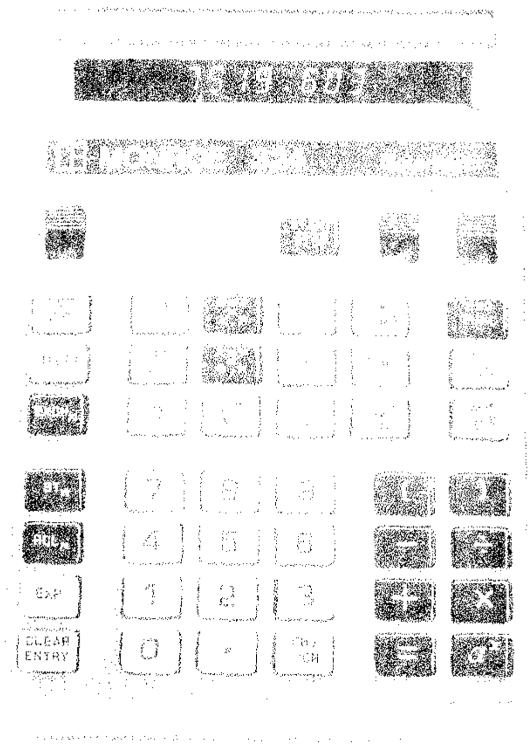
Litton MONROE

HOW TO USE

The Monroe 324 Micro Scientist

Monroe, The Calculator Company



INTRODUCTION

We'd like to extend our congratulations on your discriminating taste. In buying a Monroe Model 324 Micro Scientist, you have purchased the end result of 60-plus years of calculator experience. The Micro Scientist is designed to take the drudgery out of complicated mathematical calculations. It is much easier and faster to use than your slide rule and more accurate than the tables you've been using.

Although the Micro is simple to operate, it has many sophisticated features. This manual is written to help you get the most out of those features. Read it through and follow the examples with your Micro Scientist. Don't be afraid to experiment. Once you have studied the manual, keep it handy for quick reference.

The Micro Scientist is such a powerful tool that you'll soon wonder how you ever did without it. In a short time, it will be as indispensable as pencil and paper.

CONTENTS

HOW TO START

Turn on the Micro Scientist with the on/off switch on the back. When you turn it on all the registers are cleared to zero and the decimal point is set to two places.

Press RESET *and see* [0.00]

The Micro Scientist runs on rechargeable batteries or on house current if the AC adapter is used. When using batteries, the Micro Scientist's display begins flashing to warn you when the batteries are almost discharged. Batteries will recharge overnight with the AC adapter plugged in. See page 61 for more information on batteries and recharging.

The RUN/LOAD switch should be up (RUN) for normal keyboard operation. Set the GRAD/DEG switch to DEG unless you work with grads. When you're doing circular trigonometry, the Micro Scientist assumes either 360-degree circles or 400-grad circles depending on the setting of the switch.

CORRECTING MISTAKES

Whenever you find you've made a mistake in an operation, RESET clears the calculation so you can start it over. CLEAR ENTRY merely clears the display so that you can correct an error in entering a number without losing the calculation you were doing.

Although it's usually not necessary, it's a good idea to press RESET before starting a calculation, just to be sure there are no leftovers inside.

If you see [E - - - -] in the display (it stands for ERROR), that means you've done something mathematically illegal or have produced a number outside the Scientist's range (its capacity is 10^{-99} to 10^{+99}). Calculation is stopped. Press RESET or CLEAR ENTRY to get going again.

A complete list of error conditions is shown on page 60.

ENTERING NUMBERS

You enter numbers simply by pressing each digit of the number that you want to enter. Here are some examples:

Enter This	*See This*
1 2 3 4 5 6 7 8	12345678.00

Notice that the decimal point setting of two is maintained for up to eight digits of entry.

The display holds a maximum of 10 digits. The decimal point setting moves to protect your most significant digits. CLEAR ENTRY

Enter This	*See This*
1 2 3 4 5 6 7 8 9	123456789.0
0	1234567890.

You can enter a full 13 digits at any time. The Micro Scientist stores and uses 13 digits even though a maximum of 10 are displayed.

Enter This	*See This*
9 8 7	1.234567890 12

The last three digits are not displayed. The two digits at the far right indicate the power of ten for scientific notation.

