Chapter 4—DETAILS of the Features and Functions

This chapter provides a closer inspection of your calculator and is specifically designed as a detailed reference to be used when you are thoroughly familiar with the calculator's functions. If you encounter a concept that you are unsure of, use the calculator to experiment with that feature and/or return to earlier sections of this sourcebook to further your knowledge of the calculator.

All discussions of keyboard operations apply to both manual calculations and program calculations.

Basic Operations

Standard Display In addition to power-on indication, the display provides numerical information complete with negative sign and decimal point and shows Error for an overflow, underflow or error condition. An entry can contain as many as 10 digits. All digits entered after the tenth are ignored.

The display includes three angle mode indicators, ten possible positions for digits, a floating decimal point, and a floating minus sign. The nine possible positions for the decimal can show a decimal point between any of the ten display positions. Negative numbers are displayed with a minus sign immediately to the left of the number.

See *Appendix C* for the accuracy of displayed results.

Data Entry Keys The keys have been positioned on the keyboard to provide efficient calculator operation. Although many of the operations are obvious, some are not. The following instructions and examples will help you develop skill and confidence in using your calculator.

[0] Through [9]. Digits: Enter the numbers 0 through 9.

[.]. Decimal Point: Enters the decimal point. The decimal point can be entered wherever needed. If no decimal point is entered, it is assumed to be to the right of the number and does not appear. A zero precedes the decimal point for numbers less than 1. Trailing zeros on the decimal portion of a number are not normally displayed. Only the first decimal point entered is accepted; all others are ignored. Pressing the decimal point during exponent entry returns you to mantissa entry, allowing you to enter more digits to the right of the decimal point or change the sign of the mantissa.

[2nd] [π]. Pi: Enters the value of π to 13 significant digits (3.141592653590) for calculations; the display shows the rounded value. [CE] does not remove π. However, you can enter a value over π.

[+/−]. Change Sign: Instructs the calculator to change the sign of the displayed number. When pressed after [EE] or exponent entry, changes the sign of the exponent.

The procedure for entering a positive number is simply to press the keys in the left to right sequence exactly as the number is written. Each digit entry causes the displayed numbers to shift left as the new digit is entered. Only the first decimal point entered in any single number entry is accepted.

Example: $7.892 - \pi + (-2) = 2.750407346$

Press	Display
7.892 [−]	7.892
[2nd] [π]	3.141592654
[+]	4.750407346
2 [+/−]	−2
[=]	2.750407346

Clearing Operations

[CE], Clear Entry: Clears entries made with the digit, decimal point and change-sign keys only when pressed immediately after these keys. This key does not clear calculated results, numbers recalled from memory, or π. [CE] also clears Error from the display but leaves the values in the display. Use of this key does not affect pending operations.

[CLR], General Clear: Clears calculations in progress and the display. It resets scientific notation to standard format and clears Error from the display. This key does not affect the contents of the user data or program memories, the t register, angle mode, engineering or fix-decimal display formats or the partition.

When the [=] key is pressed to complete a calculation, the answer is displayed and the calculator is ready to start a new problem without pressing any of the clear keys. The contents of the data memories are not affected by [=].

[2nd] [CP], Clear Program: Clears all locations of program memory, clears the subroutine return register, resets all flags, clears the t register, and resets the program pointer to ST when pressed from the keyboard. When encountered within a program it only clears the t register.

[2nd] [CMs], Clear User Data Memory: Instructs the calculator to clear all user data memories as defined by the current partition.

Alternate Function Keys ([2nd] and [INV])

Most of your calculator's keys have alternate functions. The first function is shown directly on the key and the second function is printed above it. To execute a function shown on a key, simply press the key. To use the second function of a key, press the [2nd] key, then press the key just below the second function. For example, to find the natural logarithm of a number, press [lnx]. To find the common logarithm of a number, press [2nd] [log]. Pressing [2nd] places the keyboard in a state in which the next key pressed (except inverse) causes that key's second function to be used and returns the keyboard to a first function state. The second function state can be cancelled by pressing [2nd] again.

The inverse key, [INV], like the [2nd] key, adds additional computing capabilities without increasing the number of keys on the keyboard. When [INV] precedes another key, another function of that key is accessed. The inverse key can be used with the following keys to obtain the indicated function. Pressing [CE], [CLR], [INV], or a key without an inverse function cancels the inverse state.

Function	Inverse Function
[EE]	removes EE
[2nd] [Eng]	removes ENG
[2nd] [Fix]	removes Fix
[2nd] [log]	10^x
[lnx]	e^x
[y^x]	$\sqrt[x]{y}$
[2nd] [Intg]	fractional part
[2nd] [sin]	$\sin^{-1}$
[2nd] [cos]	$\cos^{-1}$
[2nd] [tan]	$\tan^{-1}$
[2nd] [Prd]	divide into memory
[SUM]	subtract from memory
[2nd] [DMS-DD]	D.DD to D.MMSS
[2nd] [P►R]	R to P
[2nd] [Σ+]	Σ−
[2nd] [$\bar{x}$]	standard deviation
[2nd] [List]	list data memories
[2nd] [x=t]	x ≠ t
[2nd] [x≥t]	x < t
[2nd] [IfF]	if flag is reset
[2nd] [StF]	reset flag
[2nd] [Dsz]	skip on nonzero
[SBR]	return

An inverse instruction can be cancelled by pressing [INV] a second time if no other key has been pressed. For those keys that have no inverse, such as [×], [LRN], etc., a preceding inverse key is ignored. When used in conjunction with the second function key, the inverse key can be pressed before or after the second function key is pressed, i.e. [INV] [2nd] [log] is the same as [2nd] [INV] [log]. For examples using [INV] with a particular key, see the section relating to that key.

Display Formats Even though a maximum of 10 digits can be entered or displayed, the internal display register always retains results to 13 digits. The results are then rounded for display only. The extra digits, called guard digits, guard the accuracy of the displayed value and are not intended to be used for extended precision. If you use the guard digits, be aware that if any inaccuracy occurs, the calculator conceals it within these digits. See Appendix C for a detailed discussion of accuracy.

Scientific Notation

[EE] Enter Exponent—Allows you to enter a number in scientific notation in the display. After the [EE] key is pressed, all further results are displayed in scientific notation format until [CLR] is pressed or until the calculator is turned off. [INV] [EE] or [INV] [2nd] [Eng] also remove scientific notation, but only if the displayed number is in the range ± 0.001 to ± 9999999999. When [EE] is pressed after a result (intermediate or final), only the value in the display is used for further calculations. Any other digits are discarded.

Any number can be entered as the product of a value (mantissa) and 10 raised to some power (exponent). Enter the mantissa (up to 7 digits), press [EE], then enter the exponent (any 2 digits).

This capability allows you to work with numbers as small as $\pm 1 \times 10^{-99}$ or as large as $\pm 9.999999 \times 10^{99}$. Numbers smaller in magnitude than 0.000000001 or larger than 9999999999 must be entered in scientific notation. The entry procedure is to key in the mantissa up to 7 digits and its sign, then press [EE] and enter the exponent of 10 and its sign.

For example, the number 320,000,000,000 can be written as 3.2×10^{11} and can be entered into the calculator as:

Press	Display
[CLR]	0
3.2	3.2
[EE]	3.2 00
11	3.2 11

More than 2 digits can be entered after pressing [EE], but only the last two entered are retained as the exponent. This feature can be used to correct an erroneous exponent entry without having to clear the entry.

In scientific notation, a positive exponent indicates how many places the decimal point of the mantissa would be shifted to the right to place the number in standard form. If the exponent is negative, the decimal would be moved to the left.

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

Example: Enter 6025×10^{20}

Press	Display
[CLR]	0
6025	6025
[EE]	6025 00
20	6025 20
[+]	6.025 23

In scientific notation, the mantissa is limited to 7 digits to allow display space for the exponent. A mantissa resulting from a calculation is also displayed to 7 digits, but is internally carried to 13 digits. This 13-digit value is the one used for all subsequent calculations. See *Appendix C* for more on these extra digits.

Note: You cannot enter scientific notation format, even though [EE] is pressed, if there are more than 7 mantissa digits entered (this includes π). If [EE] is pressed with more than 7 digits in the display, the display goes into scientific notation format when an operation or function key is pressed.

The change-sign key can be used to attach a negative sign to the mantissa and to the power-of-ten exponent. Simply press [+/−] after entry of the mantissa to change the sign of the number or after the exponent to change the sign of the power of ten. To change the sign of the mantissa or to enter extra digits after the [EE] key has been pressed, press [.] and then enter the mantissa's sign change or additional numbers to the right of the decimal point.

Example: Enter -4.962×10^{-12}. Then change the mantissa to read -4.96236×10^{12}.

Press	Display	Comments
[CLR]	0	
4.962 [+/−]	−4.962	Enter mantissa and sign
[EE]	−4.962 00	
12 [+/−]	−4.962 −12	Enter exponent and sign
[+/−]	−4.962 12	Change exponent sign
[.] [+/−]	4.962 12	Change mantissa sign
36 [+/−]	−4.96236 12	Complete the mantissa

Data in scientific notation format can be mixed with data in standard format. The calculator converts the entered data for proper calculation. After the [EE] key is pressed, the calculator displays all the results in scientific notation format until [CLR], [INV] [EE] or [INV] [2nd] [Eng] is pressed. [CE] clears an entry in scientific notation, but the format remains.

Example: $1.816 \times 10^3 - 581.432191 = 1.234568 \times 10^3 = 1234.567809$

Press	Display
[CLR]	0
1.816 [EE]	1.816 00
3 [−]	1.816 03
581.432191 [=]	1.234568 03
[INV] [EE]	1234.567809

Pressing [INV] [EE] to remove scientific notation when the number is outside of the range ±9999999999 to ±0.001 does not return the calculator to standard format. However, [INV] [EE] does set the calculator to display numbers in standard format when a calculated result lies within this range.

Example: $(7 \times 10^{11} + 5 \times 10^{10}) \div 25 \div 25 = 1200000000$

Press	Display
[CLR]	0
7 [EE]	7 00
11 [+]	7. 11
5 [EE]	5 00
10 [=] [INV] [EE]	7.5 11
[÷]	7.5 11
25 [=] [÷]	3. 10
25 [=]	1200000000

If the magnitude of calculated results exceeds 9999999999 or goes below 0.001, the display automatically goes into scientific notation. When this occurs without [EE] having been pressed, the display automatically reverts back to standard display format whenever possible.

There are two approaches to convert a calculated result to scientific notation. The first is to press [×] 1 [EE] [=] which multiplies the number in the display register by 1×10^0 and converts the display to scientific notation. The complete 13-digit number is retained. The second method is to press [EE] [=]. You should be careful in using the second method. It has the effect of instructing the calculator to use the ROUNDED quantity being displayed for subsequent calculations, discarding all guard digits.

You should avoid using the display commands which use [=] in the middle of a computation. The reason is that the [=] key completes all pending calculations. To avoid this, either use these conversion methods only after computations are complete or use [×] 1 [EE], followed by another operation.

Engineering Notation
This modified form of scientific notation is accessed by pressing [2nd] [Eng]. The displayed value in this format consists of a mantissa and an exponent that is adjusted so that the exponent is a multiple of three (10^{12}, 10^{-6}, etc.) and the mantissa has 1, 2, or 3 digits to the left of the decimal point. This allows the calculator to display results in units that are readily usable such as 10^{-12} for picofarads, 10^{-3} for millimeters, 10^6 for megahertz or 10^{-9} for nanoseconds.

Example: What is the diameter of a fiber in micrometers (1 micrometer $= 10^{-6}$ meters) whose circumference is 3×10^{-3} meters?

$d = C/\pi$

Press	Display
[CLR] [2nd] [Eng]	0.00
3 [EE]	3 00
3 [+/−] [÷]	3 −03
[2nd] [π] [=]	954.9297 −06

Pressing **[INV] [2nd] [Eng]** removes this display format. Clearing operations or **[INV] [EE]** do not clear this format.

Fix-Decimal Control

In standard display format, scientific notation, and engineering notation, you can selectively choose the number of digits to display following the decimal point. Pressing **[2nd] [Fix]** and entering the desired number of decimal places (0 to 8) instructs the calculator to round all displayed results to the selected number of decimal places. This rounding only affects the display, not the display register. All subsequent calculations use the full unrounded value.

Pressing **[2nd] [Fix]** 9 or **[INV] [2nd] [Fix]** returns the calculator to the standard display format. Data entries can still be made with 10 digits (7 in scientific notation) with all subsequent calculations using the 13-digit unrounded results except the DMS-DD conversion which uses the displayed value only. Only the display is altered to the requested number of decimal places unless you press **[EE] [INV] [EE]** to discard the portion that is not displayed. If you press **[EE] [INV] [EE]**, the calculator will discard all digits which are not displayed.

