇌R]

The θ value is displayed in the units selected by the [DRG] key.
Press [x⇌t] to get the r-value in the display.
The range of θ is from $+180°$ to $-180°$, $+\pi$ to $-\pi$ radians and 200 to -200 grads.

Caution: The [2nd] [P⇌R] and [INV] [2nd] [P⇌R] sequences clear the pending operations.



Example:
Convert R = 45 metres, $\theta = 31{,}6$ degrees to rectangular coordinates.

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operations. Press [DRG] to select the degrees mode.
45 [x⇌t] 31.6 [2nd] [P⇌R]	23.579366	y-coordinate value.
[x⇌t]	38.327712	x-coordinate value.

Section 4 — Memory operations

The TI 57-II may be set to have a maximum of seven user data memories, numbered 0 to 6, which can be used to store numbers you may need to use later (intermediate results, parameters).
Because of the calculator's Constant Memory™ feature, the contents of the user data memories are not lost when the calculator is turned off, thus allowing you to keep data in your calculator.
The following section describes the memory operations.

[2nd] [Part] m — Memory partition.
The [2nd] [Part] key sequence sets the partitioning of the calculator's memory between user data memories and user program memories. This sequence has to be followed by the number of user data memories, m, that you wish to have available (m can vary from 1 to 7).
For example to allow using all seven data memories, you have to press [2nd] [Part] 7. Please refer to chapter 2 to get more details on that important feature.

[STO] m — Store memory key.
The [STO] m key allows you to store the displayed number into the user memory specified with m, erasing automatically any number previously stored in that memory. The displayed number is not affected and remains available.

[RCL] m — Recall memory key.
The [RCL] m key recalls to the display the number in user data memory m. The contents of this memory is not altered. The number that was in the display is lost.

[EXC] m — Exchange memory key.
The [EXC] m key exchanges the value in the display with the value in user data memory m. This allows you to recall the contents of a memory without losing the displayed number.

[2nd] [CM] — Clear memories keys.
The [2nd] [CM] key clears all the user data memories as defined with the current partitioning. The display, program memories and t-register are not affected.

Example:

Press	Display	Memory	Comments
[ON/C] [ON/C]	0	0	Clear display and pending operations.
[2nd] [CM]	0	0	Clear all memories.
3 [STO] 0	3	3	Store 3 in memory 0.
5	5	3	Enter 5 in display.
[EXC] 0	3	5	Exchange display and memory 0.
[RCL] 0	5	5	Recall memory 0 to display.

NOTE: To clear any single data memory press the following sequence: [ON/C] [STO] m (with m being the number of the data memory you want to clear).

MEMORY ARITHMETIC

In addition to the basic memory functions that you just saw, the TI 57-II allows you to perform arithmetic on the numbers stored in a memory, without affecting the displayed number and the calculations in progress. To use these sequences:

- Enter the number that is to operate on the memory value.
- Press [STO].
- Enter the operation to be performed.
- Enter the number of the memory to be used.

NOTE: because of the calculator's Constant Memory™ feature, the user data memories are not cleared when the calculator is turned off. Be sure to press [ON/C] [STO] m initially to clear the desired user data memory before using any of the following key sequences. [2nd] [illegible] clears all the user data memories defined by the current partitioning.

[STO] [+] m algebraically adds the displayed value to the contents of user data memory m.

[STO] [−] m algebraically subtracts the displayed value from the contents of user data memory m.

[STO] [X] m multiplies the contents of user data memory m by the displayed value.

[STO] [÷] m divides the contents of user data memory m by the displayed value.

[STO] [y^x] m raises the contents of user data memory m to the power in the display.

[STO] [INV] [y^x] m takes the root indicated by the number in the display of the value in user data memory m.

Example:

	28.3 × 7 =	198.1
	173 + 16 =	189
	31 − 42 + 7,8 =	− 3.2
	Total	383.9

Press	Display	Comments
[ON/C] [ON/C] [STO] 0	0	Clear display, pending operations, and memory 0.
28.3 [X] 7 [=] [STO] [+] 0	198.1	Result of first problem added to user data Memory 0.
173 [+] 16 [=] [STO] [+] 0	189	Result of second problem added to user data memory 0.
31 [−] 42 [+] 7,8 [=] [STO] [+] 0	− 3.2	Result of third problem added to user data memory 0.
[RCL] 0	383.9	Sum of the problems.

[x⇄t] — "x" Exchange with "t" key.
This is a special key that exchanges the displayed number with the contents of a special data register which is called the "t" register or "test" register. This "t" register is used for polar-rectangular conversions and for decision-making operations as described in Chapter 2.
This memory can also be used as a data memory.

[2nd] [C.t] — Clear "t" register.
The [2nd] [C.t] key allows you to clear the contents of the "t" or "test" register.

NOTE: The value stored into the t-register can only be altered by the following keys: [x⇄t], [2nd] [C.t], [2nd] [P→R] and [INV] [2nd] [P→R].

Chapter II — Using your TI 57-II as a computer

Introduction.

We have now seen how to use all the scientific functions of the TI 57-II, to perform computations right from the keyboard. We are going to approach in chapter 2 the "programmable" side of this calculator, which makes it a real computer.
When you perform computations from the keyboard, you spend most of the time is thinking which key you will have to press next and in looking for keys on the keyboard. When you have repetitive or iterative calculations to perform, a programmable calculator can save you a lot of time. You teach it how to perform the calculation, and then, upon a single key push, it will perform calculations, test hypotheses, take decisions and give you quick and accurate results again and again.
The following sections show you how to use the programming power of your calculator.

Section 1 — Writing a program:

Now, let's have a look at how to program the TI 57-II.
Consider the following problem: you want to evaluate the value of the function: $F(x) = \frac{1}{x^2 + 1}$

To calculate this expression from the keyboard, you would:
- First type in the "x" value,
- Then press:

[x²] [+] [1] [=] [1/x]
To avoid repeating this key sequence any time you want to evaluate this expression, let's enter it as a program into your calculator.
Here is what you need to do:

Press	Display		Comments
[2nd] [Part] 1	48.1		
[LRN] [2nd] [CP]	--	St	Get into "learn" mode, clear memory
[x²]	34	00	
[+]	85	01	
1	01	02	
[=]	95	03	
[1/x]	33	04	
[R/S]	13	05	
[RST]	21	06	
[LRN]	48.1		Exit "learn mode"

Your program is entered in the program memory. Now you just have to execute it.

Press	Display	Comments
[RST] 1 [R/S]	0.5	F(x = 1) = 0.5
2 [R/S]	0.2	F(x = 2) = 0.2
10 [R/S]	0.009901	F(x = 10) = 0.009901

You can notice that to compute f(x), only a few keystrokes are needed.
Now let's come back to the program entry.

[LRN] — Learn key.
Pressing the [LRN] key once puts the calculator in a special mode of operation, called learn mode in which the calculator is ready to "learn" a program.

Caution: If the memory partitioning is such that there are no program steps available, the calculator will not enter the learn mode and an error message will be displayed.

To be sure to start with a clean empty program memory, select the learn mode by pressing [LRN] [2nd] [CP]. ([2nd] [CP] removes any previous program and assures that the program is keyed in beginning at location 00). You get the following display.

--	St

"St" indicates that you are pointing to the starting location of the program memory. The calculator is ready to "learn" a program. You just need to key in the functions you want to enter in your program, in the same order as if you were doing calculations from the keyboard.

Press	Display	
[LRN] [2nd] [CP]	--	St
[x²]	34	00

As you can see from that example, the learn mode display has a special format, showing the step number and the contents of that step.

34 ↓ key code

00 ↓ program location

The 2 digits located in the exponent location indicate the step number you just entered into the machine. The digits to the left of the display indicate the instruction code contained in that step. The key code identifies which keystroke has been saved into the program memory. Here 34 is the key code associated to the [x²] key. The key codes are discussed further on in this section.

Keeping track of the program location is the task of the *program counter*. This counter is automatically incremented any time you press a new key. Thus the display will always show which step you just keyed in.

Pressing [LRN] a second time will take you out of the learn mode, restoring the display to its original state.

[RST] – **Reset key:**

The [RST] key brings the program counter back to the starting location of the program memory. This key can be used as part of a program or right from the keyboard when out of the learn mode.

- When encountered in a program, this instruction will move the program counter back to the program starting location and the program execution continues from there.
- Used when out of the "learn" mode, [RST] sets the program counter back to the program starting location.

NOTE: The [RST] keys also clears the subroutine return register.

[R/S] – **Run/Stop key:**

The [R/S] key starts or stops the program depending on the status of the calculator (a running program will be stopped and a stoped program will start running from the location of the program counter). The "RUN" indication shows whether a program is running or not. This key can be used as an instruction inside a program or as a command, right from keyboard when out of the learn mode.

- When encountered in a program, the [R/S] instruction simply stops the program execution and displays whatever is in the display register. This is useful to see intermediate results, to enter new parameters for the program, or to halt the action when the program is finished.
- When out of "learn" mode, [R/S] either stops the program execution (the key has to be held down momentarily) or resumes the execution from the location indicated by the program counter.