Enter This	*See This*
6	E - - - -

Entering more than 13 digits results in error. Press RESET or CLEAR ENTRY. Two other important numeral entry keys are • and CHG SIGN. Use • to enter the decimal portion of a number and CHG SIGN to indicate a negative value or to change a negative value to a positive value. CHG SIGN may be used before, after or during numeral entry.

To enter −326.1423

Enter This	*See This*
CHG SIGN 3 2 6 • 1 4 2 3	-326.1423

To enter a number in scientific notation, use EXP .

To enter 6.025 X 10^{23}, CLEAR ENTRY

	See This
6 • 0 2 5 EXP 2 3	6.025 23

ENTERING NUMBERS

To enter 662.52 X 10^{-36}, CLEAR ENTRY

Enter This	*See This*
6 6 2 . 5 2 EXP CHG SIGN 3 6	662.52 -36

RULES FOR NUMERAL ENTRY

- Enter numbers by pressing one digit at a time.
- Up to 13 digits can be entered, but only a maximum of ten can be displayed. Entering more than 13 digits gives E----.
- Use . to enter the decimal point. Only one decimal point can be in the display.
- Use CHG SIGN to make a positive number negative or a negative number positive. CHG SIGN may be used before, after or during numeral entry.
- Use EXP to enter numbers in scientific notation.
  - Enter the mantissa
  - Press EXP
  - Enter the power of ten in scientific notation

SET DP allows you to set the decimal point to give up to nine decimal places in the fractional portion of a number. The decimal point setting only controls how answers are displayed, not how numbers are entered. Demonstrate this to yourself by entering different settings. CLEAR ENTRY

Enter This	*See This*
SET DP 7	0.0000000
SET DP 4	0.0000

Unless otherwise stated, the decimal point is set to four places for the examples in this manual.

ARITHMETIC

Arithmetic calculations with the Micro Scientist are as simple as two plus two.

To find 2 + 2

Enter This	*See This*
2 + 2 =	4.0000

To find 10 - 3

Enter This	*See This*
1 0 - 3 =	7.0000

To find 7 X 6.3

Enter This	*See This*
7 X 6 . 3 =	44.1000

To find 17 ÷ 4

Enter This	*See This*
1 7 ÷ 4 =	4.2500

CHAIN ARITHMETIC

All arithmetic must end in =. Operations may be chained without the intermediate use of =. Here is an example of chain arithmetic.

To find $\frac{2 \times 3 - 4}{5} + 6$

Enter This	*See This*
2 X 3 - 4 ÷ 5 + 6 =	6.4000

Notice that whenever you press one of the arithmetic operators, the answer to the previous calculation is displayed. These keys act as intermediate = keys.

[a^x]

[a^x] is one of the most powerful keys on the Micro Scientist. It raises a number (a) to power(x). To use it, enter(a), press [a^x], enter(x), then press [=].

To find 4^3

Enter This	*See This*
[4] [a^x] [3] [=]	64.0000

To find $4^{3.1}$

Enter This	*See This*
[4] [a^x] [3] [.] [1] [=]	73.5166

Rules for a^x

- a may be a number or a combination of operations
- x may be a number or a combination of operations
- a may be any positive quantity
- x may be any positive or negative quantity
- If a is negative, x may be negative or positive, but it must be an integer
- $a^1 = a$; $a^0 = 1$; $1^x = 1$; $0^x = 0$; $-a^x$ = error if x isn't an integer; 0^{-x} = error

DOUBLING

[+] [=] used in combination doubles the number in the display.

To double the number 3

Enter This	*See This*
[3] [+] [=]	6.0000
[+] [=]	12.0000

SQUARING

[×] [=] used in combination will square the number in the display.

To square 9

Enter This	*See This*
[9] [×] [=]	81.0000

To square 17.6

Enter This	*See This*
[1] [7] [.] [6] [×] [=]	309.7600

REPEATED [+], [×] AND [=]

Pressing [+] several times has the effect of cumulatively doubling the number in the display.

Repeated [+] *with* [3]

Enter This	*See This*
[3] [+]	3.0000
[+]	6.0000
[+]	12.0000
[+]	24.0000

Pressing [×] several times results in the cumulative multiplication of the number in the display. RESET

Repeated × *with* [illegible] .

Enter This	*See This*
3 ×	3.0000
×	9.0000
×	81.0000
×	6561.0000

Pressing = several times after a calculation using + results in a cumulative addition. RESET

Repeated = with +

Enter This	*See This*
5 + 1 =	6.0000

The 5 is retained and added to each new number in the display.