Be sure you have removed the display from engineering notation format in the previous example.

Example: 2/3 = 0.666666667

Press	Display
2 [÷]	2
3 [=]	0.666666667
[2nd] [Fix] 5	0.66667
[2nd] [Fix] 2	0.67
[2nd] [Fix] 0	1
[INV] [2nd] [Fix]	0.666666667

Remember that only the displayed number is rounded according to the current format.

Example: $1 \times 10^{-3} \div 2 = 0.0005$

Press	Display
1 [EE]	1 00
3 [+/−]	1 −03
[÷] 2 [=]	5. −04
[2nd] [Fix] 2	5.00 −04
[INV] [EE]	0.00
[2nd] [Fix] 3	0.001
[2nd] [Fix] 4	0.0005
[2nd] [Fix] 5	0.00050
[INV] [2nd] [Fix]	5. −04

Note that the zero that occurs about the middle of the example is not really a zero in the display register. The display just rounds to zero in the fix-2 display format. Always be aware that the display register is not affected by the fix-decimal option.

Display Shows Error

Error is displayed when the limits of the calculator are exceeded or an improper mathematical operation is used. Press [CE] to clear the error. [CLR] will also clear an error condition. See Appendix B for a complete list of error and overflow/underflow conditions and the results they produce.

Arithmetic Calculations

This calculator's method of entering numbers and operations allows straightforward entry of most problems just as they are mathematically stated. The calculator remembers each operation and, if necessary, stores it until it can be applied according to the standard rules of algebra.

Basic Functions—[+] [−] [×] [÷] [=]

To perform simple addition, subtraction, multiplication, or division, this calculator with its AOS™ algebraic operating system allows you to key in the problem in an order close to the way it is written.

Example: $1.6 \times 10^{-19} \times 6.025 \times 10^{23} = 9.64 \times 10^{4}$

Press	Display
[CLR]	0
1.6 [EE]	1.6 00
19 [+/−] [×]	1.6 −19
6.025 [EE]	6.025 00
23 [=]	9.64 04

Notice that the [=] key completes the arithmetic operations and displays the final answer.

Pressing [CLR] at the beginning of a new sequence clears any calculations in progress and ensures that no pending operations remain from prior calculations. This is not required if the previous problem used [=] to obtain the result. Pressing [=] clears numbers and entries from the AOS without removing scientific notation or clearing Error from the display.

After a result is obtained in one calculation it may be directly used as the first number in a second calculation. There is no need to reenter the number from the keyboard.

Example: 1.84 + 0.39 = 2.23 then (1.84 + 0.39)/365 = .006109589

Press	Display	Comments
1.84 [+]	1.84	
.39 [=]	2.23	1.84 + 0.39
[÷]	2.23	
365 [=]	0.006109589	2.23 ÷ 365

Pressing any two of the operations keys ([+] [−] [×] [÷] [y^x] [=]) in succession causes operator replacement. An operation directly following another operation takes the place of the first operator. The sequence 5 [×] [+] 6 [=] gives 11; here, the plus replaced the times.

DETAILS

Algebraic Operating System Entry Method

Algebraic hierarchy is an essential feature of the Algebraic Operating System method of entering numbers. To efficiently combine operations, the standard rules of algebraic hierarchy have been programmed into the calculator.

These algebraic rules assign priorities to the various mathematical operations. Without a fixed list of priorities, expressions such as $5 \times 4 + 3 \times 2$ could have several meanings:

$5 \times (4 + 3) \times 2 = 70$
or $(5 \times 4) + (3 \times 2) = 26$
or $((5 \times 4) + 3) \times 2 = 46$
or $5 \times (4 + (3 \times 2)) = 50$

The rules of algebraic hierarchy state that multiplication is to be performed before addition. So algebraically, the correct answer is $(5 \times 4) + (3 \times 2) = 26$. The complete list of priorities for interpreting expressions is:

1. Immediate Math Functions
2. Exponentiation (y^x) and Roots ($\sqrt[x]{y}$)
3. Multiplication, Division
4. Addition, Subtraction
5. Equals

1. Math functions (trigonometric, logarithmic, e^x, 10^x, square, square root , integer, fraction, absolute value, reciprocal and conversions) immediately replace the displayed value with its functional value.
2. Exponentiation (y^x) and roots ($\sqrt[x]{y}$) are performed next.
3. Multiplication and division are performed after completing math functions, exponentiation, root extraction, and other multiplication and division.
4. Addition and subtraction are performed only after completing all operations through multiplication and division as well as other addition and subtraction.
5. Equals completes all uncompleted operations in the above order.

An operation completes other operations of the same (or higher) priority level. Some operations are performed immediately while others are held pending until the rules say to perform them. As an illustration, consider the following example.

Example: $4 \div 5^2 \times 7 + 3 \times 0.5^{\cos 60°} = 3.241320344$

Press	Display	Comments
[2nd] [Deg]		Set angle mode to degrees
4 [÷]	4	(4÷) is stored
5 [x^2]	25	(5^2) immediate function x^2 evaluated
[×]	0.16	($4 \div 5^2$) ÷ evaluated because × is same priority as ÷
7 [+]	1.12	× is a higher priority than + so ($4 \times 5^2 \times 7$) evaluated, 1.12 + is stored
3 [×]	3	3× is stored
.5 [y^x]	0.5	.5 y^x is stored
60 [2nd] [cos]	0.5	Cos 60° evaluated immediately
[=]	3.241320344	Complete all operations $.5^{\cos 60°}$ is evaluated, then $3 \times .5^{\cos 60°}$ is evaluated next, then this is added to 1.12.

Thus, with the expression entered, the calculator correctly interprets it as

$((4 \div 5^2) \times 7) + (3 \times 0.5^{(\cos 60°)})$

The important thing to remember here is that algebraic operations are performed strictly according to their relative priority as stated in the hierarchy. The calculator remembers all stored operations and recalls each with its associated number for execution at exactly the correct time and place. Once familiar with the order of these operations, you will find most problems are extremely easy to solve because of the straightforward manner in which they can be entered into the calculator. Additional control over the order of evaluation is provided through the use of parentheses.

Parentheses

Parentheses give you a way to "cluster" numbers and operations. By placing a series of numbers and operations in parentheses, you are instructing the calculator to interpret this expression first—down to a single number—and then proceed.

To illustrate the benefit of parentheses, try the following experiment: Press [( ] 5 [×] 7 [ )], and you will see the value 35 displayed. The calculator has evaluated 5 × 7 and replaced it with 35 even though the [=] was not pressed. Because of this function of parentheses, the algebraic rules now apply their hierarchy of operations within each set of parentheses. The use of parentheses ensures that your problem can be keyed in just as you would have written it. The calculator remembers each operation and evaluates each part of the expression as soon as all necessary information is available. When a close parenthesis is encountered, all operations back to the corresponding open parenthesis are completed. You should use parentheses if you have any doubts about how the calculator is going to handle an expression.

Even though expressions are normally written as $(3+2)(4+5)$ implying multiplication between parentheses sets, you must include [×] for the operation to take place. Your calculator does not perform implied multiplication.

Example: $4 \times (5+9) \div (7-4)^{(2+3)} = 0.230452675$

Key in this expression and follow the path to completion.

Press	Display	Comments
4 [×] [( ]	4	(4×) pending
5 [+]	5	(5+) pending
9 [ )]	14	(5+9) evaluated
[÷]	56	Hierarchy evaluates (4×14)
[( ]	56	and stores 56 ÷
7 [−]	7	(7−) pending
4 [ )]	3	(7−4) evaluated
[y^x] [( ]	3	Prepares for exponent
2 [+]	2	
3 [ )]	5	(2+3) evaluated
[=]	0.230452675	$(7-4)^{(2+3)}$ evaluated then divided into 4×(5+9)

There are limits on how many operations and associated numbers can be pending at one time. As many as nine parentheses can be open at any one time and eight operations can be pending, but only in the most complex situations would this limit be approached. If you do attempt to open more than 9 parentheses or if the calculator tries to store more than eight operations, Error results.

Example: 5 + 8/(9 − (2/3)) = 5.96

Press	Display	Comments
5 [+] [( ]	5	
8 [÷] [( ]	8	
9 [−] [( ]	9	
2 [÷] 3 [ )]	0.666666667	(2/3) evaluated
[ )]	8.333333333	(9 − (2/3)) evaluated
[ )]	0.96	8/(9 − (2/3))
[=]	5.96	5 + 8/(9 − (2/3))

Because the [=] key has the capability to complete all pending operations whenever it is used, it could have been used here instead of the three [ )] keys. Try working this problem again and pressing [=] instead of any closing parentheses keys.

Example: $3 \times 4^{(2^{(-\sqrt[4]{7})})} = 4.700043401$

Press	Display	Comments
[CLR] [( ]	0	
3 [×] [( ]	3	
4 [y^x] [( ]	4	
2 [y^x] [( ]	2	
7 [INV] [y^x]	7	
4 [ )]	1.626576562	$\sqrt[4]{7}$
[+/−]	−1.626576562	$-(\sqrt[4]{7})$
[ )]	0.323855789	$2^{-(\sqrt[4]{7})}$
[ )]	1.565681134	$4^{0.323855789}$
[ )]	4.700043401	$3 \times 4^{0.323855789}$

Each time a closed parenthesis is encountered, the contents are evaluated back to the nearest open parenthesis and are replaced with a single value. Knowing this, you can structure the order of evaluation to meet your needs. In particular, you can check intermediate results.

Number Copy Operation With Parentheses
An additional technique to use with parentheses is display reentry inside parentheses without having to key in the value again. Specifically, the calculator can bring a value inside parentheses if it is needed twice in succession in an expression. Any operation following an open parenthesis reenters the display register value. Follow the example below.

Example: 3.296214 + (3.296214 × 6) = 23.073498

Press	Display	Comments
[CLR] 3.296214 [+]	3.296214	
[( ] [×]	3.296214	Reenters 3.296214
6 [ )]	19.777284	
[=]	23.073498	

The long value 3.296214 had to be entered only once.

Algebraic Functions

The simplest operations to describe and understand are single-variable functions. These functions operate on the display register value immediately replacing this value with its corresponding function value. These functions do not interfere with any calculations in progress and can therefore be used at any point in a calculation. The accuracy of these functions is discussed in Appendix C and in text where necessary.

Reciprocal

[1/x], Reciprocal: Calculates the reciprocal of the value, x, in the display register by dividing x into 1. Error results if $x = 0$ or if $x > 1 \times 10^{99}$.

Press	Display
3.2 [1/x]	0.3125

Note that as soon as one of the math function keys is pressed, the displayed value is immediately replaced with its corresponding function value.

Logarithms

[lnx], Natural Logarithm: Calculates the natural logarithm (base e) of the value, x, in the display register. Error results if $x < 0$.

[2nd] [log], Common Logarithm: Calculates the common logarithm (base 10) of the value, x, in the display register. Error results if $x < 0$.

Example: $\log (1 + \ln 1.7) = 0.184869725$

Press	Display
[CLR]	0
[( ]1 [+]	1
1.7 [lnx] [ )]	1.530628251
[2nd] [log]	0.184869725

Powers of 10 and e

[INV] [lnx], Natural Antilog (e^x): Calculates the natural antilogarithm e^x, of the value, x, in the display register. $-227.9559242 \leq x \leq 230.2585092$ or Error will be displayed.

[INV] [2nd] [log], Common Antilog (10^x): Calculates the common antilogarithm 10^x, of the value, x, in the display register. $-99 \leq x \leq 99.99999997$ or Error will be displayed.

Example: $e^{(3+10^{0.3})} = 147.7116873$

Press	Display
[CLR] [( ] 3 [+]	3
.3 [INV] [2nd] [log] [ )]	4.995262315
[INV] [lnx]	147.7116873

Angle Calculations

Your calculator can make calculations involving angles measured in any of three units. You must specify the angle measurement unit by setting the angle mode.