NOTE: To run a program right from the beginning, you need to press [RST] then [R/S]

Caution: Any program has to be ended with either a [R/S] instruction or a [RST] instruction or a branching instruction (these instructions will be discussed later). Otherwise, the program counter will keep on going up to the end of the program memory, then go back to the beginning of the program, and continues the execution from there and so on without stopping.

If this happens, unused locations, between the end of the program and the end of the program memory, are considered as "0", thus loading the display with zero.

When the calculator is running a user program the automatic power down is disabled, thus allowing the execution of very long programs.

[2nd] [Pause] – **Pause key:**

The [2nd] [Pause] key, when encountered in a program causes the current value of the display register to be displayed for one or two seconds, letting you see intermediate values without stopping the program.

[2nd] [CP] – **Clear program key:**

Pressing [2nd] [CP] while in "learn" mode clears the program memory of your TI 57-II and brings the program counter back to the program starting location. The user data memories and the t-register are not affected.

This key has no action if it is used when out of the "learn" mode.

KEY CODES

When you are entering a program into the calculator's program memory, we have seen that the display shows the program location and the key code associated to the function that was just keyed in. In order to facilitate the reading of these key codes, they have been assigned the following way.

- The code for any number key is just the number itself (00 to 09)
- The code for any first function key is a two digit number: the first digit is the row of the key, the second digit is its column (rows are numbered 1 to 9 from top to bottom, colums from 1 to 5 left to right). For second functions, column numbers are respectively 6, 7, 8, 9 and 0 (from left to right).

As an example here are some key codes:

Key	Key code	Row number	Column number
[sin]	42	4	2
[2nd] [Pause]	96	9	1
[2nd] [x!]	40	4	5

The keys which cannot be used in programming do not have a code (for Example: [LRN], [SST] ...).

MERGED INSTRUCTION CODES

In order to take the best out of the program memory available, one program location will always contain a complete instruction, including up to four keystrokes.

- Second functions are merged as shown on the following diagram.
- The [INV] key is also merged with the key to which it applies, and is shown as a negative sign in front of the standard code of that key. For example [INV] [sin] will be shown as "– 42".

- Functions followed by a number, such as [STO], [RCL], etc..., are also merged with this number and are shown in the following way:
  "key code • number code".
  For example: [STO] 2 will be shown as: "61.02".
- Memory arithmetic instructions, such as [STO][+], [STO][X], etc..., are merged the following way:

61 • XX • mm

[STO] code • Operation code • number code

For example:

[STO][+] 2 will be shown as "61.85.02"

[STO][INV][y^x] 0 will be shown as "– 61.45.00"

NOTE: A list of the key codes in numeric order is provided in the appendix C.

Caution: Make sure to complete the instructions requiring to be followed by a number. Otherwise, they will be ignored when the program will run.

The following diagram shows all the existing keycodes

Row	1	2	3	4	5
1	[2nd] **	[INV] **	[R/S] 13	[OFF] *	[ON/C] 15
2	[x=t] 26	[x≥t] 27	[SBR] 28	[Dsz] 29	[del] *
	[RST] 21	[GTO] 22	[LBL] 23	[BST] *	[SST] *
3	[log] 31	[lnx] 32	[1/x] 33	[x^2] 34	[√‾] 35
4	[DRG▸] 46	[P▸R] 47	[DMS-DD] 48	[π] 49	[x!] 46
	[DRG] 41	[sin] 42	[cos] 43	[tan] 44	[y^x] 45
5	[C.t] 56	[Fix] 57	[Intg] 58	[Frac] 59	[|x|] 50
	[x⇄t] 51	[EE] 52	[(] 53	[)] 54	[+] 55
6	[Part] *				
	[STO] 61	[7] 07	[8] 08	[9] 09	[X] 65
7	[CM] 76				
	[RCL] 71	[4] 04	[5] 05	[6] 06	[–] 75
8	[CP] *				
	[EXC] 81	[1] 01	[2] 02	[3] 03	[+] 85
9	[Pause] 96				
	[LRN] *	[0] 00	[•] 93	[+/–] 94	[=] 95
Column	1	2	3	4	5

* No Key code. These keys cannot be put in a program.

** These keys are merged with the following keystroke.

Section 2 — Editing a program.

We have now seen how we can enter simple programs into your TI 57-II. Before going any further, let's have a look to the editing features provided by this calculator, features designed to help you check that your program works correctly.

With these special keys, you will be able to read back and correct the steps you have entered in the program, delete useless steps and insert missing steps. This provides also an efficient debugging (or correcting) technique: the step by step execution.

This section is describing all these features.

[SST], [BST] Single step and Back step keys:

These two keys allow you, when you are in "learn" mode, to step through your program either forward with the [SST] key or backward with the [BST] key, thus allowing to check stored key codes. Note that these two keys have no key codes, as they cannot be entered as program steps.

The [SST] key allows to step forward in your program. Any time you press this key, the program counter is incremented by one and the corresponding step appears in the display.

The [BST] key allows to step backward in your program. As a result, the program counter is decremented by one, showing the previous step in the display.

NOTE: The [SST] key can also be used when out of "learn" mode.

Any time you press that key, the instruction pointed to by the program pointer is executed and the program pointer is updated. This feature is very important as it allows you to execute your program one step at a time, giving you the opportunity to compare what your program is actually doing with what you expected it to do. This is a very powerful debugging technique, allowing to check a program and to understand how your TI 57-II works.

The [BST] key has no effect if used when out of "learn" mode.

Example

Let's take a simple example to better understand the use of these editing keys.

We will enter a program that displays the squares of the counting numbers (0, 1, 2, 3, ... etc.). Key in the program as follows:

Press	Display		Comments
ON/C ON/C		0	Clear pending operation.
2nd Part 1		48.1	Partition memory.
ON/C		0	
LRN 2nd CP	--	St	Enter learn mode and clear the program memory.
RCL 0	71.00	00	
x^2	34	01	
2nd Pause	96	02	
1	01	03	
STO + 0	61.85.00	04	
RST	21	05	
LRN		0	
RST		0	

Now that the program is entered, let's edit it:

Press	Display		Comments
LRN	--	St	
SST	71.00	00	RCL 0 code - Step 00
SST	34	01	x^2 code - Step 01
SST	96	02	2nd Pause code - Step 02
SST	01	03	1 code - Step 03
BST	96	02	2nd Pause code - Step 02
BST	34	01	x^2 code - Step 01
LRN RST		0	Now let's execute it step by step
2 STO 0		2	Start counting from 2
SST		2	Recall memory 0
SST		4	X^2
SST		4	Pause
SST		1	Enter 1 in display
SST		1	Add 1 in memory zero
SST		1	Reset the program counter
SST		3	Recall memory zero
SST		9	X^2

If you experience problems with your programs, this feature will allow you to understand where it comes from.

Caution: In "learn", mode if SST is pressed when the program pointer is pointing to the last program step available in the current partitioning, the calculator will wrap-around to the starting location of the program. Reciprocally, BST will cause a wrap-around from the program starting location to the last program step available.

2nd Del – Delete key:
When in "learn" mode, the 2nd Del key allows you to delete the step currently in the display, moving all the following intructions up by one location.

You just have to select to the exact instruction you want to delete (with BST and SST) and then press 2nd Del.

Automatic step insertion:
The TI 57-II has an automatic insertion feature allowing to insert steps into your program.
Therefore to insert a step, the following should be done:

- Enter the "learn" mode.
- Position the program pointer at the step that has to be followed by the step you want to insert.
- Key in the step to insert.

Caution: When you insert a step in a program, all the following steps are moved down by one location. Therefore the contents of the last location of the program memory is lost.

Modifying a program step:
If you want to modify an incorrect program step you must:

- Enter the "learn" mode
- Position the program pointer on the step that has to be modified.
- Delete the step with the 2nd Del key.
- Key in the correct instruction.

Example:
Let's come back to our simple program displaying the squares of the counting numbers. We are going to modify it to display the reciprocal of these numbers.
Press LRN and SST (or BST) until you point the step number 01

Press	Display		Comments
	34	01	Step that we want to modify
2nd Del	71.00	00	The x^2 has been wiped off.
1/x	33	01	Automatic insertion of the 1/x function.
BST	71.00	00	Now let's look at the final program
SST	33	01	
SST	96	02	
SST	01	03	
SST	61.85.00	04	
SST	21	05	
LRN		0	

Section 3 — Partitioning the memory.

The TI 57-II has seven user memories (numbered 0 to 6).
These memories are available for either data storage (using the [STO] m or [EXC] m keys sequences) or for programming purpose.
In order to optimize the use of these memories, the TI 57-II allows you to select how many memories you want for data storage purpose, thus allocating the remaining memories to program storage. This feature is named:

Partitioning of the memory.

This section describes how you can partition the memory of your TI 57-II to increase its efficency.