=	11.0000
=	16.0000
=	21.0000

Pressing = several times after a calculation using × results in cumulative multiplication. RESET

Repeated = with ×

Enter This	*See This*
5 × 1 =	5.0000

The 5 is retained and multiplied by each new number in the display.

=	25.0000
=	125.0000
=	625.0000

PARENTHESES

( ) allow you to perform more complicated arithmetic operations. Use them just as you do when writing out an equation.

To find $(\frac{2+3}{7}) \times (\frac{4-6}{8})$

Enter This	*See This*
(2 + 3 ÷ 7	
) × (4 − 6	
÷ 8) =	-0.1785

Parentheses may be nested two deep. This lets you solve equations like the following.

To find $\frac{2}{5} \times [(2/3) + (3/4)]$

Enter This	*See This*
2 ÷ 5 × ((	
2 ÷ 3) + (	
3 ÷ 4)) =	0.5666

When using parentheses, keep these points in mind:

- For every open parenthesis, there must be a closed parenthesis. They are used in pairs.
- If you nest parentheses (use one pair of parentheses within another pair), make sure that for each left parenthesis there is a closing right parenthesis before you start another pair of nested parentheses.
- Closing parentheses act like [=] for the arithmetic after the previous opening parentheses.

CONSTANT ARITHMETIC

Constant arithmetic lets you repeatedly use a constant and operator without having to reenter them for each new calculation. Here are the rules:

- Enter the constant
- Press the operator ([+] , [−] , [×] , [÷] , or [a^x])
- Enter the variable
- Press [=]
- Continue to enter new variables, and press [=] for each new calculation.

Constant addition using 5

Enter This	*See This*
[5] [+] [2] [=]	7.0000
[4] [=]	9.0000
[3] [=]	8.0000

Constant subtraction using 100

Enter This	*See This*
[1] [0] [0] [−] [2] [0] [=]	80.0000
[1] [9] [=]	81.0000
[5] [.] [3] [1] [=]	94.6900

Constant multiplication using a multiplier of 3

Enter This	*See This*
[3] [×] [6] [=]	18.0000
[7] [=]	21.0000
[5] [.] [2] [=]	15.6000

Similarly the number in the display prior to the last [÷] is saved as a constant dividend.

Constant division using a dividend of 20

Enter This	*See This*
[2] [0] [÷] [2] [=]	10.0000
[3] [=]	6.6666
[2] [.] [7] [=]	7.4074

Raising the constant 2 to different powers

Enter This	*See This*
2 a^x 2 =	4.0000
3 =	8.0000
4 =	16.0000

SPECIAL FUNCTIONS

π

π puts the constant pi in the display, rounded to 13 places. SET DP 9 for this example.

Enter This	*See This*
π	3.141592653

To see the other 3 digits, subtract 3.14 from the value displayed

Enter This	*See This*
− 3 . 1 4	
=	590

3 digits not normally displayed

$\frac{1}{x}$

$\frac{1}{x}$ gives the reciprocal of the number in the display.

To find the reciprocal of 3, SET DP 4 *and*

Enter This	*See This*
3 $\frac{1}{x}$	0.3333

[π] [√] [2ND FUNC]

To find the reciprocal of π

Enter This	*See This*
[π] [1/x]	0.3183

The reciprocal of zero gives [E - - - -].

[√]

[√] gives the square root of the number in the display.

To find $\sqrt{81}$

Enter This	*See This*
8 1 [√]	9.0000

To find $\sqrt{125}$

Enter This	*See This*
1 2 5 [√]	11.1803

[√] of a negative number gives [E - - - -].

Remember, if you wish to square a number in the display, press [×] [=].

[2ND FUNC]

Many of the keys on the Micro Scientist do double duty—computing two things when you press them. The first labeled function appears in the display. The second labeled function is put in a separate register. To access the second function, press [2ND FUNC]. [2ND FUNC] exchanges the second function with the display. Repeated use of [2ND FUNC] continues exchange of the data.

[Ln LOG] [e^x 10^x]

[Ln LOG]

[Ln LOG] gives the natural logarithm of the number in the display and puts the common log (base 10) in [2ND FUNC].

To find ℓn 100

Enter This	*See This*
1 0 0 [Ln LOG]	4.6051
For log 100	
[2ND FUNC]	2.0000

If you press [2ND FUNC] again the natural log is displayed, exchanging the number in the display with the number in [2ND FUNC].