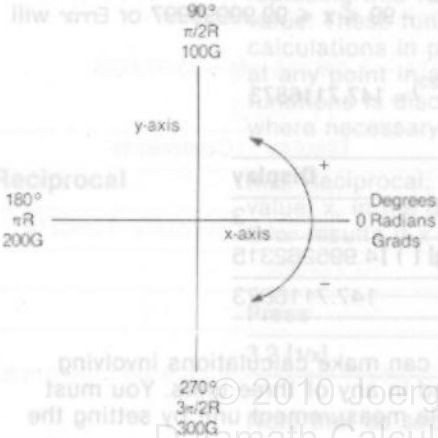
Angle Modes

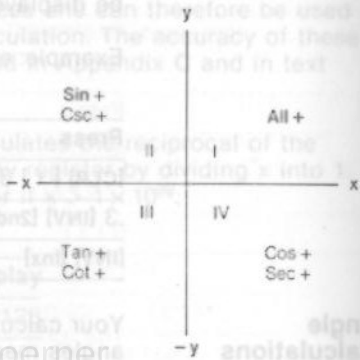
Angles can be measured in decimal degrees, radians, or grads (right angle = 90° = π/2 radians = 100 grads) by pressing either [2nd] [Deg], [2nd] [Rad] or [2nd] [Grad]. The calculator powers up in the degree mode and stays in that mode until altered by one of the other choices. Once in a certain angle mode, all entered and calculated angles are calculated in the units of that mode until another mode is selected or until the calculator is turned off. [CE] and [CLR] do not affect the angle mode.

The angle mode has absolutely no effect on calculations unless the trigonometric functions or polar to rectangular conversions are being performed. Selecting the correct angle mode is easy to do—AND EASY TO FORGET. Neglecting this step is responsible for many errors in operating any calculation device that offers a choice of angle units.

Trigonometric Functions

[2nd] [sin], [2nd] [cos], [2nd] [tan]: Sine, Cosine, Tangent of the value in the display register. All angles are measured from the x-axis. Measure counterclockwise for positive angles, clockwise for negative, as shown below.





The diagram shows in which quadrant, I through IV, the listed trigonometric functions are positive. Those functions not listed in a particular quadrant have negative values.

When measuring angles, remember that each angle has an equivalent with the opposite sign. For example −45° = 315°.

The trig functions operate using decimal degrees, not minutes and seconds. If your angle is expressed in degrees, minutes and seconds, you can use the [DMS-DD] key to convert it to decimal form. See Conversions later in this chapter. For angle calculations involving degrees, check that your calculator is in the degree mode (DEG indicator on). If it is not, press [2nd] [Deg].

Example: sin 30°13′48″ + tan 315° = −0.496527589

Press	Display
[2nd] [Deg]	
30.1348 [2nd] [DMS-DD]	30.23
[2nd] [sin] [+]	0.503472411
315 [2nd] [tan]	−1
[=]	−0.496527589

Trigonometric values can be calculated for angles greater than one revolution. As long as the trigonometric function is displayed in standard format rather than in scientific notation, all digits displayed in standard display format are accurate to ±1 in the 10th digit for the range of ±36,000 degrees, ±200π radians or ±40,000 grads. In general, the accuracy decreases one digit for each decade outside of this range. The maximum angle accepted for trigonometric functions is ±449999999°, ±2499999π radians, or ±499999999 grads. See *Appendix C* for more on accuracy.

The other trig functions can be calculated almost as easily.

csc = [2nd] [sin] [1/x]
sec = [2nd] [cos] [1/x]
cot = [2nd] [tan] [1/x]

Inverse Trigonometric Functions

[INV]. Inverse: Preceding another key, [INV] reverses the function of that key. When used with the trigonometric functions, the inverse of those functions is obtained. For example, arcsine ($\sin^{-1}$) is obtained by pressing [INV] [2nd] [sin].

The inverse trigonometric functions calculate the angle whose functional value is in the display. The largest angle resulting from an arc function is 180 degrees (π radians or 200 grads). Because these functions have many angle equivalents, i.e., arcsin .5 = 30°, 150°, 390°, etc., the angle returned by each function is restricted as follows:

Function	Range of Resultant Angle
arcsin of	
positive argument	0 to 90°, $\pi/2$ radians, or 100 grads
negative argument	0 to −90°, $-\pi/2$ radians, or −100 grads
arccos of	
positive argument	0 to 90°, $\pi/2$ radians, or 100 grads
negative argument	90° to 180°, $\pi/2$ to π radians, or 100 to 200 grads
arctan of	
positive argument	0 to 90°, $\pi/2$ radians, or 100 grads
negative argument	0 to −90°, $-\pi/2$ radians, or −100 grads

For arcsin x and arccos x, x must be in the range $-1 \leqslant x \leqslant 1$.

Example: $\pi/4 + \tan^{-1}(.2\pi) = 1.34638028$

Press	Display
[2nd] [Rad]	
[2nd] [π] [÷]	3.141592654
4 [+]	0.785391634
[( ] .2 [×] [2nd] [π] [ )]	0.628318531
[INV] [2nd] [tan]	0.560982116
[=]	1.34638028

The selection of the radian mode could have been made at any point before [INV] [2nd] [tan]. It is generally best, though, to select the angle mode at the start of a problem. This assures that the mode is correctly set before you get involved in keying in the problem. The angle mode, whenever selected, only affects angle measurements.

The inverses of the other trig functions can be calculated as follows.

arccsc = [1/x] [INV] [2nd] [sin]
arcsec = [1/x] [INV] [2nd] [cos]
arccot = [1/x] [INV] [2nd] [tan]

Degree, Radian, Grad Conversions
It is frequently necessary to convert angle values from one unit of measurement to another. Use the following table of conversion factors for the purposes you need.

FROM TO	degrees	radians	grads
degrees		$\times \pi \div 180$	$\div 0.9$
radians	$\times 180 \div \pi$		$\times 200 \div \pi$
grads	$\times 0.9$	$\times \pi \div 200$	

These operations can be performed in any angle mode setting of the calculator.

Example: Convert 120 degrees to radians and grads.

Press	Display	Comments
120 [×] [2nd] [π] [÷]	376.9911184	
180 [=] [×] 200 [÷] [2nd] [π] [=]	2.094395102 133.3333333	Radians Grads
[×]	133.3333333	
.9 [=]	120	Degrees

Because of the independence of these conversions from the angle mode of the calculator, you must be careful when using the results for further calculations. The angle mode must be selected to match the units of the results.

Integer and Absolute Value

[2nd] [Intg], Integer: Discards the fractional part of the number in the display register. [INV] [2nd] [Intg] discards the integer portion of the number in the display register.

[2nd] [|x|], Absolute Value: Makes the value in the display register positive.

Example: Find the absolute value of the integer portion of −13/5.

Press	Display
13 [+/−] [÷]	−13
5 [=]	−2.6
[2nd] [Intg]	−2
[2nd] [|x|]	2

These functions are particularly useful in programming sequences.

Remember that the integer sequence operates on the value in the display register, not on the displayed value. This means that when [2nd] [Intg] is pressed and 4.9999999999 is in the display register (which rounds to 5 in the display), 4 is the integer that remains in the display. To make the integer portion 5, press [EE] [INV] [EE] to round the internal number to the displayed value before pressing [2nd] [Intg].

In scientific notation, the *actual* fractional part of a number is discarded by the integer key, not the apparent fraction. For instance, 1.2345×10^3 [2nd] [Intg] yields 1.234×10^3; actually 1234.5 becomes 1234.

Square and Square Root

[x^2], Square: Calculates the square of the number in the display register. If $x \geqslant 10^{\pm 50}$, Error results.

[$\sqrt{x}$], Square Root: Calculates the square root of the number in the display register. If x is negative, Error results.

Example: $(\sqrt{3.1452} - 7 + (3.2)^2)^{1/2} = 2.239078197$

Press	Display
[CLR] [( ]	0
3.1452 [$\sqrt{x}$] [−]	1.773471173
7 [+]	−5.226528827
3.2 [x^2]	10.24
[ )]	5.013471173
[$\sqrt{x}$]	2.239078197

Powers and Roots

[y^x], Powers: Raises the display register value, y to the x power. The entry sequence is y [y^x] x followed by an operation or equals key. If $y < 0$ or if $y = 0$ and $x < 0$, Error is displayed.

[INV] [y^x], Roots ($\sqrt[x]{y}$ or $y^{1/x}$): Takes the x root of the value, y, in the display register. The entry sequence is y [INV] [y^x] x followed by an operation or equals key. If $y < 0$ or $x = 0$ or if $y = 0$ and $x < 0$, Error is displayed. When x and y both equal zero, the answer is 1.

These math functions do not act on the display register immediately. They require entry of a second value followed by an operation before the function can be completed.

Example: $\sqrt[3]{2.36^{-.23}} = .9362893421$

Press	Display	Comments
2.36 [y^x]	2.36	Enter y for y^x
.23 [+/−]	−0.23	Enter x for y^x
[INV] [y^x]	0.820786565	Produces y for $\sqrt[x]{y}$
3 [=]	0.936289342	Enter x for $\sqrt[x]{y}$ to produce the answer

The y^x function uses logarithms to evaluate this functions and the standard mathematical definitions yield the following results to various x and y combinations.

		Function Result	
y	x	y^x	$\sqrt[x]{y}$
0	0	1	1
0	negative	Error	Error
0	positive	0.	0.
positive	0	1.	Error
negative	any number	Error	Error

Memory Capabilities

User data memories allow you to store or accumulate data for later use. These storage areas are generally just referred to as user data memory or data memory as opposed to program memory where programs are stored. You can use the memory keys at any point in a calculation because they do not affect calculations in progress.

Usually, all data memories currently partitioned are equally suited to storing values. However, data memories 01-06 are used internally when working statistics calculations. If you want to preserve any values and perform these statistical calculations, use data memories other than 01-06. If you are using very many data memories for storage, you will probably need some form of bookkeeping to remember what values are stored in which memories.

Selection of Memory Size (Partitioning)

There is a memory storage area within your calculator. This area is for data storage and program storage. When the calculator is off, this memory area is preserved by the constant memory. This area can be depicted as follows.

Last program step									
N/A*	7	15	23	...	479	487	495	503	511
Highest user data memory									
63	62	61	60	....	3	2	1	0	N/A*

*None available

which can be stated algebraically as:

Highest user data memory =
(511 − last program step) ÷ 8 − 1

or

Last program step =
511 − 8 × (Highest user data memory + 1)

You partition the memory area for a number of user data memories from 0 to 64. What is not partitioned as user data memory becomes program steps. The trade-off is eight program steps for each user data memory. The indication given when partitioning tells you the last available program step and the highest available memory. When partioning is for 2 user data memories, the available memories are 0 and 1. The number of user data memories is one more than the highest available memory.

Remove all fix-decimal, scientific notation and engineering formats before partitioning. You can partition this memory area into the combination you desire by groups of ten user data memories or by single user data memories. To partition the storage area, enter the number of sets of 10 user data memories you need, 0-6, and press **[OP]** 17 or press **[2nd] [Part]** and the desired number of user data memories. For 20 user data memories, press 2 **[OP]** 17 and the display shows 351.19

This shows that there are 20 user data memories, 00-19, available for data storage and 352 program steps, 000-351, allocated for program storage.

Equivalently, you could have pressed **[2nd] [Part]** 20. The difference is that the partition key partitions memory for any number of user data memories from 0 to 64 and the OP code partitions in increments of ten from 0 to 60.

To check the current placement of the partition at any time, press **[OP]** 16 and the existing partition is displayed. The power-up partitioning is 255.31.

Because you can use up to 64 data memories, you must specifiy which data memory you are using by entering its two-digit address, XX, immediately after pressing any memory-related key. You can use a short form of addressing and enter a single digit address if the address is less than 10 and you follow it with a nonnumeric key entry. This versatile memory system can manipulate data in a variety of ways.

Clearing Data Memory

[2nd] [CMs], Clear Data Memory: Instructs the calculator to clear all user data memories as defined by the current partition.

Use of this key does not affect the t register, program memory, memory partitioning, the display, or calculations in progress.

Storing and Recalling Data

[STO] XX, Store: Stores the display register value in user data memory XX. Any previously stored data in that memory is lost.

[RCL] XX, Recall: Recalls and displays the value stored in user data memory XX and retains the value in data memory XX. A recalled number can be used as a number entry in any mathematical expression.

Example: Store and recall 3.012 in memory 22.

Press	Display
3.012 [STO] 22	3.012
[CLR]	0
[RCL] 22	3.012

Use of these keys can save you keystrokes by storing long numbers that are to be used several times.

Example: Evaluate $3x^2 - x - 7.1$ for $x = 2.9467281$

Press	Display
[CLR] 3 [×]	3
2.9467281 [STO] 12	2.9467281
[x^2] [−]	26.04961949
[RCL] 12	2.9467281
[−] 7.1 [=]	16.00289139

The long value of x only had to be entered once. The storage and recall did not interfere with calculations in progress.

You can also use the data memories to hold intermediate results as well as repetitive numbers.

Example: Evaluate $\frac{(\sin(3x/2) - \cos(3x/2))}{x}$ for x = 20.6821776 degrees.