[2nd] [Part] m — Memory partition key:
The [2nd] [Part] key sets the partitioning of the calculator's memory, between data storage memories and program storage memories. The key is followed by the number of user data memories, m, that you wish to have available. As a result, the display will show the number of program steps and user data memories available with the following format: ss.m where ss is the number of program steps available and m is the number of user memories available.
The calculator can be set with a maximum of seven user data memories, and no program steps or a minimum of one user data memory and 48 program steps.

Each data memory converted to program memory is equivalent to eight program steps (or instructions).
The following table shows you all the partitions available.

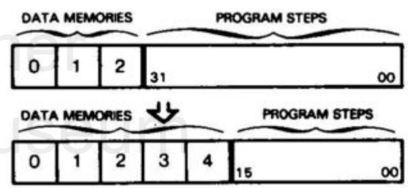
[2nd] [Part] m	Number of data memories	Data memories labels	Number of program steps	Program step labels
m = 1	1	0	48	00 to 47
m = 2	2	0 to 1	40	00 to 39
m = 3	3	0 to 2	32	00 to 31
m = 4	4	0 to 3	24	00 to 23
m = 5	5	0 to 4	16	00 to 15
m = 6	6	0 to 5	8	00 to 07
m = 7	7	0 to 6	0	—

The following example illustrates how to use the above table:
Set the number of user data memories to three by pressing [2nd] [Part] 3. Those memories are numbered 0,1 and 2. In addition there are 32 program steps available, numbered 00 to 31.

Caution: When you repartition the memory the contents of the data memories transformed into program step or the contents of program steps transformed into data memories are lost. If the program pointer is at a location which becomes user data memory, it is reset to "St".

Example:
Let's consider changing the partitioning from 3 data memories to 5
Before changing the partition, the calculator has 3 data memories labeled 0 to 2 and 32 program steps labeled 00 to 31.

Press	Display	Comments
[ON/C] [ON/C]	0	Clear display and pending operations.
[2nd] [Part] 5	16.5	Indicates that 16 program steps are available (00 to 15) and 5 data memories 0 to 4).



Above diagram illustrates the modification of the calculator memory when moving from 3 data memories to 5. Program steps 16 to 31 are lost. Memories 3 and 4 come with 0 inside. Program steps 00 to 15 and memories 0 through 2 are not affected.
Reciprocally, if you now move back to 3 data memories, you will go back to the previous configuration thus losing the contents of memories 3 and 4. Program steps 16 to 31 are available and contain the code 00 (zero). Program steps 00 to 15 and memories 0 through 2 remain unaffected.

NOTE: [2nd] [Part] 0, 8 or 9 are invalid key sequences and will result in the display of an "ERROR" message.

Warning: Before using [2nd] [Part] m key, remove the scientific mode and the fixed decimal mode by pressing [INV] [EE], [INV] [2nd] [Fix], in order to go back to standard display format. Otherwise, you will not get the number of program steps and user data memories with the ss.m format.

[INV] [2nd] [Part] — Display current partitioning:
The [INV] [2nd] [Part] key sequence recalls to the display the current partitioning with the ss.m format where ss is the number of program steps available and m the number of user data memories.

Example:

Press	Display	Comments
[ON/C] [ON/C]	0	
[2nd] [Part] 1	48.1	Set partitioning to one user data memory and 48 program steps.
[ON/C]	0	Clear display.
[INV] [2nd] [Part]	48.1	Recall the current partitioning.

Section 4 — Branching instructions.

When solving a problem, it very often happens that you wish to execute the computations in a different way, depending on the parameters that you know.

For example, when converting degrees Celsius into degrees Fahrenheit:

$$t°C = \frac{5}{9}(T°F - 32)$$

You may have to convert temperatures from °C to °F or from °F to °C. So, both following equations need to be computed.

$$t°C = \frac{5}{9}(T°F - 32) \quad (1)$$

$$t°F = \frac{9}{5}(T°C + 32) \quad (2)$$

If you want to write a program able to compute equations (1) and (2), you need to be able to go and execute the correct computation:

You need to *branch* to some part of the program. The branching instructions of the TI57LCD include basically two instructions:

- The "GOTO" instruction telling the calculator where the execution has to continue.
- The "Label" instruction allowing to "mark" the destination location.

[LBL] n — Label key sequence.
This key allows you to label any point in the program memory area. This function operates only in "Learn" mode. The key sequence is [LBL] n, where n is any digit 0 through 9. Therefore, this lets you label up to 10 points in the program memory (Label numbers cannot be used more than once each).

The key sequence [LBL] n is merged in one program step, with the following code: 23 • 0n.

[GTO] n — Goto key sequence:
The [GTO] n sequence has for result the positioning of the program pointer to the label number n (n is any digit 0 through 9).
This key sequence can be used either as part of a program (when entered in "learn" mode), or right off the keyboard (when out of "learn" mode).

During a program execution, when the program pointer encounters a [GTO] n instruction, it goes right to label n and continues on from that point. The [GTO] n sequence is stored as one program step, with the following code: 22 • 0n.
Used when out of "Learn" mode, the [GTO] n sequence positions the program pointer to the corresponding label, thus allowing to select part of a program, but the execution won't start until you press the [R/S] key.

Caution: If the corresponding label does not exist, the [GTO] n sequence will result in an error message.

The [GTO] n sequence is called an "unconditional transfer".

Now let's key the temperature conversion program. Here is how you can proceed:

Press	Display		Comments
[2nd] [Part] 1	48.1		Set the partition to 1 data memory
[LRN] [2nd] [CP]	--	St	Enter "learn" mode and clear the program memory area.
[LBL] 0	23.00	00	Entry point for the conversion from °Fahrenheit to °Celsius.
[(]	53	01	
[−]	75	02	
[3]	03	03	
[2]	02	04	
[)]	54	05	
[X]	65	06	
[5]	05	07	
[÷]	55	08	
[9]	09	09	
[=]	95	10	
[R/S]	13	11	End of t°C computation
[LBL] 1	23.01	12	Entry point for the conversion from °Celsius to °Fahrenheit
[X]	65	13	
[9]	09	14	
[÷]	55	15	
[5]	05	16	

(Continued)

(Continuation)

[+]	85	17	
[3]	03	18	
[2]	02	19	
[=]	95	20	
[R/S]	13	21	End of T°F computation
[LRN] [RST] [ON/C]	0		

Now you select either the conversion from degrees Fahrenheit to degrees Celsius by pressing [GTO] 0 or the reverse conversion by pressing [GTO] 1. Then you just have to enter T°F (or t°C) and press the [R/S] key to start the execution.

Example:

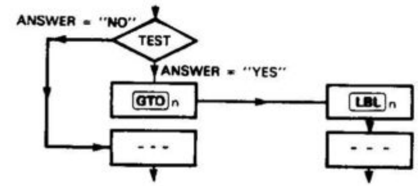
Press	Display	Comments
[ON/C] [ON/C]	0	
[GTO] 0	0	
104 [R/S]	40	(40°C are equivalent to 104 °F)
[ON/C]	0	
[GTO] 1	0	
20 [R/S]	68	(68°F are equivalent to 20°C)

Section 5 — Conditional transfers.

Let's consider another example: compute the cost of sending a parcel through the mail. The post office company tells you that the cost is £2.4 for parcels weighing less than 3 kg. Above 3 kg, the price is 0.8£/Kg.
To write a program performing these computations, you need to take a decision based on the weight of the parcel: is the weight greater or equal to 3 Kg?
Your TI 57-II can take these decisions: it will perform a test, and based on the result of this test (true or false), it will act in a different way.
The TI 57-II allows you to perform four different comparisons between the displayed value (named x) and the value loaded in the t-register (or test register). These four tests are:

[2nd] [x=t]	Is x = t? (Is x equal to t?)
[INV] [2nd] [x=t]	Is x ≠ t? (Is x not equal to t?)
[2nd] [x≥t]	Is x≥t? (Is x greater than or equal to t?
[INV] [2nd] [x≥t]	Is x<t? (Is X less than t?)

If the answer to the test is YES the program pointer goes on to the instruction immediately following the test. If the answer is NO, the program pointer skips the instruction immediately following and continues with the next one.



Above chart shows the structure of the decision-making instructions.
The instruction following a test can be any instruction but the most useful one is the [GTO]n, allowing to branch to another part of the program (as shown on this chart).

The [x⇌t] key allows to load the t-register with the number currently in the display, preparing it for a test execution.

Example:
Now let's see how we would compute our shipping cost: The mathematic equations for this problem are:

- If the weight is≥3, then the cost is: 0,8 × weight
- Otherwise, the cost is: £2.4.

Here is how we can proceed.