[Ln LOG] may be used over the full range of positive numbers 10^{-99} to 10^{+99}.

[Ln LOG] for numbers less than or equal to zero result in error.

[e^x 10^x]

[e^x 10^x] calculates anti log base$_e$ of the number in the display and stores anti log base 10 in [2ND FUNC].

To find the anti log of 3.219

Enter This	*See This*
3 . 2 1 9	
[e^x 10^x]	25.0031
[2ND FUNC]	1655.7699

To find the natural constant, e

Enter This	*See This*
1 [e^x 10^x] *or* [CLEAR ENTRY] [e^x 10^x] [e^x 10^x]	2.7182

e^x / 10^x SIN / COS SIN^{-1} / COS^{-1}

Using e^x/10^x with numbers outside the range -99 to +99 causes error since the antilog-base-10 exceeds the capacity of the Scientist. When this occurs, and if the number is between -225 and +229, the Micro Scientist puts the correct antilog-base-e into the second function register. Push CLEAR ENTRY 2ND FUNC to see it.

SIN / COS

SIN/COS gives the sine and cosine of an angle in the display. The angle may be of any size, up to $|\theta| \leq 9.0 \times 10^{14}$, and it can be positive or negative. It must be expressed in decimal degrees or grads depending on the position of the degree-grad switch.

To find the Sine and Cosine of 30°

Enter This		*See This*
Sine:	3 0 SIN/COS	0.5000
Cosine:	2ND FUNC	0.8660

SIN^{-1} / COS^{-1}

SIN^{-1}/COS^{-1} Calculates the Arc sine and Arc cosine of the number in the display.

To find the angle that has a sine of 0.85

Enter This		*See This*
Arc sine	. 8 5 SIN^{-1}/COS^{-1}	58.2116
Arc cosine	2ND FUNC	31.7883

SIN^{-1}/COS^{-1} outside the range −1 to +1 results in E. SIN^{-1} ranges from −90 to +90 degrees or −100 +100 grads. COS^{-1} ranges from 0 to 180 degrees or 0 to 200 grads.

TAN TAN^{-1}

TAN

TAN computes the tangent of the number in the display. The angle may be positive or negative and of any value except 90 degrees or 100 grads, or an odd multiple of 90 degrees or 100 grads. Tangents of these angles result in error.

To find the tangent of 38°

Enter This	*See This*
3 8 TAN	0.7812

TAN^{-1}

TAN^{-1} calculates Arc tangent of the number in the display.

To find the angle that has a tangent of 0.756

Enter This	*See This*
. 7 5 6 TAN^{-1}	37.0892

NOTE: The Arc tangent is limited to the range of -10^{49} to 10^{49}. The resulting angle is between -90 and +90 degrees (-100 to +100 grads). Arc tangents outside the range result in error.

θ→R/R→θ

Twelve trigonometric functions are readily accessible from your Micro Scientist. The following table summarizes the key sequences for each of them:

Function		*Key Sequence*
Sine	enter θ	SIN/COS
Cos	enter θ	SIN/COS 2ND FUNC
Tangent	enter θ	TAN
Cotangent	enter θ	TAN 1/x
Secant	enter θ	SIN/COS 2ND FUNC 1/x
Cosecant	enter θ	SIN/COS 1/x
Arc sine	enter sin θ	SIN^{-1}/COS^{-1}
Arc cosine	enter cos θ	SIN^{-1}/COS^{-1} 2ND FUNC
Arc tangent	enter tan θ	TAN^{-1}
Arc cotangent	enter cot θ	1/x TAN^{-1}
Arc secant	enter sec θ	1/x SIN^{-1}/COS^{-1} 2ND FUNC
Arc cosecant	enter csc θ	1/x SIN^{-1}/COS^{-1}

θ→R/R→θ

θ→R/R→θ converts decimal degrees to radians or radians to decimal degrees. θ→R/R→θ assumes the number in the display to be in degrees. It converts the number to radians and puts radians in the display.

At the same time it assumes the number in the display to be in radians, it makes another conversion (from radians to degrees) and puts degrees in the 2ND FUNC register. If the Grad-Degree switch is in the grad position (GRAD/DEG), it converts grads to radians and radians to grads.