Press	Display	Comments
[2nd] [Deg]		
[2nd] [CMs] [( ] [( ] 3 [×]	3	
20.6821776 [STO] 14	20.6821776	Store x in memory 14
[÷] 2 [ )] [STO] 17	31.0232664	Store 3x/2 in memory 17
[2nd] [sin] [−]	0.515386107	
[RCL] 17	31.0232664	Recall 3x/2 from memory 17
[2nd] [cos] [ )] [÷]	−0.341571979	
[RCL] 14	20.6821776	Recall x from memory 14
[=]	−0.016515281	Answer

Attempting to use a user data memory beyond the partition causes Error to be displayed.

Memory Arithmetic

You can store a displayed number at any time during a calculation without affecting the calculation in any way. Additionally, you can add, subtract, multiply and divide into any data memory. The display register itself is not changed when using a memory arithmetic key. Error following one of these operations indicates that you have exceeded the calculator's operating limit in that memory (assuming that you did not call for a register number outside of the current partition, which also causes Error).

[SUM] XX, Memory Sum: adds the display register value to the contents of data memory XX and stores the result in XX.

[INV] [SUM] XX, Memory Subtract: Subtracts the display register value from the contents of data memory XX and stores the result in XX.

[2nd] [Prd] XX, Memory Product: Multiplies the contents of data memory XX by the display register value and stores this product in XX.

[INV] [2nd] [Prd] XX, Memory Divide: Divides the contents of data memory XX by the display register value and stores the result in XX.

These capabilities provide an alternative to using the display register for arithmetic operations and save steps when the result is to replace a number in memory.

Example: Evaluate $x^2 + 9$ for $x = -1$, 2, and 3 and total the results.

Press	Display	Memory 3
1 [+/−] [x²] [+]	1	Unknown
9 [=] [STO] 03	10	10
2 [x²] [+]	4	10
9 [=] [SUM] 03	13	23
3 [x²] [+]	9	23
9 [+]	18	23
[RCL] 3	18	23
[=]	41	23

Notice that the first evaluation was placed in memory 03 using the [STO] key. This is a recommended procedure when performing memory arithmetic to ensure that you have only the values you intend to accumulate in that particular memory. The [STO] discards any previous content of that memory by replacing it with the new value.

Example: The percentage of students completing each year at a particular college is 76.8% first year, 81.3% second year, 92.2% third year, and 95.9% last year. What percentage of the students graduate and what percentage complete their third and fourth years?

Press	Display
.768 [×]	0.768
.813 [×]	0.624384
.922 [STO] 11 [×]	0.575682048
.959 [2nd] [Prd] 11 [=]	0.552079084
[RCL] 11	0.884198

About 55% of the students that enter the school graduate. Over 88% of those entering their junior year graduate.

Memory/Display Exchange

[2nd] [Exc] XX, Memory Exchange: Exchanges the contents of data memory XX with the display register. The display value is stored in memory XX and the previously stored value is displayed.

The exchange key has several uses in addition to saving keystrokes. You can use it to examine two calculated results without losing either. Also, numbers can be temporarily stored in memory XX and used as needed.

Example: Evaluate $A^2 + AB + 2B^2$ for A = .258963 and B = 1.255632

Press	Display	Comments
.258963 [STO] 13	0.258963	Store A in memory 13
[x²] [+] 1.255632 [×]	1.255632	Enter B
[2nd] [Exc] 13	0.258963	Store B, recall A
[+] 2 [×]	2	
[RCL] 13	1.255632	Recall B from memory 13
[x²] [=]	3.545447504	Answer

Special Control Operations

There are operations accessed through use of the [OP] key that increase the capabilities of your calculator. Some of these special operations can be used in any calculator mode while others are designed for a specific mode or for use with the optional PC-200 printer.

Each special control operation is called by pressing [OP] nn where nn is the 2-digit code assigned to each operation. Short form addressing can be used for OP codes. Each OP code is briefly described below, with complete definitions following. When nn > 39, Error is displayed.

Code nn	Function
00*	Initializes the print register.
01*	Enters 8 digits in the display as 4 alphanumeric codes for far left quarter of print column.
02*	Enters 8 digits in the display as 4 alphanumeric codes for inside left quarter of print column.
03*	Enters 8 digits in the display as 4 alphanumeric codes for inside right quarter of print column.
04*	Enters 8 digits in the display as 4 alphanumeric codes for far right quarter of print column.
05*	Prints the contents of the print register.
06*	Prints the 4 characters of OP 04 with current display value.
07*	Plots an * in column 0-15 as specified by the display.
08*	Lists the labels currently used in program memory.
09	Error.
10	Applies signum function to display register value.

*Designed specifically for use with the optional PC-200 printer

11	Calculates variances.
12	Calculates slope and intercept.
13	Calculates correlation coefficient.
14	Calculates new y prime (y') for an x in the display.
15	Calculates new x prime (x') for a y in the display.
16	Displays current partition of memory storage area.
17	Repartitions memory storage area.
18	If no error condition is in effect, sets flag 7.
19	If an error condition is in effect, sets flag 7.
20-29	Increment a data memory 0-9 by 1.
30-39	Decrement a data memory 0-9 by 1.

Printer Capabilities, [OP] 00-08

These control operations are designed for use with the optional PC-200 printer. The printer increases the flexibility of your calculator by providing "hard copy" results of your calculations. The control operations further expand the benefit of the printer by furnishing the options to print alphanumeric messages, to plot data and to list the labels present in a program along with their program locations. The PRINTER CONTROL chapter explains the use of each of these special control operations.

Error, [OP] 09

When encountered in a program, [OP] 09 stops execution and causes Error to be displayed.

Signum Function, [OP] 10

Special control [OP] 10 applies the signum function to the value "x" currently in the display register and responds with the following.

Display Register Value x	Display Response
x> 0	1.
x = 0	0.
x< 0	− 1.

Statistics,
[OP] 11-15

The special control operations 11-15 are used for statistical analysis and are discussed fully in the section on Statistics.

Partitioning,
[OP] 16-17

Pressing [OP] 16 immediately displays the highest available program step and data memory, separated by a decimal point. Remember that program step and user data memory numbering both begin with 0.

To partition the memory area, enter the number of sets of 10 user data memories needed and press [OP] 17 and the new partition is displayed. See the section covering Selection of Memory Size for the details of partitioning.

Test Operations,
[OP] 18-19

Operations 18 and 19 are designed to monitor the error status of a program that is running. When encountered in a program, [OP] 18 sets flag 7 if no error condition exists. [OP] 19 sets flag 7 if an error condition does exist. Flag 7 can then be monitored by the "if flag" test instruction and appropriate action can then be taken from the results of the test. See the section on Flags later in this chapter for more information.

Increment/ Decrement Data Memories,
[OP] 20-29/30-39

The OP instruction also allows you to increment or decrement the contents of any user data memory 0-9 by 1 without altering the display register.

To increment memory n by 1, press [OP] 2n, where n is a user data memory number 0-9.

To decrement memory n by 1, press [OP] 3n, where n is a user data memory number 0-9.

Each time either of these sequences is pressed or encountered in a program the contents of memory n are appropriately adjusted by 1 regardless of the content of memory n.

For example, [OP] 34 subtracts 1 from the contents of memory 4.

Conversions

Your calculator can readily convert between the polar and rectangular coordinate systems. It can also transform angles expressed in degrees, minutes, and seconds to decimal degrees and vice versa.

Angle Conversions

[2nd] [DMS-DD], Degrees/Minutes/Seconds To Decimal Degrees: Converts an angle measured in degrees, minutes, and seconds to its decimal degrees equivalent. [INV] [2nd] [DMS-DD] reverses this conversion. Minutes and seconds can each be any two-digit number. This conversion operates on the displayed value only.

The input format for degrees, minutes and seconds is to place a decimal point between the degrees and minutes, DD.MMSSsss.

Press	Display	Comments
47.131272 [2nd] [DMS-DD]	47.2202	DD.dddd
[INV] [2nd] [DMS-DD]	47.131272	DD.MMSSss

DD represents degrees and dddd is for the decimal fraction of a degree. MM is minutes and SSss is for seconds and fractional parts of seconds. Be sure to use two digits each for minutes and seconds. For instance, 5°4'3" must be entered as 5.0403 to be interpreted correctly.

The input range of [2nd] [DMS-DD] is $.001 \leqslant |x| \leqslant 9.999999 \times 10^{99}$, which is also the range of [INV] [2nd] [DMS-DD]. Therefore, [2nd] [DMS-DD] should not be used for angles less than 10 seconds and [INV] [2nd] [DMS-DD] should not be used for angles less than .001 degrees.

The same key can be used to convert hours, minutes, and seconds to decimal hours and vice versa. If the minutes portion of the entry is 60 or more, the number of degrees or hours in the answer will exceed the number of degrees or hours in the entry and the reverse conversion will not be the same as the original entry.

Try converting 1 hour and 90 minutes to decimal hours and back.

Press	Display	Comments
[CLR]	0	Clear display and pending operations and select the degree mode
1.90 [2nd] [DMS-DD]	2.5	Result in decimal hours
[INV] [2nd] [DMS-DD]	2.3	Result returned to hours/minutes/seconds

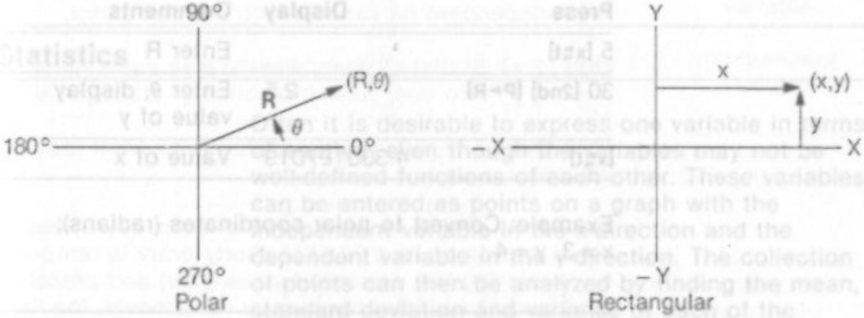
Polar/Rectangular System Conversions

[x⇆t], x Exchange t: Exchanges the display register value x with the t register value t. The uses of this key include data entry for coordinate conversions, statistics and storing a value for making a comparison in a program.

The t register is independent of all other storage areas, but functions essentially like a data memory. It is only externally accessible through the [x⇆t] key that places the display register value into the t register and brings the contents of the t register into the display register and the display as well. The t register is cleared or altered by CP, polar to rectangular conversion, CSR, statistics calculations, and turning off the calculator.

[2nd] [P►R], Polar/Rectangular: Converts polar coordinates to rectangular. [INV] [2nd] [P►R] converts rectangular coordinates to polar.

The polar/rectangular conversion is used to convert from polar coordinates which describe any point by a radius, R, and an angle, θ, to rectangular coordinates which describe any point by two vectors, x, and, y, measured at right angles to each other.



Polar To Rectangular Sequence
- Enter R
- Press [x⇆t]
- Enter θ
- Press [2nd] [P►R] to display y
- Press [x⇆t] to display x

Rectangular to Polar Sequence
- Enter x
- Press [x⇆t]
- Enter y
- Press [INV] [2nd] [P►R] to display θ
- Press [x⇆t] to display R

Be sure to set the desired angle mode for θ before entering or calculating the angle.

The θ calculated from the rectangular to polar sequence is:

$-180° \leq \theta \leq 180°$ and the radian and grad equivalents.

Thus the calculated angles that occur in the third and fourth quadrants are displayed as negative angles.

This conversion routine monitors the angle mode of the calculator to determine the angle units desired for both entry and resultant data.

Example: Convert to rectangular coordinates: R = 5, $\theta = 30°$

Set angle mode to degrees.

Press	Display	Comments
5 [x⇌t]		Enter R
30 [2nd] [P►R]	2.5	Enter θ, display value of y
[x⇌t]	4.330127019	Value of x

Example: Convert to polar coordinates (radians): x = 3, y = 4

Press	Display	Comments
[CLR] [2nd] [Rad]	0	Set angle mode to radians
3 [x⇌t]		Store x
4 [INV] [2nd] [P►R]	0.927295218	Enter y, display vaue of θ radians
[x⇌t]	5	Value of R

Statistics Considerations

The calculator uses memories 1-6 and the t register for statistics calculations.