Press	Display		Comments
[ON/C] [ON/C]	0		
[2nd] [Part] 1	48.1		
[LRN] [2nd] [CP]	--	St	Enter "learn" mode and clear the program memory area.
[STO] 0	61.00	00	Save parcel weight
[3]	03	01	The test value is 3
[x⇌t]	51	02	Load test value
[RCL] 0	71.00	03	Recall parcel weight
[2nd] [x≥t]	27	04	Execute test
[GTO] 0	22.00	05	Go to label 0 because the weight is≥3 Kg
[2]	02	06	
[•]	93	07	
[4]	04	08	Cost of shipping = 2.4£
[=]	95	09	
[R/S]	13	10	End of that part
[RST]	21	11	Prepare for next computation
[LBL] 0	23.00	12	Go here when the weight is≥3 kg.
[RCL] 0	71.00	13	
[X]	65	14	
[•]	93	15	
[8]	08	16	

[=]	95		17	Cost of shipping = 0.8 × weight
[R/S]	13		18	Stop execution
[RST]	21		19	Prepare for next computation
[LRN]		48.1		
[ON/C]		0		

Your program is ready. To use it you just have to reset the program pointer to start the program right from the beginning.
For example: computation of the shipping cost for the following weights: 5, 2.5, 3.

Press	Display	Comments
[ON/C] [ON/C]	0	Clear pending operations
[RST]	0	Position pointer
5 [R/S]	4	Cost for a 5 kg parcel
2.5 [R/S]	2.4	Cost for a 2.5 kg parcel
3 [R/S]	2.4	Cost for a 3 kg parcel

To summarize this section:
The calculator can perform four kinds of tests as shown in the following table:

Key sequence	Test	Go to next step if	Skip next step if
[2nd] [x=t]	x = t?	x = t	x ≠ t
[INV] [2nd] [x=t]	x ≠ t?	x ≠ t	x = t
[2nd] [x≥t]	x≥t?	x≥t	x<t
[INV] [2nd] [x≥t]	x<t?	x<t	x≥t

- The calculator compares the displayed value and the contents of the test register.
- If the answer to the test is YES the calculator executes the following instruction.
- If the answer is NO, the next instruction is skipped.

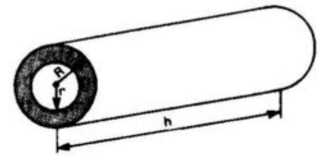
NOTE: When executed right off the key board these functions have no effect.

Caution: The comparisons are done on the displayed value (eight significant digits), and not on the 11 digits included in the display register.

This allows to have comparisons results in agreement with what is in the display. The full accuracy of the displayed number (11 digits) remains available for further computations.

Section 6 — Subroutines

In many calculations you can isolate parts that are used more than once. For example, il you want to calculate the volume of different tubes, you can compute the volume of the external cylinder (V) and the volume of the internal cylinder (v), and then calculate the difference to get the volume of the tube (Vt).



We have: $V = \pi R^2 h$ (1)

$v = \pi r^2 h$ (2)

$Vt = \pi R^2 h - \pi r^2 h$

The equations (1) and (2) are identical except the value of the radius.
In order to save program steps, there is a programming technique consisting in isolating a part of a program that will be used several times into a "mini program" called a SUBROUTINE (or subprogram).
In this example, the subroutine would be a program to compute the volume of a cylinder, by the equation $\pi x^2 h$, where x is the radius of the cylinder.
In the principal program, called "MAIN PROGRAM', any time we need to compute the volume of a cylinder we will "CALL" the subroutine, providing it with the value of the radius of the cilinder we consider.

Here is how we can show the flow of the main program.

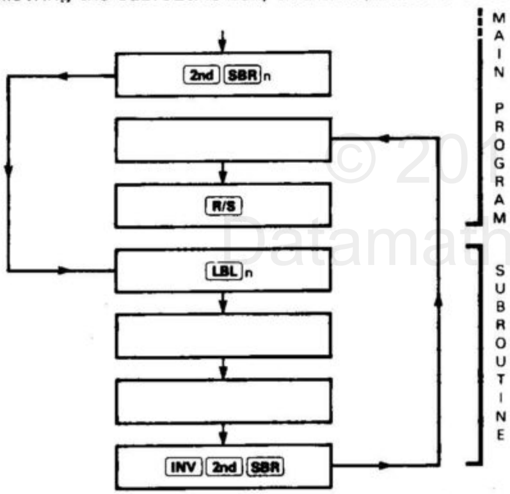
- Enter the inner radius (r)
- Call the subroutine computing the volume $v = \pi r^2 h$
- Save the result in memory 0.
- Enter the other radius (R)
- Call the subroutine computing the volume
- Subtract the volume of the inner cylinder.

Following is a description of the subroutine capabilities of the TI 57-II.

[2nd] [SBR] n — Call subroutine key sequence:
The [2nd] [SBR] n key sequence allows you to call a subroutine from a main program. The digit n indicates the label of the subroutine.
The subroutine (or subprogram) can be located anywhere but has to start with a label. The end of the subroutine has to be marked with a special instruction: [INV] [2nd] [SBR] which tells the calculator that the subroutine is completed.

[INV] [2nd] [SBR] — Subroutine return key sequence:
The [INV] [2nd] [SBR] signals the end of a subroutine and instructs the calculator to go back to the main program.
Note that [INV] [2nd] [SBR] will act as an [R/S] instuction if it is not used as part of a subroutine called with a [2nd] [SBR] n instruction.
The following diagram illustrates the way [2nd] [SBR] n and [INV] [2nd] [SBR] work.

- When [2nd] [SBR] n is encountered the calculator "notes" the number of the following program step.
- Then the program pointer goes right to label n and starts executing the subroutine, until it meets the [INV] [2nd] [SBR] instruction.
- When [INV] [2nd] [SBR] is encountered, the program pointer returns to the step following the subroutine call, and continues the execution.



Caution: Within a subroutine, it is impossible to call another subroutine, because the TI 57-II can only note one "return" address. Therefore, if this happens, the calculator will just remember the last "return" address, the first [INV] [2nd] [SBR] instruction encountered will be executed, but the calculator will stop at the next one.

NOTE: The [=] key has to be used very carefully within a subroutine, because it completes all pending operations, even those coming from the main program and it can produce results that you did not expect.

Example:
Coming back to our tube problem let's write a program to compute the volume of tubes that are 1.5 m long and with variable values for R and r.

Press	Display		Comments
[ON/C] [ON/C]			
[2nd] [Part] [1]	48.1		
[LRN] [2nd] [CP]	--	St	
[2nd] [SBR] [0]	28.00	00	Calculate the volume of the inner cylinder
[STO] [0]	61.00	01	Save it in memory 0
[R/S]	13	02	Wait for R input
[2nd] [SBR] [0]	28.00	03	Calculate outer cylinder volume
[−]	75	04	
[RCL] [0]	71.00	05	
[=]	95	06	Compute tube volume
[R/S]	13	07	Display result and stop execution
[RST]	21	08	Prepare for next calculation
[LBL] [0]	23.00	09	Start subroutine
[(]	53	10	
[x^2]	34	11	Calculate the square of radius
[X]	65	12	
[1]	01	13	
[•]	93	14	Value of h
[5]	05	15	
[X]	65	16	
[2nd] [π]	49	17	
[)]	54	18	Compute $V = \pi x^2 h$
[INV] [2nd] [SBR]	− 28	19	Return to main program
[LRN]	48.1		

Now let's compute the volume of a tube for which R = 0.6m and r = 0.5

Press	Display	Comments
[RST]	0	Position pointer for execution
[•] [5]	.5	Entry of r
[R/S]	1.1780972	Volume of internal cylinder
[•] [6]	.6	Entry of R
[R/S]	0.5183628	Volume of the tube

Section 7 — Building loops.

Iterative computations are frequently used in mathematics applications. For example if you have £275 in a saving account with an annual interest rate of 8.2%, what would be the total in your account after 5 years? After 10 years?
At the end of a year, the interest accumulated are 8.5% of the starting amount, and the new total is increased by the amount of these interests payments.
At the end of the first year, the value of the interests is:

$$275 \times \frac{8.2}{100} = £22.55$$

The total value of your saving is now:

$$275 + 22.55 = £297.55$$

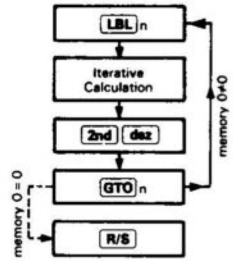
And so on, until you reach the end of the 5 years or 10 years.
Your TI 57-II allows you to repeat the same calculation a given number of times by using a special function: [2nd] [Dsz]

[2nd] [Dsz] — Decrement and skip on zero-key sequence:
This key sequence allows you to set up a repetitive calculation (or "loop") for as many times as you select. You just have to store the number of repetitions you want into memory zero, before starting the iterative calculation.
When the program pointer comes to a [2nd] [Dsz] key sequence, here is what happens:

- First, the contents of memory zero is decreased by one.
- The calculator then asks: is the contents of memory zero equal to zero?
- If the answer is NO, the program pointer proceeds right on to the step following [2nd] [Dsz].
- If the answer is YES, the program pointer skips the following instruction and continues the execution.

The instruction following the [2nd] [Dsz] instruction can be any instruction, but most often it will be a [GTO] n instruction allowing repetitive calculations as shown in the following chart.