θ→R/R→θ DEC→D.MS

To convert 360 degrees to radians

Enter This	*See This*
3 6 0 θ→R/R→θ	6.2831

360° = 6.2831 radians

To convert π radians to degrees

Enter This	*See This*
π θ→R/R→θ 2ND FUNC	180.0000

π radians = 180° which is just what we learned in school.

DEC→D.MS

DEC→D.MS converts any angle displayed in decimal degrees to the D.MS format.

To convert 45.5 degrees to Degrees, Minutes, and Seconds.

Enter This	*See This*
4 5 • 5 DEC→D.MS	45.3000

45.5° = 45°30'00"

To convert 45.1234 to Degrees, Minutes, and Seconds.

Enter This	*See This*
4 5 • 1 2 3 4 DEC→D.MS	45.0724

45.1234° = 45°07'24"

Don't try to take the sine, cosine, or tangent of an angle that is displayed as D.MS. The Micro Scientist always reads it as decimal degrees, even though you may be thinking in degrees, minutes, and seconds.

DEC→D.MS has no effect with the degree grad switch in the GRAD/DEG position.

[D.MS→DEC] [x⇄y / θ,r]

D.MS→DEC

[D.MS→DEC] converts degrees, minutes, and seconds to decimal degrees. It is the inverse of the [DEC→D.MS] key.

To express 45°15'42" in decimal degrees.

Enter This		*See This*
4 5 . 1 5 (*Degrees*, *Minutes*)		
4 2 [D.MS→DEC] (*Seconds*)	*45°15'42" = 45.2616°*	45.2616

The [DEC→D.MS] and [D.MS→DEC] keys are used together in many problems. Use the [D.MS→DEC] key to convert for calculations. After all calculations are performed, convert back to the DMS format using the [DEC→D.MS] key. Think of the DMS format only as an entry and display format to avoid any confusion.

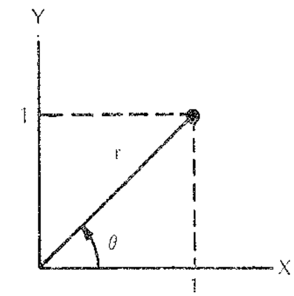
[x⇄y / θ,r]

[x⇄y / θ,r] converts rectangular coordinates to polar coordinates.

To use it, follow this sequence:

1. "Display" the x coordinate
2. Press [x⇄y / θ,r]
3. "Display" the y coordinate
4. Press [=]. This operation puts θ in the display and r in the [2ND FUNC] register.
5. Press [2ND FUNC]. The radius r is displayed.

If a point has the rectangular coordinate of x = 1 and y = 1, what are its polar coordinates?



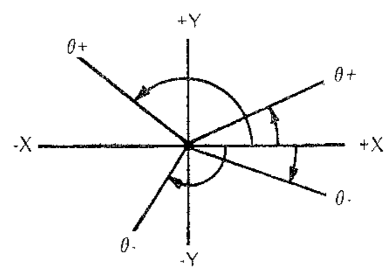
Enter This		*See This*
1 [x⇄y / θ,r] 1 [=]	(θ)	45.0000
[2ND FUNC]	(r)	1.4142

The point (x,y) = (1, 1) when expressed in polar coordinates is given by:

$$\theta = 45°$$

$$r = 1.4142$$

Note that the angle θ is measured counter-clockwise from the X axis for positive values of Y and clockwise from the X axis for negative values of Y. The following drawing clarifies this for you.



X and Y may be positive or negative. r is always positive and θ ranges from -180 to +180 degrees or -200 to +200 grads.

[θ⇄r x,y]

[θ⇄r x,y]

[θ⇄r x,y] converts polar coordinates to rectangular coordinates. To use it, follow this sequence:

1. Display θ
2. Press [θ⇄r x,y]
3. Display r
4. Press [=]. This operation puts X in the display and Y in the [2ND FUNC] register.
5. Press [2ND FUNC]. Y is displayed.

If a point has polar coordinates of θ = 220° and r = 10, find its rectangular coordinates.

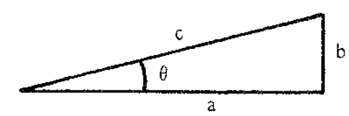
Enter This		*See This*
2 2 0 [θ⇄r x,y]		
1 0 [=]	x	-7.6604
[2ND FUNC]	y	-6.4278

When using this key, angles may have any positive or negative value. The above example gives the same results with θ = –140°. Try it!

FUNCTION MIXING

You can begin to appreciate the power of the Micro Scientist when you need to make calculations involving several functions.