Statistics

Often it is desirable to express one variable in terms of another, even though the variables may not be well-defined functions of each other. These variables can be entered as points on a graph with the independent variable in the x-direction and the dependent variable in the y-direction. The collection of points can then be analyzed by finding the mean, standard deviation and variance of each of the variables. A straight line can be fitted to the points (using the technique of linear regression) with the slope and intercept of the line accessible to the user. Then additional data points can be interpolated or extrapolated and a correlation coefficient is available to tell you how closely the line approximates the collection of data points.

Data Entry

[2nd] [CSR], Clear Statistics Registers: Initializes the calculator for statistics by zeroing user data memories 1-6 and the t register.

[x⇆t], x Exchange t: Exchanges the t register value, t, with the display register value, x. In statistics, enters the independent (x) variable into the calculator.

[2nd] [Σ+], Statistics Sum: Assimilates each variable pair (x_i, y_i) into user data memories 01 through 06.

[INV] [2nd] [Σ+] removes unwanted data point (pair of values).

To enter a single variable array of data, key in each value and press [2nd] [Σ+]. A faulty entry is removed by reentering the unwanted value (pair) and pressing [INV] [2nd] [Σ+]. After each entry (or removal) the total number of values entered is displayed.

Two-dimensional statisical data are entered using the following key sequence for each data point (x_i, y_i) where i = 1, 2, 3, ... N.

x_i [x⇆t] y_i [2nd] [Σ+]

The data point number is displayed after each point is entered. To remove an unwanted data point, reenter both the undesired x and y values again, but press [INV] immediately before [2nd] [Σ+]. The total number of points N input thus far is automatically decremented by 1.

As each data point is keyed in, it is assimilated into data memories 01 through 06 as follows.

DATA MEMORY	CONTENTS	
01	Σy	dependent
02	Σy^2	variable
03	N	
04	Σx	independent
05	Σx^2	variable
06	Σxy	

Data that have already been so grouped can be entered directly into these registers and analysis can begin immediately.

[2nd] [Σ+] sums the incoming values, their squares, their product and counts them in these memory registers. The contents of these registers should initially be cleared to prevent an erroneous accumulation of statistical data. The key sequence [2nd] [CSR] should be used to zero registers 01 through 06 and the t register before data entry.

Mean, Variance, and Standard Deviation

After all the data are entered (or at any intermediate point after two or more data points have been entered), the mean, standard deviation, and variance of each array of data can be calculated.

[2nd] [$\overline{x}$], Mean of Data: Calculates and displays the mean of the dependent (y-array) data. Press [x⇌t] next to display the mean of the independent (x-array) data.

[INV] [2nd] [$\overline{x}$], Standard Deviation: Calculates and displays the standard deviation of the dependent (y-array) data. When [x⇌t] is pressed next, the standard deviation of the independent (x-array) data is displayed.

The following equations are used by the calculator.

Mean of x-array $= \overline{x} = \dfrac{\Sigma x}{N}$

Mean of y-array $= \overline{y} = \dfrac{\Sigma y}{N}$

where N is the total number of data points entered.

$$\text{Standard deviation of x-array} = \sigma_x^2 = \left[\frac{\Sigma x^2 - \dfrac{(\Sigma x)^2}{N}}{N-1}\right]^{1/2}$$

$$\text{Standard deviation of y-array} = \sigma_y^2 = \left[\frac{\Sigma y^2 - \dfrac{(\Sigma y)^2}{N}}{N-1}\right]^{1/2}$$

$$\text{Variance of x-array} = \sigma_x^2 = \frac{\Sigma x^2}{N} - \overline{x}^2$$

$$\text{Variance of y-array} = \sigma_y^2 = \frac{\Sigma y^2}{N} - \overline{y}^2$$

For your convenience, the option has been provided to select N or N – 1 weighting for standard deviation and variance calculations. N weighting results in a maximum likelihood estimator that is generally used to describe populations, while the N – 1 is an unbiased estimator customarily used for sampled data.

The variance function uses N weighting and standard deviation uses N – 1 weighting. Variance is the square of the standard deviation by definition. So, to find the variance with N – 1 weighting, press [INV] [2nd] [$\bar{x}$] [x^2] and [x⇌t] [x^2]. Standard deviation with N weighting is found by pressing [OP] 11 [$\sqrt{x}$] and [x⇌t] [$\sqrt{x}$]. See the table below for these key sequences.

Function	Weighting	Key Sequences y-array	x-array
Mean		[2nd] [$\bar{x}$]	[x⇌t]
Standard Deviation	N	[OP] 11 [$\sqrt{x}$]	[x⇌t] [$\sqrt{x}$]
Variance	N	[OP] 11	[x⇌t]
Standard Deviation	N – 1	[INV] [2nd] [$\bar{x}$]	[x⇌t]
Variance	N – 1	[INV] [2nd] [$\bar{x}$] [x^2]	[x⇌t] [x^2]

For single-variable data, you do not need to use the [x⇌t] key as this key is needed only for entering and displaying attributes of independent variables (x-array). Memory registers 01 through 06 and the t register are all still used.

Example: Analyze the following test scores: 96, 81, 87, 70, 93, 77

Press	Display	Comments
Press	Display	Comments
[2nd] [CSR]		Initialize
96 [2nd] [Σ+]	1	1st Entry
81 [2nd] [Σ+]	2	2nd Entry
97 [2nd] [Σ+]	3	3rd Entry (incorrect)
97 [INV] [2nd] [Σ+]	2	Remove 3rd Entry
87 [2nd] [Σ+]	3	Correct 3rd Entry
70 [2nd] [Σ+]	4	4th Entry
93 [2nd] [Σ+]	5	5th Entry
77 [2nd] [Σ+]	6	6th Entry
[INV] [2nd] [$\overline{x}$]	9.879271228	Standard Deviation
[2nd] [$\overline{x}$]	84	Mean
[OP] 11	81.33333333	Variance
[RCL] 01	504	Total of Scores (Σy stored in memory 1)

Note that the standard deviation can be calculated first even though the mean is used to determine the standard deviation.

Example: A quantity of tubing that has been ordered cut into 100 cm sections is to be checked for length accuracy and uniformity that should be 6.0 g/cm ±0.01. The test requires that 6 samples be analyzed at a time.

Sample	1	2	3	4	5	6
Length (cm)	101.3	103.7	98.6	99.9	97.2	100.1
Weight (g)	609	626	586	594	579	605

What is the average weight of the samples taken? How accurate is the cutting machine? What is the uniformity of the samples?

Press	Display	Comments
[2nd] [CSR]		Intialize
101.3 [x⇄t]	0	Enter x_1
609 [2nd] [Σ+]	1	Enter y_1
103.7 [x⇄t]	102.3	Enter x_2
626 [2nd] [Σ+]	2	Enter y_2
98.6 [x⇄t]	104.7	Enter x_3
586 [2nd] [Σ+]	3	Enter y_3
99.9 [x⇄t]	99.6	Enter x_4
594 [2nd] [Σ+]	4	Enter y_4
97.2 [x⇄t]	100.9	Enter x_5
579 [2nd] [Σ+]	5	Enter y_5
100.1 [x⇄t]	98.2	Enter x_6
605 [2nd] [Σ+]	6	Enter y_6
[2nd] [$\overline{x}$]	599.8333333	Average of y array (weight)
[÷] [x⇄t]	100.1333333	Average of x array (length)
[=]	5.990346205	Average uniformity (g/cm)
[INV] [2nd] [$\overline{x}$]	17.05774507	Average deviation
[x⇄t]	2.240238083	Length deviation

The average weight of the samples is about 599.8 grams. The machine is cutting the length to about 100.1 centimeters. The uniformity is better than 5.99 grams/centimeter, easily within the acceptable tolerance. In addition, the standard deviation of the weights of the various pieces is about 17 grams with the deviation of the lengths about 2-1/4 centimeters on the average.

Linear Regression

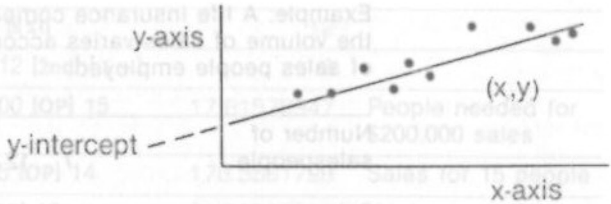
[OP] 12: Calculates and displays the y-intercept of the line fitted to the data points. [x⇌t] when pressed after [OP] 12 displays the slope of the line fitted to the data points. Data points that depict a vertical line are a special case that has no y-intercept and has an infinite slope and is an invalid operation for this calculator.

[OP] 13: Calculates and displays the correlation coefficient of the individual data points in relation to the line fitted to these points. The value will be between −1 and 1 with +1 being a perfect correlation. If the slope of the line is 0 or infinite, Error is displayed. This condition can be monitored in a program through use of [OP] 19 and program flags.

[OP] 14: Computes and displays a linear estimate of y' on the linear regression line corresponding to the x in the display. If the previously input data represent a vertical line (infinite slope) or if the slope is 0, Error is displayed.

[OP] 15: Computes and displays a linear estimate of x' on the linear regression line corresponding to the y in the display. If the previously input data represent a vertical line (infinite slope) or if the slope is 0, Error is displayed.

Data entry for linear regression is the same as for mean, standard deviation and variance calculations that can also be used here. Actually, once a data set is entered, all the statisical functions can be used to analyze the data. Linear regression allows you to analyze one variable's relationship to another. The method is to perform a least-squares linear regression which is designed to minimize the sum of the squares of the deviations of the actual data points from the straight line of best fit. In practice, a plot of the data points is made and a line is constructed that uniformly divides the data points. Using this method, the square root of the squares of their perpendicular offsets is minimized.



This line is described by $y = mx + b$, where m is the slope of the line and b is the y-intercept.

Because the data are seldom perfectly linear, you can measure how well the line fitted to the data actually does approximate the data. This measure is called the correlation coefficient and may be calculated from the data and the linear equation parameters.

The slope and y-intercept of the regression line are determined as follows:

$$\text{slope} = m = \frac{\Sigma xy - \frac{\Sigma x \Sigma y}{N}}{\Sigma x^2 - \frac{(\Sigma x)^2}{N}}$$

$$\text{y-intercept} = b = \frac{\Sigma y - m\Sigma x}{N}$$

$$\text{The correlation coefficient} = R = \frac{m\,\sigma_x}{\sigma_y}$$

Additional data points can be predicted by choosing a new x or y value and having the calculator compute a corresponding y or x value on the regression line. This process uses the line equation $y = mx + b$, where m (slope) and b (y-intercept) are determined from the data previously submitted.

Example: A life insurance company has found that the volume of sales varies according to the number of sales people employed.

Number of salespeople	7	12	3	5	11	8
Sales in thousands/mo.	99	152	81	98	151	112

How many salespeople does this company need for $200,000 monthly sales? What monthly sales should 15 salespeople generate?

Press	Display	Comments
[2nd] [CSR]		Initialize
7 [x⇄t]	0	First x value
99 [2nd] [Σ+]	1	Data point 1
12 [x⇄t]	8	Second x
152 [2nd] [Σ+]	2	Data point 2
3 [x⇄t]	13	etc.
81 [2nd] [Σ+]	3	
5 [x⇄t]	4	
98 [2nd] [Σ+]	4	
22 [x⇄t]	6	Incorrect entry
151 [2nd] [Σ+]	5	
22 [x⇄t]	23	Remove incorrect entry
151 [INV] [2nd] [Σ+]	4	
11 [x⇄t]	22	
151 [2nd] [Σ+]	5	

8 [x⇄t]	12	
112 [2nd] [Σ+]	6	
200 [OP] 15	17.81578947	People needed for $200,000 sales
15 [OP] 14	176.5561798	Sales for 15 people
[OP] 12	51.66853933	Y-intercept of line
[x⇄t]	8.325842697	Slope of line

The slope and y-intercept have been calculated so that the line can be plotted, if desired. The slope is incremental sales per person. The y-intercept is independent sales.

By performing any of the math functions on one or both elements of the random-variable pair, other types of regression are available. For example, by taking the logarithm of one of the variables before entering it as a data point, you can obtain a semilogarithmic curve fit. These variations can be achieved by using natural logarithms, exponentials, roots and powers, and reciprocals.

When initially analyzing your data, you must select the type of curve that characterizes your particular situation. You can try several types of curves to see which best fits your needs.

Example: A city published the following census data. Predict the population in the year 1990 and predict the year the population will be 50,000 inhabitants.

Year	1940	1950	1960	1970	1980
Population	3221	5361	9212	15410	27612

Population data characteristically follow an exponential curve of the form $y = ae^{bx}$. Taking the log of both sides of this equation yields $\ln y = \ln a + bx$. Therefore by plotting x vs. lny (semilog), you can plot a straight line.