If the contents of the register 0 is initially equal to p, the iterative calculation will be done p times, thus looping back to label n. Then, when the contents of memory zero will be equal to 0 it will skip the "branching" instruction and continue the program execution (or stop as shown on the chart).



Example:

Press	Display	St	Comments
[2nd] [Part] [2]	40.2		
[LRN] [2nd] [CP]	--	St	
[STO] [1]	61.01	00	Save initial capital
[R/S]	13	01	Wait for number of years input
[STO] [0]	61.00	02	Save number of iterations
[LBL] [0]	23.00	03	Loop starting adress
[RCL] [1]	71.01	04	
[X]	65	05	
[•]	93	06	
[0]	00	07	
[8]	08	08	
[2]	02	09	
[=]	95	10	Calculate the period interests
[STO] [+] [1]	61.85.01	11	Add them to capital
[2nd] [Dsz]	29	12	Decrement register 0
[GTO] [0]	22.00	13	- Goto label 0 if non completed
[RCL] [1]	71.01	14	- Recall the final capital if completed
[R/S]	13	15	Stop execution
[RST]	21	16	Prepare for next program execution
[LRN] [ON/C]	0		Exit learn mode.
[RST]	0		Prepare for program execution

Program execution: you just have to type in the starting amount, press [R/S], type in the number of years, then [R/S] again.

Press	Display	Comments
275 [R/S]	275	Initial capital
[5]	5	Number of years
[R/S]	407.82045	Final amount

Caution: [2nd] [Dsz] decreases the contents of memory zero by one if its contents is positive. If it is negative, il will be incremented by one. If the contents of memory zero is not an integer value, the calculator will react as if the next integer number is in memory zero.
Following table shows the number of loops done by the calculator depending on the initial value stored in memory zero.

Initial Value	Number of loops
5	5
5.2	6
5.9'	6
−5	5
−5.2	6
−5.9	6

Since memory zero is involved whenever [2nd] [Dsz] is used, it cannot be used for other storage purposes inside the loop, but however the number in memory zero can be recalled as part of the computational sequence.

NOTE: If the initial contents of memory zero is equal to zero, then the instruction following the [2nd] [Dsz] instruction is skipped but the memory 0 is not affected.

[INV] [2nd] [Dsz] – **Decrement and skip if not zero** - key sequence:
This key sequence works in the same fashion as [2nd] [Dsz] except that the step following this instruction is skipped if the contents of memory 0 is different from zero.
So when the program pointer comes to an [INV] [2nd] [Dsz] here is what happens:

- One is substracted from the contents of memory zero (or added if the number in memory zero is negative)
- Then the calculator asks: is the contents of memory zero equal to zero?
- If the answer is NO, the program pointer skips the instruction immediately following the [INV] [2nd] [Dsz] sequence.
- If the answer is YES, the program pointer proceeds right on the instruction immediately following [INV] [2nd] [Dsz].

Other characteristics of this function are identical to those of the [2nd] [Dsz] instruction.

Section 8 — Conclusion

STEPS IN WRITING A PROGRAM:

Writing a program requires a structured approach which can be summarized as follows:

1. Study the problem, gather the equations solving it.
2. Determine the goals of your program: during that phase you need to define how your program will interact with you. For example, which keys you will have to press, which parameters you will have to enter

and what results will be displayed.

3. Draw the program flow, indicating all the phases of the execution. This can be done with a flowchart (or schematic diagram). During this step, you need also to determine what will be stored in the data memories of your TI 57-II (parameters, data).
4. Write down the actual program steps, keeping track of the use of the memories and of the branching labels.
5. Enter the program into your calculator.
6. Test your program with test data for which you already know what the results should be, this is in order to be sure that your program is working correctly.
7. Edit and correct your program using the "debugging" and editing functions provided by your TI 57-II.
8. Once your program is working, document it carefully in order to keep it for future use. This documentation should include:
   - The name of the program
   - The goal of the program
   - The program flow
   - The memory map showing the use of data memories
   - The complete list of the program steps including the memory partitioning required
   - The test data in order to verify the program
   - The instructions to use the program

DEBUGGING A PROGRAM:

If the program does not give you the expected results, here are some "tips" on how you can solve the problem:

- Check that your program has been correctly keyed in.
- Execute the program step by step, with the [SST] key and compare the result of each step with what you expected. This is a good way to determine the "trouble maker" step.
- Check that you use numeric labels no more than once each, and that each label you are referring to has been defined.
- Check that subroutines are ended with an [INV] [2nd] [SBR] instruction.
- Verify that you deal with the correct user's data memories. Do not forget that [2nd] [Dsz] affects the contents of the memory "zero". Check that all the user memories, you want to use, exist in the current partitioning.
- Verify that you do not use more than four levels of pending operations, and that you comply to the rules of the Algebraic Operating System (AOS™).
- Verify that your calculator is set to the correct angular mode for trigonometric computations.
- If an "ERROR" message is displayed, refer to the Appendix A to help you locate the problem.

Chapter III — Applications

Section 1 — Statistical Analysis

- This program computes the mean and the standard deviation of a set of data.

- **Equations:**

  Mean: $\bar{x} = \Sigma \frac{x_i}{N}$

  Standard deviation: $\sigma n = \sqrt{\Sigma \frac{(x_i - \bar{x})^2}{N}}$

- **Memory map:**

Mo	M1	M2	Program steps	
Σx	Σx²	N		
			31	00

- **Running this program:**
  - Press 2nd CM then RST
  - To enter data point: enter x_i
    press R/S
  - To compute statistics results: press GTO 0, R/S → $\bar{x}$
    R/S → σn

- **Example:**

Press	Display	Comments
ON/C	0	
2nd CM	0	
RST	0	
1.75 R/S	1	Display the number
1.62 R/S	2	of observations entered
1.84 R/S	3	
1.72 R/S	4	
1.68 R/S	5	
GTO 0 R/S	1.722	$\bar{x}$
R/S	0.0733212	σn

- **Program listing:**

Keys	Code	Step #	Comments
2nd Part 3			
LRN 2nd CP	--	St	
STO + 0	61.85.00	00	Set partitioning to 3 data memories
x²	34	01	Enter data
STO + 1	61.85.01	02	
1	01	03	
STO + 2	61.85.02	04	
RCL 2	71.02	05	
R/S	13	06	
RST	21	07	
LBL 0	23.00	08	Mean computation
RCL 0	71.00	09	
÷	55	10	
RCL 2	71.02	11	
=	95	12	
R/S	13	13	
RCL 1	71.01	14	Standard deviation computation
÷	55	15	
RCL 2	71.02	16	
−	75	17	
(	53	18	
RCL 0	71.00	19	
÷	55	20	
RCL 2	71.02	21	
)	54	22	
x²	34	23	
=	95	24	
√x	35	25	
R/S	13	26	
LRN			

Section 2 — Combinations and permutations

- This program computes the possible combinaisons (C_n^p) or permutations (A_n^p) of n objects taken p at a time.

- **Equations:**

  $C_n^p = \frac{n!}{p!\,(n-p)!}$

  $A_n^p = \frac{n!}{(n-p)!}$

• **Memory map:**

M0	M1		
n	p	Program steps	
		39	00

• **Running this program:**

To compute C_n^p: enter n
press [GTO] 0 [R/S]
enter p
press [R/S]

To compute A_n^p: enter n
press [GTO] 1 [R/S]
enter p
press [R/S]

• **Example**

	Press	Display	Comments
C_{12}^3	12 [GTO] 0 [R/S]	12	
	3 [R/S]	220	$C_{12}^3 = 220$
A_8^5	8 [GTO] 1 [R/S]	8	
	5 [R/S]	6720	$A_8^5 = 6720$

• **Program listing:**

Press	Code	Step #	Comments
[2nd] [Part] [2]			Set partitioning to two user memories
[LRN] [2nd] [CP]	– –	St	
[LBL] [0]	23.00	00	C_n^p Computation
[2nd] [SBR] [2]	28.02	01	
[RCL] [0]	71.00	02	
[2nd] [x!]	40	03	
[+]	55	04	
[RCL] [1]	71.01	05	
[2nd] [x!]	40	06	
[+]	55	07	
[2nd] [SBR] [3]	28.03	08	
[=]	95	09	
[R/S]	13	10	
[LBL] [1]	23.01	11	A_n^p Computation
[2nd] [SBR] [2]	28.02	12	

Press	Code	Step #	Comments
[RCL] [0]	71.00	13	
[2nd] [x!]	40	14	
[+]	55	15	
[2nd] [SBR] [3]	28.03	16	
[=]	95	17	
[R/S]	13	18	
[LBL] [2]	23.02	19	Subroutine for n and p entry
[STO] [0]	61.00	20	
[R/S]	13	21	
[STO] [1]	61.01	22	
[INV] [2nd] [SBR]	–28	23	
[LBL] [3]	23.03	24	Subroutine computing (n – p)!
[(]	53	25	
[RCL] [0]	71.00	26	
[–]	75	27	
[RCL] [1]	71.01	28	
[)]	54	29	
[2nd] [x!]	40	30	
[INV] [2nd] [SBR]	–28	31	
[LRN]			

Section 3 — Hyperbolic functions

• This program computes the hyperbolic sine, cosine and tangent.