To find the length of side c of a triangle with side a, and the angle θ known



when
a = 3
θ = 32°
$c = \dfrac{a}{\cos\theta}$

Enter This		*See This*
3 [÷] 3 2		
[SIN COS] [2ND FUNC] [=]	*side c*	3.5375

To find the 12th root of 100

Enter This	*See This*
1 0 0 [a^x]	
1 2 [1/x] [=]	1.4677

To find $(7+\sqrt{3})^{1/1.5}$

Enter This	*See This*
7 [+] 3 [√] [a^x]	
1 . 5 [1/x] [=]	4.2404

DATA STORAGE REGISTERS

ST_n RCL_n EXCH_n

The Micro Scientist has 10 storage registers. Each storage register has a capacity of 13 digits plus exponent. ST_n and RCL_n when used with the numeral keys let you STORE numbers and RECALL numbers. To store a number that's in the display, press ST_n and one of the 10 numerical keys 0 through 9.

To store 1.1234 in register 0

Enter This	*See This*
1 . 1 2 3 4	
ST_n 0	1.1234

After a number is stored, you may do any operation that you wish, (e.g. CLEAR ENTRY). Later when you need the number again, recall it.

Enter This	*See This*
RCL_n 0	1.1234

When a number is stored, it replaces whatever number is in that register. When you recall a register, the number in the register isn't changed. It's copied into the display and also kept in the register. You can demonstrate this for yourself by doing a CLEAR ENTRY and another RCL_n 0

EXCH_n exchanges the number in the display with the number in any of the 10 storage registers. This is useful if you have filled all 10 registers and want to use a stored number without losing the number in the display. It can also be useful as a step saver in many situations.

To demonstrate EXCH_n , enter and store the number 2 into the register 0 (2 ST_n 0)and do the following steps.

Enter This	*See This*
√	1.4142
EXCH_n 0	2.0000
EXCH_n 0	1.4142
EXCH_n 0	2.0000

Remember, that if you turn your Micro off, all registers are cleared to zero.

USING REGISTERS DURING ARITHMETIC

When doing calculations, you can add, subtract, multiply, divide, a^x, and compute other functions using registers as you would numbers.

Starting with a simple number, we'll show a series of operations that demonstrate how the registers can be used in making calculations.

Enter This	*See This*
1 . 0 0 0 1	
ST_n 0	1.0001
2 + RCL_n 0 =	3.0001
(Same as 2 + 1.0001)	
ST_n 1	3.0001
RCL_n 1 − RCL_n 0 =	2.0000
(Same as 3.0001 - 1.0001)	
ST_n 0	2.0000
RCL_n 1 × RCL_n 0 =	6.0002
(Same as 3.0001 X 2.0000)	

[÷] [RCL$_n$] 0 [=] → 3.0001

(Same as 6.0002 ÷ 2.0000)

This example shows operations that are strictly algebraic. There is another way to use the registers in your calculations that isn't conventional but is shorter and faster. We call it direct register arithmetic.

DIRECT REGISTER ARITHMETIC

ARITHMETIC INTO REGISTERS

Rules for Arithmetic into Registers

- [ST$_n$] [+] n adds the display to register n
- [ST$_n$] [−] n subtracts the display from register n
- [ST$_n$] [×] n multiplies the display into register n
- [ST$_n$] [÷] n divides the display into register n

We'll demonstrate this faster technique by storing our starting number in register 0.

Enter This	*See This*
4 . 4 4 4 2	
[ST$_n$] 0	4.4442
To add 2 to register zero	
2 [ST$_n$] [+] 0	2.0000
[RCL$_n$] 0	6.4442
To subtract 6 from register zero	
6 [ST$_n$] [−] 0	6.0000

[RCL$_n$] 0	0.4442
To multiply register zero by 2	
2 [ST$_n$] [×] 0	2.0000
[RCL$_n$] 0	0.8884
To divide register zero by 4	
4 [ST$_n$] [÷] 0	0.8884
[RCL$_n$] 0	0.2221

The procedure is easy if you remember to press [ST$_n$] first, the arithmetic operation next, and the register number last.

For all the store-arithmetic operations, the display is not changed and the register contains the result of the arithmetic.