Press	Display
[2nd] [CSR]	
1940 [x⇌t]	0
3221 [lnx] [2nd] [Σ+]	1
1950 [x⇌t]	1941
5361 [lnx] [2nd] [Σ+]	2
1960 [x⇌t]	1951
9212 [lnx] [2nd] [Σ+]	3
1970 [x⇌t]	1961
15410 [lnx] [2nd] [Σ+]	4
1980 [x⇌t]	1971
27612 [lnx] [2nd] [Σ+]	5
1990 [OP] 14 [INV] [lnx]	46081.80973
50000 [lnx] [OP] 15	1981.524472

The population in 1990 should be approximately 46,082 and the town should have 50,000 residents in 1991.

Trend-Line Analysis

For data collected at periodic intervals, such as yearly or daily, or data accumulated per event, the calculator can automatically increment the value of x by 1 for each data point entered. The calculator initially assigns whatever value is in the t register for the x value of the first data point, then adds 1 for the second, 1 for the third, etc. All data points are entered by pressing the value followed by [2nd] [Σ+] only. The starting x value can be set to any number by entering the first x value, then letting the calculator increment from there: x_1 [x⇄t], y_1 [2nd] [Σ+], y_2 [2nd] [Σ+], y_3 [2nd] [Σ+], etc.

To remove the previous data entry, press [x⇄t] [−] 1 [=] [x⇄t] and then enter the unwanted y value.

Example: A computer dating service has the following annual profits:

Year	1972	1973	1974	1975-1980	1981	1982	1983	1984
Profit in millions	− 2.1	− 0.3	0.8	inactive	2.9	2.8	3.6	4.0

What profit can be expected in 1990 and when will the company break the $10 million mark?

Press	Display	Comments
[2nd] [CSR]		Initialize
1972 [x⇆t]	0	Initialize x
2.1 [+/−] [2nd] [Σ+]	1	1972 loss
.3 [+/−] [2nd] [Σ+]	2	1973 loss
.8 [2nd] [Σ+]	3	1974 gain
1981 [x⇆t]	1975	Reinitialize x
2.9 [2nd] [Σ+]	4	1981 gain
2.8 [2nd] [Σ+]	5	1982 gain
3.6 [2nd] [Σ+]	6	1983 gain
4 [2nd] [Σ+]	7	1984 gain
1990 [OP] 14	6.521819788	
10 [OP] 15	1988.297787	

In 1990 the company can expect $6.5 million profit and to reach the $10 million mark in 1998.

Statistics in Calculations

Statistical operations can be performed during complex calculations. For instance, an insurance company may compute its overhead by the formula $3 + 2 \times 1.2^{(4-N)}$ where N is the number of people needed for $200,000 sales. Simply key in (3 + 2 × 1.2^(4 − and following the example that found insurance sales, calculate x' for y = 200 using the statistics keys. After projecting the number of people for $200,000 sales, press [ )] [ )] to complete the calculation. Notice that four operations are pending while the statistics calculation is in progress.

General Programming

This section presents the details of working with programs and programming keys. For a guide to programming, refer to Chapter 3

Programming Your Calculator

To solve a problem from the keyboard, you determine a sequence of operations and functions needed to give you the solution to that problem and key your solution into the calculator. Basic programming is little more than entering the learn mode and telling the calculator to remember a specific key sequence. The keystrokes are stored in locations in program memory and each becomes a program instruction. The series of keystrokes (instructions) is a program. When the instructions of the program are executed (run) they produce the same result that the equivalent manual keystrokes would have yielded. Once stored, this program can be executed again and again by supplying new sets of variables instead of entering all the program keystrokes. This not only saves you input time, but decreases the chances of making an entry error.

The program stays in program memory until it is replaced by another program or cleared by pressing [2nd] [CP]. Meanwhile, the program can be used whenever you need it. For example, while performing a series of manual operations, you may find you need an answer from a stored program. Simply call and execute the program, then return to your calculations with the program results.

Storage Capacity and Partitioning

The memory area within your calculator is for data storage and program storage. When the calculator is off, this memory area is preserved by the constant memory.

If you have a program that goes past step 255 and the calculator is turned off, you will need to partition for fewer than 32 data memories before running the program again; otherwise, the last part of the program will be missing. The constant memory stores the instructions in steps 256-263 as values in data memory 31, steps 264-272 in memory 30, and so on up to steps 504-511 in memory 00. When enough of the memory area is converted to program steps, the program will be restored. If you alter a data memory that is storing program steps, the instructions will be different when that memory is converted back to program steps.

See the section covering Selection of Memory Size for the details of partitioning.

Fundamental Program Control Functions

Understanding several fundamental control functions will allow you to begin programming your calculator.

[LRN], Learn: Pressing this key once puts the calculator in the learn mode of operation. This allows you to begin writing a program into program memory which can be run later. Pressing [LRN] again puts the calculator back under keyboard control and restores the display to its original state. Use of the learn key always preserves the value in the display. You cannot enter the learn mode if partitioning is for zero program steps.

[2nd] [CP], Clear Program: When pressed from the keyboard, [2nd] [CP] clears all locations of program memory, clears the subroutine-return register, resets all user flags, clears the t register and resets the program pointer to ST. When encountered within a program, it only zeros the t register.

[R/S], Run/Stop: Reverses the status of processing. Pressing [R/S] starts program processing at the current position of the program pointer. Pressing [R/S] while a program is running stops the program. The exact stopping position of the program pointer, however, cannot be predetermined. Entering [R/S] as a program instruction in the learn mode causes program processing to stop at that point when the program is running.

[RST], Reset: Instructs the calculator to reset the program pointer to ST, clears the subroutine return register and resets all program flags.

[2nd] [Pause], Pause: When encountered during program execution, causes the current value of the display register to be displayed for slightly over one second. Pause instructions can be used wherever needed, even consecutively. When held down on the keyboard during program execution, it displays the result of each program step.

Learn Mode

After a calculation sequence has been determined, select the learn mode by pressing [2nd] [CP] [LRN], then key the sequence into program memory. [2nd] [CP] assures that the program is keyed in beginning at location 000 and the program area is cleared. When you enter the learn mode, the display has two groups of characters. The left-hand group indicates the step number and is always three digits. The right-hand group indicates the instruction at that program location and can vary in the number of alphanumeric characters, depending on an instruction's mnemonic.

The display shows a mnemonic for each program instruction. A mnemonic is a three (or fewer) letter or alphanumeric representation of an instruction displayed in the special alpha positions. An instruction's mnemonic is similar to the instruction's name. For example, the mnemonic for pause is PAU and the mnemonic for run/stop is R/S. Appendix E provides a complete list of program mnemonics.

When a program is being keyed in, the instruction just entered is displayed. When an instruction uses a memory address, the two digits are included in the same step. When an instruction uses a program address, the first digit occupies a step and the other two occupy the next step.

Keeping track of exactly where you are in program memory is the function of the program pointer. In the learn mode, this indicator advances through program memory indicating the program step just used.

After keying a program into program memory, press [LRN] again to return the calculator to keyboard control. The variables can then be entered and program execution begun.

Entering Your Program

The sequence for keying your program into program memory is:

1. From the keyboard press [RST] or [2nd] [CP]. Either sequence positions the program pointer to ST, the first location of program memory. The [2nd] [CP] sequence also clears program memory.
2. Press[LRN]to place the calculator in the learn mode. The display takes on a unique format in this mode.
3. Key in your program one step at a time, including all necessary [2nd] and [INV] prefixes. The display shows the step you just keyed in, which was automatically inserted. All later program steps are shifted one step due to the automatic insert action. The step at the end of program memory is discarded.
4. Make sure your program does not exceed program memory size. When the last partitioned location is filled, the calculator automatically switches to keyboard control where the learn mode display is conspicuously absent.
5. Switch from the learn mode to keyboard control by pressing [LRN] again.
6. Run a test problem with known results to be sure the program is correct and edit your program if necessary.

The following example illustrates the previous comments.

Example: Create a program to calculate the volume of a right circular cylinder of radius, r, and height, h.

Necessary equation: $V = \pi r^2 h$

Desired Program Operation: Enter r
Start Program
halt for h entry
Calculate volume, halt
and display the answer

Key Sequence	Display	Comments
[2nd] [CP]		Sets program pointer to location ST and erases program memory
[LRN]	ST	Places calculator in learn mode
[x^2]	000 X^2	r^2 (r will be entered before program execution)
[×]	001 X	
[2nd] [π]	002 PI	Occupies 1 keystroke
[×]	003 X	πr^2 in display register while multiplication is pending
[R/S]	004 R/S	Halts for "h" entry
[=]	005 =	Calculates result
[R/S]	006 R/S	Stops program, "V" displayed
[RST]	007 RST	Return to instruction at 000
[LRN]		Returns to keyboard control

The last two keystrokes entered have specific functions. The [R/S] key halts processing and displays the final answer. The [RST] key provides a natural return to location 000 when "r" is entered for a new set of variables and [R/S] is pressed.

Running Your Program

When a program is run, the instructions are executed in sequential order beginning at the current location of the program pointer (unless a transfer takes place—these are discussed later). To initiate this process, press the [R/S] (Run/Stop) key. The program pointer keeps up with exactly where processing is in the program. If the calculator attempts to execute past the program boundary, processing halts and Error is displayed. Therefore, programs should end with a R/S (or with a RTN or a transfer, both discussed later).

With the volume problem in program memory, calculate the volume for r = 3 and h = 9.

Press	Display	Comments
[RST]		Positions program pointer to ST
3	3	Enter r
[R/S]	(blank)	Begins program execution
	28.27433388	πr^2 – value in display register when [R/S] encountered, halting program
9	9	Enter h
[R/S]	254.4690049	Program halts, displays V.

Note that the display is blank while a program is being executed.

When a sequence is executed, the program pointer controls the flow of processing by pointing to each instruction in turn. As additional programming capabilities are introduced, the role of the program location pointer will be expanded.

Working With Programs

[SST], Single Step: Increments the program pointer by one. In the learn mode, pressing this key causes the next step to be displayed. Pressing this key from the keyboard causes the program to be executed one step at a time, with the result of each step being displayed.

[BST], Backstep: In the learn mode, decrements the program pointer by one and shows that step's instruction. This key is inoperative from the keyboard and when a program is running.

The step keys provide free movement in program memory to permit you to efficiently check out and "debug" your programs. As you single-step and back-step through a program while in the learn mode, you see a mnemonic stored at each location to represent the various instructions.

The single-step instruction is usable from the keyboard as well as in the learn mode. When pressed while under keyboard control, [SST] causes actual execution of the stored program, one instruction at a time. Each time [SST] is pressed, the instruction located at the current position of the program pointer is executed and the program location pointer is advanced just as if the program had been running. The display shows the calculated values resulting from that instruction. Sometimes several single-step keystrokes are necessary before anything appears to happen, but this is only because a multistep operation is in progress. For example, the sequence + RCL 09 would take three single-step keystrokes before the recall of the memory register 09 content would actually take place.

Keystroke Storage

An instruction occupies one location in program memory even though some instructions depend on their adjacent instructions for their meaning. Often, instructions are formed by pressing a sequence of keys that combine to occupy a single location. The [2nd] key is combined with the instruction following it to occupy one location. The two-digit addresses accompanying instructions for data memory and special control operations are combined to occupy one location. For instance, [RCL] 16 occupies only two locations as does [2nd] [StF] 1 with [2nd] [StF] placed in one location and the flag number 1 in the next. The calculator automatically does this for you.

Unconditional transfer addresses such as [GTO] 123 are stored with [GTO] in one location, 01 in the next and 23 in the next. As you key in this sequence, the calculator assimilates the keystrokes, and places them in the correct locations. No special effort on your part is required.

For certain instructions, [Ind] combines with the instruction to occupy one location, and is assigned a merged mnemonic. The indirect instructions affected are:

Key Sequence	Mnemonic
[2nd] [Exc] [2nd] [Ind]	EX*
[2nd] [Prd] [2nd] [Ind]	PD*
[STO] [2nd] [Ind]	ST*
[RCL] [2nd] [Ind]	RC*
[SUM] [2nd] [Ind]	SM*
[GTO] [2nd] [Ind]	GO*
[OP] [2nd] [Ind]	OP*

Indirect instructions that are not merged are represented by two mnemonics. For instance, SBR appears in one location and IND in the next. Uses of indirect sequences are explained in the next section.

Editing Programs

Be very careful of what you edit because of merging. Consider the following sequence.

```
.
.
.
019 =
020 STO
021 12
022 GTO
.
.
.
```

If the STO is deleted, the memory address 12 is treated as a NOP. If you need STO 13 instead of STO 12, positioning the pointer to 020 and entering 13 places the 1 in 021 and the 3 in 022 not 13 in 021. To achieve this change you must delete locations 21 and 20 and enter [STO] 13 from location 019.