• **Equations:** $\sinh x = \dfrac{e^x - e^{-x}}{2}$

$$\cosh x = \frac{e^x + e^{-x}}{2}$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

• **Memory map:**

M0	M1		
x	cosh x	Program steps	
		39	00

- **Running this program:**

To compute cosh x: enter x
press GTO 0 R/S → cosh x
sinh x: enter x •
press GTO 1 R/S → sinh x
tanh x: enter x
press GTO 2 R/S → tanh x

- **Example:**

		Press	Display
cosh (3.2)	:	3.2 GTO 0 R/S	12.286646
sinh (8.1)	:	8.1 GTO 1 R/S	1647.2339
tanh (0.5)	:	.5 GTO 2 R/S	0.4621172

- **Program listing:**

Press	Code	Step #	Comments
2nd Part 2			
LRN 2nd CP	--	St	
LBL 0	23.00	00	Compute cosh x
STO 0	61.00	01	
INV lnx	−32	02	
+	85	03	
RCL 0	71.00	04	
+/−	94	05	
INV lnx	−32	06	
=	95	07	
÷	55	08	
2	02	09	
=	95	10	
INV 2nd SBR	−28	11	
LBL 1	23.01	12	Compute sinh x
STO 0	61.00	13	
INV lnx	−32	14	
−	75	15	
RCL 0	71.00	16	
+/−	94	17	
INV lnx	−32	18	
=	95	19	
÷	55	20	
2	02	21	
=	95	22	
INV 2nd SBR	−28	23	

LBL 2	23.02	24	Compute tanh x
2nd SBR 0	28.00	25	
STO 1	61.01	26	
RCL 0	71.00	27	
2nd SBR 1	28.01	28	
÷	55	29	
RCL 1	71.01	30	
=	95	31	
R/S	13	32	
LRN			

Section 4 — Inverse hyperbolic functions

- This program computes the following functions: $\sinh^{-1} x$, $\cosh^{-1} x$, $\tanh^{-1} x$.

- Equations: $\sin h^{-1} x = \mathrm{Ln}\,(x + \sqrt{x^2 + 1})$
$\cos h^{-1} x = \mathrm{Ln}\,(x + \sqrt{x^2 - 1}), \quad x \geq 1$
$\tan h^{-1} x = \frac{1}{2}\,\mathrm{Ln}\,(\frac{1 + x}{1 - x}), \quad -1<x<1$

- **Memory map:**

Mo		
x	Program steps	
	47	00

- **Running this program:**

Enter x then press GTO 0 R/S to compute $\cosh^{-1} x$
or press GTO 1 R/S to compute $\sinh^{-1} x$
or press GTO 2 R/S to compute $\tanh^{-1} x$

- **Examples:**

		Press	Display
$\cosh^{-1}(6.2)$	:	6.2 GTO 0 R/S	2.5111285
$\sinh^{-1}(3.7)$	:	3.7 GTO 1 R/S	2.0192607
$\tanh^{-1}(0.5)$	:	.5 GTO 2 R/S	0.5493061

- **Program listing:**

Press	Code	Step #	Comments
2nd Part 1	48.1		
LRN 2nd CP	--	St	
LBL 1	23.01	00	Compute $\sinh^{-1} x$
STO 0	61.00	01	
x^2	34	02	
+	85	03	
1	01	04	
=	95	05	
$\sqrt{x}$	35	06	
+	85	07	
RCL 0	71.00	08	
=	95	09	
lnx	32	10	
R/S	13	11	
LBL 0	23.00	12	Compute $\cosh^{-1} x$
STO 0	61.00	13	
x^2	34	14	
−	75	15	
1	01	16	
=	95	17	
$\sqrt{x}$	35	18	
+	85	19	
RCL 0	71.00	20	
=	95	21	
lnx	32	22	
R/S	13	23	
LBL 2	23.02	24	Compute $\tanh^{-1} x$
STO 0	61.00	25	
+	85	26	
1	01	27	
=	95	28	
÷	55	29	
(	53	30	
1	01	31	
−	75	32	
RCL 0	71.00	33	
)	54	34	
=	95	35	
lnx	32	36	
÷	55	37	
2	02	38	

Press	Code	Step #
=	95	39
R/S	13	40
LRN	48.1	

Section 5 — Quadratic equations

- This program solves the equation: $ax^2 + bx + c = 0$ $(a \neq 0)$
- **Equations:**

$\Delta = b^2 - 4ac$

1. $\Delta > 0$ $\quad x1 = \frac{-b - \sqrt{\Delta}}{2a}$

   $x2 = \frac{-b + \sqrt{\Delta}}{2a}$

2. $\Delta = 0$ $\quad x1 = x2 = \frac{-b}{2a}$

3. $\Delta < 0$ $\quad$ no root belonging to $\mathbb{R}$, 2 complex roots

   $z1 = \alpha + j\,\beta$

   $z2 = \alpha - j\,\beta$

   with $\alpha = \frac{-b}{2a}$ and $\beta = \frac{-\sqrt{-\Delta}}{2a}$

- **Memory map**

M_0	M_1	
a	b	Program steps
		39 … 00

- **Program listing**

Press	Code	Step #	Comments
2nd Part 2			
LRN 2nd CP	--	St	
2nd C.t	56	00	
STO 1	61.01	01	Save b value
x^2	34	02	
−	75	03	
R/S	13	04	Get c value
×	65	05	
4	04	06	

(Continued)

(Continuation)

[X]	65	07	
[RCL] [0]	71.00	08	
[=]	95	09	
[R/S]	13	10	Display Δ
[INV] [2nd] [x≥t]	−27	11	Test sign of Δ
[GTO] [1]	22.01	12	
[√x]	35	13	. Δ ⩾0
[+]	85	14	
[RCL] [1]	71.01	15	
[=]	95	16	
[2nd] [SBR] [0]	28.00	17	
[R/S]	13	18	Display x1
[+/−]	94	19	
[−]	75	20	
[RCL] [1]	71.01	21	Note: $x2 + x1 = -\frac{b}{a}$
[÷]	55	22	
[RCL] [0]	71.00	23	
[=]	95	24	
[R/S]	13	25	Display x2
[LBL] [1]	23.01	26	Compute complex roots
[+/−]	94	27	
[√x]	35	28	. Δ<0
[2nd] [SBR] [0]	28.00	29	
[R/S]	13	30	Display β
[RCL] [1]	71.01	31	
[LBL] [0]	23.00	32	
[÷]	55	33	
[2]	02	34	
[÷]	55	35	
[RCL] [0]	71.00	36	
[=]	95	37	
[+/−]	94	38	
[INV] [2nd] [SBR]	−28	39	Display α
[LRN]			

- **Running this program:**

Enter a, then press [STO] 0
Enter b, then press [RST] [R/S]
Enter c, then press [R/S] . → Δ
1 If Δ ⩾0 , press [R/S] → x1
press [R/S] → x2
2 If Δ<0 , press [R/S] → β
press [R/S] → α

- **Examples:**

$x^2 - 2x + 5 = 0$

Press	Display	Comments
[1] [STO] [0]	1	
[2] [+/−] [RST] [R/S]	4	
[5] [R/S]	−16	Δ is <0, z1 and z2 are complex numbers
[R/S]	−2	β } z1 = 1 − 2j
[R/S]	1	α } z2 = 1 + 2j

$x^2 + x - 6 = 0$

Press	Display	Comments
[1] [STO] [0]	1	
[1] [RST] [R/S]	1	
[6] [+/−] [R/S]	25	Δ is >0, x1 and x2 are real numbers
[R/S]	−3	x1 = −3
[R/S]	2	x2 = 2

Section 6 — Function plotter

- This program computes the values of a function in a given interval. The function to plot has to be entered as a program under label 0 and has to be followed by [INV] [2nd] [SBR]

- **Equations:**

F(x) is computed for x varying from a to b with step c.

- **Memory map:**

Mo	M1		
a	c	F(x) definition	program
		39 19	18 00

- **Running this program:**

Once F(x) has been entered in the program memory:

- enter a, press [RST], [R/S]
- enter b, press [R/S]
- enter c, press [R/S]

- **Example:**
- Set calculator to degrees mode with the DRG key.
- Calculate $f(x) = e^{\sin x}$ for x varying from 0 to 180°, with a 10° step.