ARITHMETIC OUT OF REGISTERS

Rules for Arithmetic out of Registers

- [RCL$_n$] [+] n adds register n to the display
- [RCL$_n$] [−] n subtracts register n from the display
- [RCL$_n$] [×] n multiplies register n into the display
- [RCL$_n$] [÷] n divides register n into the display

For all the recall-arithmetic operations, the register is not changed and the display shows the result of the arithmetic.

To demonstrate arithmetic out of registers we'll start by storing the number 12 in register 3.

Enter This	*See This*
1 2 [ST$_n$] 3	12.0000

To add the contents of register 3 to the number (20) in the display

2 0 RCL n + 3 — 32.0000

To subtract register 3 from the number (30) in the display

3 0 RCL n − 3 — 18.0000

To multiply register 3 by the number (4) in the display

4 RCL n × 3 — 48.0000

To divide the number (24) in the display by register 3

2 4 RCL n ÷ 3 — 2.0000

Direct register arithmetic is a powerful shortcut. It saves time and steps and is well worth mastering.

APPLICATIONS

Now that we have studied all the keyboard functions, let's put our knowledge to work in solving some typical problems.

PHYSICS

A prediction of the special theory of relativity is that the mass of a moving body is greater than its mass when at rest. The equation that describes this is given by:

$$M = \frac{M_o}{\sqrt{1 - v^2/c^2}}$$

where

- M = the mass of the moving body
- M_o = the rest mass of the body
- V = the speed of the body relative to an observer
- C = the speed of light (2.9979×10^8 meters/second)

Using the Micro Scientist, find the mass of an electron traveling at 99% of the speed of light. The rest mass of an electron is 9.1086×10^{-31} kg.

If v = 0.99C, then the equation becomes:

$$M = \frac{M_o}{\sqrt{1 - \frac{(0.99C)^2}{C^2}}} = \frac{M_o}{\sqrt{1-(0.99)^2}}$$

Substituting numerical values:

$$M = \frac{9.1086 \times 10^{-31}\ \text{kg}}{\sqrt{1 - (0.99)^2}}$$

Using the keyboard, the solution works like this:

Enter This	*See This*
9 . 1 0 8 6	9.1086
EXP CHG SIGN 3 1	9.1086 -31
÷ (1 − (	
. 9 9	0.99 0
× =)) √ =	6.456915343 -30

The mass of an electron traveling at 99% of the velocity of light is 6.4569×10^{-30} kg.

Although the example was quite simple, it demonstrates some basic keyboard techniques. Let's try something else.

HYPERBOLIC FUNCTIONS

Suppose that you are working with the hyperbolic function: sinh x, cosh x, and tanh x. With the Micro Scientist, evaluating these functions is easy.

Function	Equation
sinh (x)	$\frac{e^x - e^{-x}}{2}$
cosh (x)	$\frac{e^x + e^{-x}}{2}$
tanh (x)	$\frac{e^x - e^{-x}}{e^x + e^{-x}}$ or $\frac{\sinh x}{\cosh x}$

To find the sinh x for x = 0.98

Enter This	*See This*
. 9 8 e^x/10^x	2.6644
− 1/x	0.3753
÷ 2 =	1.1445

sinh (.98) = 1.445

To find the cosh (.98), simply use a + *in the sequence instead of the* − .

Try it!

If you get 1.5198, you can throw away your slide rule and tables.

To find the tanh (.98), you can calculate it directly from the equation like this:

Enter This	*See This*
. 9 8 e^x/10^x STn 0	2.6644
− 1/x ÷	2.2891
(RCLn 0 + 1/x)	3.0397
= *Tanh (0.98)*	0.7530

Or you can use the registers to store your calculations for sinh (x) and cosh (x) and then calculate tanh (x) from the ratio:

$$\frac{\sinh (x)}{\cosh (x)}$$

QUADRATIC EQUATIONS

Finding the real roots of a quadratic equation is often useful.* Using the general formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Let's find the roots of the equation:

$$3x^2 - 10x + 6 = 0$$

a = 3

b = -10

c = 6

To save time and possible confusion, analyze the problem so that it can be solved by a logical sequence of steps. Here's one way to do it.

Enter This		*See This*
3 STn 0	*store a*	3.0000
1 0 CHG SIGN STn 1	*store b*	-10.0000
6 STn 2	*store c*	6.0000
RCLn × 0 × 4		
=	*4ac*	72.0000
CHG SIGN + (RCLn 1		
× =) = √	$\sqrt{b^2 - 4ac}$	5.2915
STn 3		

*A method for finding both the real and complex roots for quadratic equations is described in the section on programming (see page 53).