Labels

A label serves only as a recognizable point in a program. Execution can look for a label and go to it but there is no numerical meaning for a label.

All labels that are used in the current program memory can be listed on the PC-200 Printer along with the locations in program memory. Simply press [RST] (to position the program pointer to location ST), then press [OP] 08 and the table is printed out.

Transfer Instructions

Go To Instruction [GTO] L or nnn—Go To Instruction—When used in a program, Go To instantly diverts the flow of processing to label L or program location nnn. From the keyboard, Go To positions the program location pointer to the part of the program labeled L or to location nnn, but does not start program execution. The keyboard use of this key is useful for rapid access to any part of a program for editing or other purposes.

Transferring to a specific program location (nnn) is called absolute addressing.

A common application of GTO is in a program loop. You could count by fours from the keyboard by pressing [+] 4 [=] repeatedly but consider doing this with a program loop. This program can be done several ways.

000 LBL 001 SUM 002 + 003 4 004 = 005 PAU 006 GTO 007 SUM	000 + 001 4 002 = 003 PAU 004 GTO 005 00 006 00	000 LBL 001 C 002 + 003 4 004 = 005 PAU 006 GTO 007 C	000 + 001 4 002 = 003 PAU 004 RST
Common Label	Absolute Addressing	User-Defined Label	Reset

Pressing [GTO] 7 is the short-form address which automatically stores the sequence in the proper three locations GTO 00 07 as soon as a nonnumeric key is pressed. Remember, at least one digit must always be submitted or GTO will accept the next entry as a label.

Conditional Transfers (Test Instructions)

These instructions transfer program operation only if a specific condition is satisfied: the comparison of two numbers or the number of times a loop has been executed or the status of a flag.

T-Register Comparisons

[x⇆t]	Exchanges display register value x with t register value t.
[2nd] [x=t] L or nnn	Asks "Is the display register value exactly equal to the t register value?"
[INV] [2nd] [x=t] L or nnn	Asks "Is the display register value unequal to the t register value?"
[2nd] [x≥t] L or nnn	Asks "Is the display register value greater than or exactly equal to the t register value?"
[INV] [2nd] [x≥t] L or nnn	Asks "Is the display register value less than the t register value?"

When the answer is "yes" to any of the above questions the flow of processing branches to the address or label that immediately follows the instruction. If the answer is "no", processing skips the accompanying address and goes on to the next instruction.

These tests do not affect pending operations, hence they can be used wherever desired in a program.

Example:

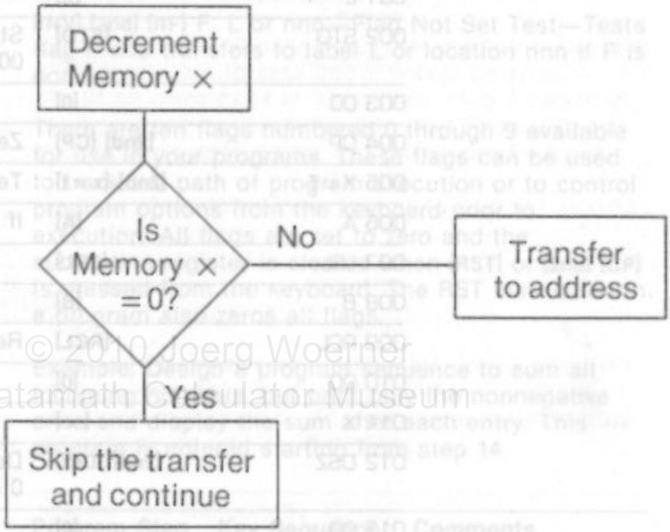
Program Step	Key Sequence
022 RCL	[RCL]
023 01	1
024 =	[=]
025 X≥T	[2nd] [x≥t]
026 B	[B]
027 GTO	[GTO]
028 00	3
029 03	
030 LBL	[LBL]
031 B	[B]
032 R/S	[R/S]

In this sequence, the result produced at location 024 is tested in location 025 to see if it is greater than or equal to the value stored in the t register. If the answer to the test is yes, the flow of processing jumps to label B where processing stops and the result is displayed. If the answer is no, the transfer to label B is skipped and GTO 003 transfers to location 003 where processing continues.

Decrement And Skip On Zero (DSZ)

[2nd] [Dsz] X, L or nnn—Decrement and Skip on Zero—Decreases the magnitude of the contents of memory X (0-9) by one and transfers processing to label L or location nnn when memory X contains a nonzero value. The transfer is skipped when zero is in memory X.

This powerful programming instruction is an effective counter as well as a test instruction. If you need to repeat a sequence y number of times, just store y in a user data memory (0-9) before execution reaches the sequence and include a DSZ at the end of the sequence. After y iterations the looping ceases and the program continues. The DSZ instruction operates as follows:



If memory X is not 0, transfer to L or nnn. If memory X is 0, skip the transfer and continue processing.

This illustration shows that if you place the DSZ instruction at the beginning of a sequence, it counts first and then performs the calculation sequence. With DSZ at the end of the sequence, the function is performed first and then the count is made. This means that to obtain the correct number of passes, y, through a sequence, either enter y into user data memory X initially and perform DSZ at the end of the sequence or perform INV DSZ at the beginning of the sequence.

Example: Write a program to calculate X! (X factorial) where X! = 1 × 2 × 3 × ... × X. (0! = 1 by definition)

Program Step	Key Sequence	Comments
000 LBL	[LBL]	
001 E	[E]	
002 STO	[STO]	Store X in register 00
003 00	[0]	
004 CP	[2nd] [CP]	Zero t register
005 X=T	[2nd] [x=t]	Test F=0?
006 A	[A]	If yes, transfer to A
007 LBL	[LBL]	
008 B	[B]	
009 RCL	[RCL]	Recall X
010 00	[0]	
011 X	[×]	
012 DSZ	[2nd] [Dsz]	Decrement memory 0 by 1
013 00	[0]	
014 B	[B]	If memory 0 is not 0, transfer to B
015 LBL	[LBL]	If memory 0=0, proceed to the end
016 A	[A]	
017 1	[1]	
018 =	[=]	
019 R/S	[R/S]	Stops and displays X!

To execute this sequence, simply enter a number and press [E]. X must be less than 70 or the calculator overflows because $70! > 9.9999999 \times 10^{99}$.

Flags

[2nd] [StF] F—Set Flag—Sets or turns on flag F where 0 ≤ F ≤ 9. [INV] [2nd] [StF] F resets or zeros flag F.

[2nd] [IfF] F, L or nnn—Flag Set Test—Tests flag F to see if it is set. If so, transfer is made to label L or location nnn.

[INV] [2nd] [IfF] F, L or nnn—Flag Not Set Test—Tests flag F and transfers to label L or location nnn if F is not set.

There are ten flags numbered 0 through 9 available for use in your programs. These flags can be used to track the path of program execution or to control program options from the keyboard prior to execution. All flags are set to zero and the subroutine register is cleared when [RST] or [2nd] [CP] is pressed from the keyboard. The RST instruction in a program also zeros all flags.

Example: Design a program sequence to sum all incoming numbers, but print only the nonnegative ones and display the sum after each entry. This program is entered starting from step 14.

Program Step	Key Sequence	Comments
		Press [2nd] [CP] when outside learn mode to remove the previous program to prevent duplication of label A.
015 LBL	[LBL]	
016 A	[A]	
017 INV	[INV]	
018 ST.F	[StF]	
019 03	3	

020 X≥T	[2nd] [x≥t]	"Is number nonnegative?"
021 B	[B]	If so, go to label B
022 ST.F	[2nd] [StF]	If not, set flag 3
023 03	[3]	
024 LBL	[LBL]	
025 B	[B]	
026 SUM	[SUM]	Sum all numbers
027 12	[1] [2]	
028 IF.F	[2nd] [IfF]	"Is flag 3 set?"
029 03	[3]	
030 C	[C]	If so (if number is negative), go to C
031 PRT	[2nd] [Prt]	If not, print number
032 LBL	[LBL]	
033 C	[C]	
034 INV	[INV]	
035 ST.F	[2nd] [StF]	Reset flag 3 for next entry
036 03	[3]	
037 RCL	[RCL]	
038 12	[1] [2]	
039 R/S	[R/S]	

Be sure that user data memory 12 is clear, the t register = 0, and flag 3 is reset before entering a series of numbers. Entering a number and pressing [A] sums that number into user data memory 12, prints the entry if it is positive, and displays the total of all entries.

Flags And Error Conditions

Flag 8 is designated to determine program operation according to the error condition status of a program. Normally, a program continues running even when an arithmetic error occurs. If flag 8 is set either from the keyboard or in a program, program execution is suspended when such an error condition occurs. If flag 8 is reset, only non-arithmetic errors halt the program.

[OP] 18 sets flag 7 only if no error condition exists in a program. [OP] 19 sets flag 7 only if an error condition does exist in a program. Flag 7 can then be monitored to determine the error status of your program and appropriate responses can be made.

If either of these tests is false, flag 7 is not altered.

Indirect Addressing

[2nd] [Ind] XX—Indirect Suffix—When used after one of the following operations, recalls the contents of user data memory XX and uses these as the actual information to use in processing.

Indirect Key Sequences

Mnemonics	Purpose
[STO] [Ind] XX ST* XX	Indirect store
[RCL] [2nd] [Ind] XX RC* XX	Indirect recall
[2nd] [EXC] [2nd] [Ind] XX EX* XX	Indirect exchange
([INV]) [SUM] [2nd] [Ind] XX (INV) SM* XX	Indirect sum to (subtract from) memory
([INV]) [2nd] [Prd] [2nd] [Ind] XX (INV) PD* XX	Indirect multiply by (divide into) memory
[GTO] [2nd] [Ind] XX GO* XX	Indirect go to
[OP] [2nd] [Ind] XX OP* XX	Indirect special control
[SBR] [2nd] [Ind] XX SBR IND XX	Indirect subroutine

[2nd] [Fix] [2nd] [Ind] XX FIX IND XX	Indirect fix-decimal
([INV]) [2nd] [x=t] [2nd] [Ind] XX (INV) X=T IND XX	Indirect x=t (x≠t) test
([INV]) [2nd] [x≥t][2nd] [Ind] XX (INV) X≥T IND XX	Indirect x ≥ t (x<t) test
([INV]) [2nd] [StF] [2nd] [Ind] XX (INV) ST.F IND XX	Indirect set (reset) flag
([INV]) [2nd] [IfF] [2nd] [Ind] XX L or nnn (INV) IF.F IND XX or nnn	Indirect flag number set (reset) test
([INV]) [2nd] [IfF] y [2nd] [Ind] XX (INV) IF.F Y IND XX	Indirect address, flag set (reset) test
([INV]) [2nd] [IfF] [2nd] [Ind] YY [2nd] [Ind] XX (INV) IF.F IND YY IND XX	Indirect address and flag number, set (reset)
([INV]) [2nd] [Dsz] [2nd] [Ind] XX L or nnn (INV) DSZ IND XX or nnn	Indirect DSZ register skip on zero (non-zero) test
([INV]) [2nd] [Dsz] X [2nd] [Ind] XX (INV) DSZ X IND XX	Indirect address, skip on zero (nonzero) test
([INV]) [2nd] [Dsz] [2nd] [Ind] YY[2nd] [Ind] XX (INV) DSZ IND YY IND XX	Indirect DSZ memory, skip on zero (non-zero) test, indirect destination
[2nd] [Part] [2nd] [Ind] XX PAR IND XX	Indirect partition

The indirect instruction indirectly locates the memory address by finding it in user data memory XX. The effect of pressing [RCL] [2nd] [Ind] 04 would not be to recall the contents of register 04, but to use memory 04 as the requested memory address. If 27 is stored in register 04 then 27 becomes the memory address. The indirect address "points" to the actual address.

Any pointer used in conjunction with an indirect instruction must point to an existing address. If the value in the indirect register is less than zero, the calculator uses register 00. If the value is beyond the partition, processing halts and Error is displayed.

Chapter 5—PRINTER CONTROL

Details on how to use the printer functions are covered in this chapter, but general operating and service information for the printer is included with the printer.

The optional PC-200 thermal printer can be used with your TI-66 to perform a number of different printing tasks. With the printer connected you can:

- Print the contents of the display.
- Print alphanumeric messages.
- Make a plot of data from the keyboard or automatically from a program.
- List the program in program memory.
- List the contents of all user data memories.
- List all program labels and the program location of each.
- Print results from any point of a program.
- Print each step of calculator operation by tracing calculations made from the keyboard or in a program.