Press	Display	Comments
0 RST R/S	0	a
180 R/S	180	b
10 R/S	0	x
	1	f(x)
	10	x
	1.189637	f(x)
	20	x
	1.4077887	f(x)

- **Program listing:**

Press	Code	Step #	Comments
2nd Part 2			
LRN 2nd CP	– –	St	
STO 0	61.00	00	
R/S	13	01	
x⇄t	51	02	
ON/C	15	03	
R/S	13	04	
STO 1	61.01	05	
RCL 0	71.00	06	
LBL 1	23.01	07	
2nd Pause	96	08	
SBR 0	28.00	09	Call subroutine computing f(x)
2nd Pause	96	10	
RCL 1	71.01	11	
STO + 0	61.85.00	12	
RCL 0	71.00	13	
INV 2nd x≥t	– 27	14	
GTO 1	22.01	15	
2nd x=t	26	16	
GTO 1	22.01	17	
R/S	13	18	
LBL 0	23.00	19	Function definition
sin	42	20	(can use up to 21 steps)
INV lnx	– 32	21	In this example: $F(x) = e^{\sin x}$
INV 2nd SBR	– 28	22	
LRN			

Section 7 — Limit of a function

- This program is designed to evaluate the behaviour of a function (f(x)) as its variable approaches some specific value. In some cases the function may "diverge" or "go to infinity" right at the limit point or it may converge to a specific limit value.

- **Equations:**

Compute $\lim_{x \to x0} f(x)$.

F(x) is evaluated for values of x that move from x1 to x0, such as

$$xn = \frac{xn-1 + x0}{2}$$

, thus halving the distance from x to x0 at each iteration.

- **Memory map:**

M_0	M_1		
x	x_0	F(x) Equation	Program
		39 15	14 00

- **Running this program:**

Once f(x) has been entered in the program memory
enter x0, press RST, R/S
enter x1, press R/S
This program will run indefinitely. Hold R/S down to stop its execution.

- **Example:**

Evaluate the limit of $f(x) = \frac{\text{Lnx}}{x - 1}$ for x approaching 1.

Press	Display	Comments
1 RST R/S	1	x0
2 R/S	2	x1
	0.6931472	f(x1)
	1.5	x2
	0.8109302	f(x2)
	1.25	x3
	0.8925742	f(x3)
		etc.

As you watch the display you will notice that when x approaches 1, the limit of f(x) approaches 1.

- **Program listing:**

Press	Code	Step #	Comments
2nd Part 2			
LRN 2nd CP	--	st	
STO 1	61.01	00	Limit point
R/S	13	01	Get starting point (x_1)
LBL 1	23.01	02	
STO 0	61.00	03	
2nd SBR 0	28.00	04	Call subroutine computing f(x)
2nd Pause	96	05	Display f(x)
RCL 0	71.00	06	
+	85	07	
RCL 1	71.01	08	
=	95	09	
÷	55	10	
2	02	11	
=	95	12	Calculate next value of x (x_n)
2nd Pause	96	13	Display x
GTO 1	22.01	14	Go back to next iteration
LBL 0	23.00	15	Subroutine computing f(x)
ln x	32	16	
÷	55	17	(can use up to step # 39)
(	53	18	
RCL 0	71.00	19	in this example, $f(x) = \frac{Lnx}{x - 1}$
−	75	20	
1	01	21	
)	54	22	
=	95	23	
INV 2nd SBR	−28	24	
LRN			

Section 8 — HI-LO Game

How many guesses will it take to you to discover a "secret" number between 0 and 1000? After each guess, you are told whether your guess was "high" or "low".

- **Program listing:**

Press	Display	Step #	Comments
2nd Part 2			
LRN 2nd CP	– –	St	
2nd CM	76	00	
+	85	01	
2nd π	49	02	
=	95	03	
y^x	45	04	
8	08	05	
=	95	06	
2nd Frac	59	07	Generate a pseudo random
X	65	08	number y such as
3	03	09	$= \text{INT}\ (10^3 \times \text{FRAC}\ [(\pi + x)^8])$
INV log	−31	10	x being the number introduced.
=	95	11	
2nd Intg	58	12	
x⇄t	51	13	
ON/C	15	14	
LBL 0	23.00	15	
=	95	16	
R/S	13	17	Enter guess
STO 0	61.00	18	
1	01	19	
STO + 1	61.85.01	20	Increment number of tries
RCL 0	71.00	21	
2nd x=t	26	22	Is the guess correct?
GTO 3	22.03	23	
INV 2nd x≥t	−27	24	
GTO 2	22.02	25	. no - too high
1	01	26	
GTO 0	22.00	27	
LBL 2	23.02	28	. no - too low
1	01	29	
+/−	94	30	
GTO 0	22.00	31	
LBL 3	23.03	32	. yes - it is correct
RCL 0	71.00	33	Recall the correct guess
2nd Pause	96	34	
RCL 1	71.01	35	Recall the number of tries
2nd Pause	96	36	
GTO 3	22.03	37	
LRN			

- **To play the game:**
  - Press [•], then enter the time of the day and the number of the day (or any positive number less than 1).
  - Press [RST], [R/S] the calculator displays a "0", telling you it's ready to play.
  - Now enter your guess and press [R/S]
    - If your guess is "high" a "1" is displayed
    - If it is low, a "−1" is displayed.
    - If your guess is right, the calculator displays alternatively the correct number and the number of tries it took you to get it.

- To guess again, enter your new number and press [R/S]
- After you guessed correctly, you can restart a new game by pressing [R/S], [RST] and re-entering a new seed.
- **Example:**

Press	Display	Comments
.222312 [RST]	0.222312	
[R/S]	0	
500 [R/S]	1	
300 [R/S]	−1	
400 [R/S]	−1	
450 [R/S]	1	
425 [R/S]	−1	
434 [R/S]	434	
	6	

Appendix A — Error conditions

The following list the circumstances which cause "error" to be displayed. When this occurs, no entry from the keyboard except [OFF] is accepted until [ON/C] is pressed. Presing [ON/C] clears the error condition and all pending operations. When the error occurs in a program, the program pointer points to the instruction following the one which caused the error, except for branching errors which bring the program pointer back to the program starting location. The analysis of the causes of the error will allow you take appropriate action to correct it.

1. Number entry or calculation result (including in memories) is outside the range $\pm 1 \times 10^{-99}$ to $\pm 9.9999999 \times 10^{99}$.
2. Multiplying a number greater than 1×10^{99} by another number may cause an error condition.

3. Dividing a number by zero.
4. Calculating [log], [lnx], or [1/x] or zero, calculating the 0th root of any number or zero to the zero power.
5. Calculating [log], [lnx], a power or root of a negative number.
6. Inverse of sine or cosine (arcsine or arccosine) when the absolute value in the display is greater than 1.
7. Tangent of 90° or 270°, $\pi \div 2$ radians or $3\pi \div 2$ radians, 100 grads or 300 grads, or their rotational multiples such as 450°.
8. Having more than 15 open levels of parentheses or more than 4 pending operations.
9. Factorial of any number except a non-negative integer less than 70.
10. Trying to set the memory partitioning to 0, 8 or 9 data memories.
11. Using [RCL], [STO], or [EXC] for a memory that is not defined by the current partition.
12. When using memory arithmetic, following [STO] with two memory arithmetic operations and a valid user data memory number.
13. When using memory arithmetic, following [RCL] or [EXC] with an operation instead of a valid user data memory number.
14. Pressing [LRN], [SST], [R/S] when no program steps are available. (calculator set to partition with 7 data memories).
15 Using [GTO] n or [2nd] [SBR] n with an unexisting label n definition.
16. Using an argument outside the range given in Accuracy Information (see Appendix B) for the logarithmic and trigonometric functions.

Appendix B: Accuracy Information

Accuracy Information

Each calculation produces an 11-digit result which is rounded to an 8-digit standard display. The 5/4 rounding technique used adds 1 to the least significant digit in the display if the next nondisplayed digit is five or more. If this digit is less than five, no rounding occurs. In the absence of these extra digit, inaccurate results would frequently be displayed, such as $1 \div 3 \times 3 = 0.9999999$

Because of rounding, the answer is given as 1, but is internally equal to 0.9999999999.

The higher order mathematical functions use iterative calculations. The cumulative error from these calculations in most cases is maintained beyond the 8-digit display so that no inaccuracy is displayed. Most calculations are accurate to ± 1 in the last displayed digit. There are a few instances in the solution of higher order functions where display accuracy begins to deteriorate as the function approaches a discontinuous or undefined point. For example, the tangent of 87° is accurate for all displayed digits. However, the tangent of 89.99999° is accurate to only three places. Another example is when the y^x function has a y value that approaches 1 and an x value that is a very large positive or negative number. The displayed result for 1.05^{-160} is accurate for all displayed digits, while 1.0000005^{-16000} is accurate to only five places.

Trigonometric values can be calculated for angles greater than one revolution. As long as the trigonometric function result is displayed in normal form rather than in scientific notation, all displayed digits are accurate for any angle from −36,000° to 36,000° and −40,000 to 40,000 grads. The equivalent range in radians (± 200 π) is comparable to degrees and grads in accuracy except at rotation multiples of π and π÷ 2. The rounded value of π limits accuracy at these points. In general, the accuracy decreases one digit for each decade outside this range.

B-2 — Accuracy Information

The following gives the limits within which the display must be when calculating certain functions.

Function	Limit
$\sin^{-1} x$, $\cos^{-1} x$	$-1 \le \lvert x\rvert \le 1$
	x=0
ln x, log x	$1 \times 10^{-99} \le x < 1 \times 10^{100}$
e^x	$-227.95592 \le x \le 230.25850$
10^x	$-99 \le x < 100$
x!	$0 \le x \le 69$ where x is an integer

The following gives the range of results of the inverse trigonometric functions.

Arc Function	Range of Resultant Angle
arcsin x	0 to 90°, π÷2 radians, or 100G
arcsin −x	0 to −90°, −π÷2 radians, or −100G
arccos x	0 to 90°, π÷2 radians, or 100G
arccos −x	90° to 180°, π÷2 to π radians, or 100G to 200G
arctan x	0 to 90°, π÷2 radians, or 100G
arctan −x	0 to −90°, −π÷2 radians, or −100G

Appendix C — Keycodes in numeric order

Code	Key	Code	Key
00	0	53	(
01	1	54	)
02	2	55	÷
03	3	56	2nd C.t
04	4	57.0n	2nd Fix n
05	5	58	2nd Intg
06	6	59	2nd Frac
07	7	61.0n	STO n
08	8	61.45.0n	STO y^x n
09	9	61.55.0n	STO ÷ n
13	R/S	61.65.0n	STO X n
15	ON/C	61.75.0n	STO − n
21	RST	61.85.0n	STO + n
22.0n	GTO n	65	X
23.0n	LBL n	71.0n	RCL n
26	2nd x=t	75	−
27	2nd x≥t	76	2nd CM
28.0n	2nd SBR n	81.0n	EXC n
29	2nd Dsz	85	+
31	log	93	•
32	lnx	94	+/−
33	1/x	95	=
34	x^2	96	2nd Pause
35	$\sqrt{x}$		
40	2nd x!		
41	DRG		
42	sin		
43	cos		
44	tan		
45	y^x		
46	2nd DRG▸		
47	2nd P⇄R		
48	2nd DMS DD		
49	2nd π		
50	2nd |x|		
51	x⇄t		
52	EE		

Appendix D: Summary of clearing keys

Key pressed	Display digits	Pending operations	Fixed decimal	User data memories	Program	t-Register
[ON/C]	Cleared	None	None	None	None	None
[ON/C] [ON/C]	Cleared	Cleared	None	None	None	None
[OFF] [ON/C]	Cleared	Cleared	Cleared	None	None	None
[2nd] [CM]	None	None	None	Cleared	None	None
[2nd] [CP]	None	None	None	None	Cleared	None
[2nd] [C.t]	None	None	None	None	None	Cleared

In case of Difficulty

In the event that you have difficulty with your calculator, the following instructions will help you to analyse the problem.

You may be able to correct the calculator problem yourself without returning the unit to a service facility.

A. Self service

1. Problem: Calculator gives erroneous answers or displays "Error".

Remedy: Check the paragraphs on error conditions, accuracy information or the paragraphs of the manual describing the functions used during your calculations. Re-initialize the calculator using the following key sequence:

[ON/C] [ON/C] [2nd] [Part] 7 [2nd] [CM] [2nd] [C.t] [2nd] [Part] 1 [ON/C]

2. Problem: The display is blank, shows erratic numbers, or grows dim.

Remedy: The batteries may be discharged. Insert new batteries, following the instructions in the paragraph "Battery Replacement".

4. Problem: The "RUN" indicator is on and the display shows a steady zero or flashing numbers.

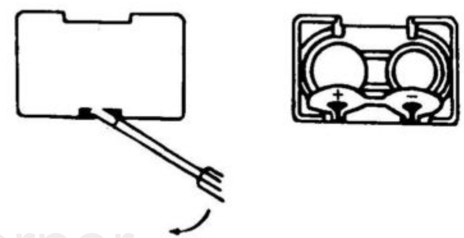
Remedy: You calculator is running a program. Wait for a few moments until calculation is completed or press [R/S] or press [OFF] [ON/C] and eventually reinitialize your calculator.

B. Assistance:

If the above remedies do not correct the difficulty, please contact your retailer.

Battery Replacement

NOTE: The calculator cannot hold data in its users data memories or mode registers when the batteries are removed or become discharged.
The calculator uses 2 of any of the following batteries for up to 750 hours of operation: Panasonic LR-44, Ray-O-Vac RW-82, Union Carbide (Eveready) A-76 or the equivalent. For up to 2000 hours of operation use Mallory 10L-14 or D357, Union Carbide (Eveready) 357, Panasonic WL-14, Toshiba G-13, Ray-O-Vac RW-42 or the equivalent.



1. Turn the calculator off. Place a small screwdriver, paperclip, or other similar implement into the slot and gently lift the battery cover.
2. Remove the discharged batteries and install new ones as shown. Be careful not to crease the film contacts while installing the new batteries. Be sure the film contacts are positioned to lie on top of the batteries after they are installed.
3. Replace the cover, top edge first, then gently press until the bottom edge snaps into place.
4. Press [ON/C] [ON/C] [2nd] [Part] 7 [2nd] [CM] [2nd] [C.t] [2nd] [Part] 1 [ON/C]. The display then shows 0 and is ready to be used.

CAUTION: Do not incinerate the old batteries.

TWO-YEAR WARRANTY

In case of breakdown or damage, please consult your local Texas Instruments retailer.

1. The terms and conditions set out hereinunder shall not apply where you have purchased this calculator directly from Texas Instruments Ltd. in which case the conditions of sale of Texas Instruments Ltd. shall apply.
2. This electronic calculator (including charger if applicable) from Texas Instruments is warranted to the original purchaser for a period of two (2) years from the original purchase date -under normal use and service -against defective materials or workmanship. For those calculators designed to incorporate batteries, this warranty does not cover damage resulting from any battery leakage. Batteries delivered with calculators are for demonstration purposes only.

   This warranty is void if : the calculator has been damaged by accident or unreasonable use, neglect, improper service or other causes not arising out of defects in material or workmanship.

   During the above two-year period, the calculator or its defective parts will be repaired, adjusted and/or replaced with a reconditioned model of equivalent quality, ("RECONDITIONED") at manufacturer's option without charge to the purchaser when the calculator is returned, by way of the dealer to Texas Instruments with proof-of-purchase date. UNITS RETURNED WITHOUT PROOF-OF-PURCHASE DATE WILL BE REPAIRED AT THE SERVICE RATES IN EFFECT AT THE TIME OF RETURN.
   In the event of replacement with a reconditioned model, the replacement unit will continue the warranty of the original calculator product or 90 days, whichever is longer.

   THIS CONDITION 2 SHALL NOT AFFECT THE STATUTORY RIGHTS OF A CONSUMER AS DEFINED IN THE CONSUMER TRANSACTIONS (RESTRICTIONS ON STATEMENTS) ORDER 1976 (AS AMENDED).
3. Save as expressly provided in Condition 2, Texas Instruments shall be under no liability of whatsoever kind, howsoever caused whether or not due to the negligence or wilful default of Texas Instruments or its servants or agents arising out of or in connection with this calculator provided that nothing contained in this condition 3 shall exclude or restrict :
   (I) Any liability of Texas Instruments for death or personal injury resulting from the negligence of Texas Instruments or its servants or agents; or
   (II) Any liability of Texas Instruments for loss or damage arising from this calculator proving defective while in consumer use (within the meaning of Sec. 5 (2) (A) Unfair Contract Terms Act. 1977) and resulting from the negligence of Texas Instruments or its servants or agents.

3. Save as expressly provided in Condition 2, Texas Instruments shall be under no liability of whatsoever kind, howsoever caused whether or not due to the negligence or wilful default of Texas Instruments or its servants or agents arising out of or in connection with this calculator provided that nothing contained in this condition 3 shall exclude or restrict :
   (I) Any liability of Texas Instruments for death or personal injury resulting from the negligence of Texas Instruments or its servants or agents; or
   (II) Any liability of Texas Instruments for loss or damage arising from this calculator proving defective while in consumer use (within the meaning of Sec. 5 (2) (A) Unfair Contract Terms Act. 1977) and resulting from the negligence of Texas Instruments or its servants or agents.

SUGGESTIONS

Because of the number of suggestions which come to Texas Instruments from many sources, containing both new and old ideas, Texas Instruments will consider such suggestions only if they are freely given to Texas Instruments. It is the policy of Texas Instruments to refuse to receive any suggestions in confidence. Therefore, if you wish to share your suggestions with Texas Instruments, or if you wish us to review any calculator program key sequence which you have developed, please include the following in you letter:

"All of the information forwarded herewith is presented to Texas Instruments on a nonconfidential, nonobligatory basis; no relationship, confidential or otherwise, expressed or implied, is established with Texas Instruments by presentation. Texas Instruments may use, copyright, distribute, publish, reproduce, or dispose of the information in any way without compensation to me".

GB

TEXAS INSTRUMENTS

PRINTED IN ITALY
IMPRIMÉ EN ITALIE