The calculator contains a set of symbols that are referred to as audit trail symbols. Originally, the term "audit trail" pertained to the record of financial calculations kept by a business that are examined when the business audits its books; the audit trail was a column to the side of the numbers that beside each number showed how the number was used by the calculation. This term has carried over to printed calculations in general and now refers to the symbols printed by the arguments of each calculation to identify the operation that has been performed. In the trace mode, the calculator prints an audit trail symbol to identify the calculation just performed. A list of the audit trail symbols is included later in this chapter.

The calculator will perform functions intended for the printer whether or not the printer is connected. When you use a printer function, the display goes blank until the calculator has completed the operation.

Selective Printing

When [2nd] [Prt] is pressed from the keyboard or encountered in a program, the display value is printed.

Consider the following program that prints multiples of 4:

```
000 +
001 4
002 =
003 PRT
004 RST
```


Enter zero as the starting point, press [RST] [R/S], and the following is obtained.

Results

4.
8.
12.
16.
.
.
.

Press [R/S] to stop the program. Press [CLR] to clear possible pending operations.

Printed results can be separated by using the paper advance key [2nd] [Adv] on the calculator. When used from the keyboard, it advances the paper until you release the key. When used in a program, it advances the paper one line. To separate the multiples of 4, insert ADV into the location after PRT and you get the following.

Results

4.

8.

12.

16.

.

.

.

Press [R/S] to stop the program. Press [CLR] to clear possible pending operations.

To obtain additional space, use additional ADV instructions.

Listing Your Program

Note: All listing activities can be performed only from the keyboard.

To list a program, press [2nd] [List]. The program is then listed from the current position of the program pointer to the end of program memory. You can manually stop the listing at any time by pressing [R/S]. For a complete program listing, press [RST] [2nd] [List] when outside of the learn mode. The multiples-of-4 program lists like this:

```
ST
000 +
001 4
002 =
003 PRT
004 ADV
005 RST
006 0
007 0
```


Press [R/S] to stop the listing.

Listing User Data Memories

The sequence [INV] [2nd] [List] lists the contents of all user data memories beginning with the memory number shown in the display. The listing continues until the contents of the highest numbered data memory are listed or until you press [R/S] from the keyboard. Then the calculator is back under keyboard control. A listing of the contents of memory 27 up to the partition at 31 is shown below.

The numbers listed depend on the numbers stored in memories 27 through 31

Memory Contents	Memory Number
14.18181818	27
−5.54881-12	28
665.8568182	29
110.9761364	30
0.	31

This listing was obtained by pressing 27 [INV] [2nd] [List].

Tracing Your Calculations

By pressing [2nd] [Trace] or setting flag 9, you cause every step of a calculation to be printed. The calculated value and the instruction that created it are displayed. This is true for both keyboard calculations and program calculations.

The [2nd] [Trace] key causes trace mode operation for all calculations. In this mode every new function or result is automatically printed. A number entry is only printed if followed by an operation or function. Operation in the trace mode continues until the [2nd] [Trace] key is pressed again or until flag 9 is reset. When an error condition occurs, Error is printed.

With the multiples-of-4 program still in program memory, press [2nd] [Trace]. Now, press [CLR] [RST] [R/S] to obtain the following trace of the calculations that take place.

Display Register	Audit Symbols
	CLR
	RST
0.	+
4.	=
4.	
4.	PRINT
	RST
4.	+
4.	=
8.	
8.	PRINT
	RST
8.	+
4.	=
12.	

Press [R/S] [CLR] to stop the program and [2nd] [Trace] to leave the trace mode.

Audit Trail Symbols In Trace Mode

For some operations, the audit trail symbol is the same as the mnemonic in a program listing while for others, the audit trail symbol is unique to the trace mode. A complete list of all audit symbols and the key sequence for each one follows.

Printer Listing	Key Sequence
*	[2nd] [Ind] (Suffix to indirect functions)†
√ X	[√x]
1/X	[1/x]
A-E	[A]-[E]
A'-E'	[2nd] [A']-[2nd] [E']
ADV	[2nd] [Adv]

CE	[CE]
CLR	[CLR]
CMS	[2nd] [CMs]
COS	[2nd] [cos]
CP	[2nd] [CP]
CSR	[2nd] [CSR]
D-DMS	[INV] [2nd] [DMS-DD]†
DEG	[2nd] [Deg]
DMS-D	[2nd] [DMS-DD]
DSZ	[2nd] [Dsz]
EE	[EE]
ERROR	(Error condition)†
EXC	[2nd] [Exc]
EXC*	[2nd] [Exc] [2nd] [Ind]
FIX	[2nd] [Fix]
GRAD	[2nd] [Grad]
GTO	[GTO]
GTO*	[GTO] [2nd] [Ind]
ICOS	[INV] [2nd] [cos]†
IDSZ	[INV] [2nd] [Dsz]†
IΣ+	[INV] [2nd] [Σ+]†
IFF	[2nd] [If F]
IFIX	[INV] [2nd] [Fix]†
IIFF	[INV] [2nd] [If F]†
IINTG	[INV] [2nd] [Intg]†
ILOG	[INV] [2nd] [log]†
ILnX	[INV] [lnx]†
INTG	[2nd] [Intg]
IP->R	[INV] [2nd] [P►R]†
IPROD	[INV] [2nd] [Prd]†
ISBR	[INV] [SBR] (keyboard only)†
ISIN	[INV] [2nd] [sin]†
ISTF	[INV] [2nd] [StF]†
ISUM	[INV] [SUM]†
ITAN	[INV] [2nd] [tan]†
IX=T	[INV] [2nd] [x=t]†
IXI	[2nd] [|x|]
IX≥T	[INV] [2nd] [x≥t]†
IY^x	[INV] [y^x]†
I $\bar{x}$	[INV] [2nd] [$\overline{x}$]†
LOG	[2nd] [log]
LnX	[lnx]

†Printed in trace mode only.

OP	[OP]
OP*	[OP] [2nd] [Ind]
P->R	[2nd] [P►R]
PART	[2nd] [Part]
PAUSE	[2nd] [Pause]
PRINT	[2nd] [Prt]
PROD	[2nd] [Prd]
PROD*	[2nd] [Prd] [2nd] [Ind]
R/S	[R/S]
RAD	[2nd] [Rad]
RCL	[RCL]
RCL*	[RCL] [2nd] [Ind]
RETN	[INV] [SBR] (from a program only)
RST	[RST]
SBR	[SBR]
SIN	[2nd] [sin]
STF	[2nd] [StF]
STO	[STO]
STO*	[STO] [2nd] [Ind]
SUM	[SUM]
SUM*	[SUM] [2nd] [Ind]
TAN	[2nd] [tan]
TRACE	[2nd] [Trace]
X^2	[x^2]
X=T	[2nd] [x=t]
X≥T	[2nd] [x≥t]
X:T	[x⇆t]
Y^x	[y^x]
x̄	[2nd] [x̄]

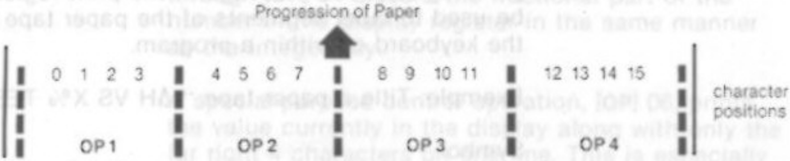
SYMBOLS	
Σ+	[2nd] [Σ+]
π	[2nd] [π]
(	[( ]
)	[ )]
÷	[÷]
×	[×]
−	[−]
+	[+]
=	[=]

Special Control Operations For Printing

Special control operations (OP codes) 00 through 08 are specifically designed for use with the printer.

Alphanumeric Printing, [OP] 00-06

The first seven OP codes allow you to create and print out alphanumeric messages. Sixteen characters can be printed on each line. They are assembled and stored in groups of 4 characters at a time as shown below.



Each printed character is represented by a two-digit, row-column address code according to the following table:

TENS DIGIT \ UNITS DIGIT	0	1	2	3	4	5	6	7
0	blank	A	B	C	D	E	F	G
1	H	I	J	K	L	M	N	O
2	P	Q	R	S	T	U	V	W
3	X	Y	Z	n	√	×	÷	=
4	0	1	2	3	4	5	6	7
5	8	9	−	+	'	.	?	blank
6	Σ	(	)	≠	/	!	e	π
7	^{x}	<	>	≥	%	^{2}	Δ	·
8	↑	$\bar{x}$	≤	‖	"	★	°	⇄

For instance, A is code 01 and + is code 53. The codes for four characters (eight digits) are used by OP codes 01 through 04. If you do not specify all eight digits, zeros are assumed to precede the digits entered (each zero pair represents a blank space). If you specify more than eight digits, only the eight digits directly to the left of the decimal are used by the calculator. To obtain spaces after characters, enter pairs of zeros after the codes of the characters.

After the display contains a series of character codes, OP codes 01, 02, 03, or 04 tell the calculator the quarter of the line on which these characters are to be printed.

[OP] 01—far left quarter of line
[OP] 02—inside left quarter of line
[OP] 03—inside right quarter of line
[OP] 04—far right quarter of line

Pressing [OP] 00 clears the print register. [OP] 05 instructs the calculator to print the contents of the print register. The message in the print register can be used to label segments of the paper tape from the keyboard or within a program.

Example: Title a paper tape "ΔH VS X% TESTS "

Symbol	Δ	H			V	S		X	%		T	E	S	T	S	
Code	00	76	10	00	26	23	00	30	74	00	24	05	23	24	23	00

Press	Display	Comments
[CLR] [OP] 00	0	Clear display and print register
00761000 [OP] 01	761000	Store " ΔH " printing on far left quarter
26230030 [OP] 02	26230030	Store "VS X" for printing on inside left quarter
74002405 [OP] 03	74002405	Store "% TE" for printing on inside right quarter
23242300 [OP] 04	23242300	Store "STS " for printing on far right quarter
[OP] 05	23242300	Prints complete title on printer

Note that a blank is the first thing needed for the far left quarter. Leading zeros in the print register produce this blank. On the far left 00 is the first character code needed but it need not be entered. By entering only six digits, you can imply that the first two are 00. The quarters can be loaded in any order and can be written over by another set of codes.

Always remove fix-decimal, scientific notation, and engineering display formats before entering alphanumeric messages. Also note that [OP] 01 through [OP] 04 discard the fractional part of the number in the display register in the same manner as the integer key.

A special-purpose control operation, [OP] 06, prints the value currently in the display along with only the far right 4 characters on one line. This is especially useful to label program results.

Example: Design a program to calculate e and label the result.

000 6	
001 1	
002 6	
003 6	
004 6	
005 2	
006 OP	Store (e) for printing
007 04	
008 1	
009 INV	
010 LNX	
011 =	
012 OP	Print
013 06	
014 R/S	

When this program is executed, the printer will produce 2.7182818 (e).

Plotting Data, [OP] 07

Special control operation 07 plots an * for current display value (0-15) in character position 0-15 with 0 being on the left. Primarily designed for use in a program, this operation allows you to plot curves or histograms. Only one * is plotted per line and the value x to be plotted must be $-1 < x < 16$ (but normally $0 \leqslant \text{INT}(x) \leqslant 15$ is used). If the displayed value is not within this range, the value is not plotted and Error is displayed when the program halts. For each point, OP 07 discards the fractional portion of the number in the display and uses the remaining integer to determine the position of the *.

Example: Design a program to plot a sine curve sampled every 18 degrees.

Key Sequence	Results
[LBL]	0 3 6 9 12 15 Position
[A]	
[RCL]	
[1]	
[2nd] [sin]	
[×]	
[7]	
[+]	
[7]	
[.]	
[5]	
[=]	
[OP]	
[0] [7]	
[1]	
[8]	
[SUM]	
[1]	
[GTO] [A]	

Notice that the values are scaled up by 7, making the range −7 to 7. The result is made positive by adding 7.5 to all values. The values now range from .5 to 14.5. The plot varies symmetrically about position 7 due to the integer portion being plotted. The program is executed by storing a starting angle in register 01 then pressing [A]. The program continues until [R/S] is pressed.

List Program Labels Used, [OP] 08

To obtain a sequential listing of all labels and the locations at which they occur in program memory, press [OP] 08. This listing begins at the current position of the program pointer, so to list all the labels, press [GTO] 0 or [RST] before [OP] 08. See the sample listing below.

Program Location	Audit Symbols
001	A
018	B
062	D
129	C
205	DMS·D
239	RAD

The listing indicates the labels used and where they are found in program memory. Depending on labels in your program, this listing will vary.