RCLn 1 CHG SIGN RCLn +
3 ÷ 2 RCLn ×
0 = *1st Real Root* 2.5485
RCLn 1 CHG SIGN RCLn −
3 ÷ 2 RCLn ×
0 = *2nd Real Root* 0.7847

For the equation:

$$3x^2 - 10x + 6 = 0$$

The real roots are:

x = 2.5485 and

x = 0.7847

This same sequence of steps will work for any quadratic equation with real roots, simply by changing the coefficients a, b, and c that are stored in the first few steps. This principle of using the same sequence of steps to solve a general problem leads us quite naturally into scratch pad programming which will be discussed in the next sections.

SCRATCHPAD PROGRAMMING

The added capability of scratchpad programming opens a whole new dimension of computation. Programming is most useful if you need to make the same general calculation many times with a lot of different data. Scratchpad programming is essentially automatic key pressing. Once you have found an efficient sequence of steps that solves a particular problem, you can program those steps to work for a generalized problem with a full range of variables.

The 324 has two separate 80-step memories. This means that you can store up to 80 (or 2 x 80) keyboard operations at one time and they work automatically for as long as you want.

Here are a few basic rules to keep in mind when using the programming feature of the Micro Scientist.

- There are 80 program steps. If you go past 80 steps, you will end up back at step 01 and your original program will be erased by the new steps.
- Programming is essentially working a particular calculation sequence when the machine is in RUN/LOAD (LOAD). Once loaded, the Micro Scientist automatically performs the sequence any time you need it.
- If you make a mistake while loading a program, switch from LOAD to RUN and back to LOAD. Start over.
- Use START/STOP at the points in your program where you want to enter variables or data and at the points where you want answers to be held in the display.
- If you have room in your program, use identification numbers to indicate variable entry points and to help keep track of where you are in the program.
- Once the program is loaded, switch to RUN/LOAD (RUN) for execution.
- In general, press START STOP to start the program and get to the point where you enter the first variable.

- Program 1 and 2 are independent. The program switch [1 PROG 2] can't be used as a program step. You can only switch from one memory to the other when RUN/LOAD is up [RUN/LOAD].
- If [E····] displays during loading, [RESET], switch to RUN and then switch back to LOAD. Start loading the program again.
- During the running of a program, you should be able to do independent calculations when the program is at a stop. This means that you should not leave any algebra hanging in the program at the stop. (e.g., [2] [+] [START STOP]).
- If you use [RESET] or [CLEAR ENTRY] while a program is running, the program stops and the display is cleared. [START STOP] starts the program running again from step 01.
- Digit entry keys, [•], [EXP] and [CHG SIGN] cannot be used as program steps immediately after [START STOP].

With these rules in mind, let's write a simple program. The volume of a sphere is given by the equation,

$$V = \frac{4}{3}\pi r^3$$

With scratchpad programming, calculating the volume once automates finding the volume of a sphere of any radius.

At the top of the keyboard is the [START STOP] button and the [RUN/LOAD] switch.

There is also a program selector switch [1 PROG 2]. This switch is discussed in the next section. For now just leave it in the "1" position, [1 PROG 2], and forget about it.

Put the RUN/LOAD switch in the load position. Press [RESET] and [RUN/LOAD]. The display breaks into three sections and looks like this:

[0.0000000 00 01]

Mantissa Exponent Step Number

At the far right of the display is a two digit number that tells the step number to be loaded *next*. At the left is the displayed-with-seven-digits mantissa followed by the two digit exponent for scientific notation.

To load a program that will solve our example problem, follow this procedure:

	Enter This	*See This*
1	[START STOP]	0.000000 00 02
2	[x]	1.000000 00 03
3	[3]	3.000000 00 04
4	[=]	0.000000 00 05
5	[×]	0.000000 00 06
6	[π]	3.141592 00 07
7	[×]	0.000000 00 08
8	[4]	4.000000 00 09
9	[÷]	0.000000 00 10
10	[3]	3.000000 00 11
11	[=]	0.000000 00 12
12	[START STOP]	0.000000 00 13

[RUN/LOAD] *0.0000 is displayed*

The program is loaded and ready to run.

RUNNING THE PROGRAM

Programs run in different ways depending on how they are written. To run this program follow these steps.*

1. START/STOP
2. Enter a value for the radius (r). 2 for example.
3. START/STOP The calculation for V displays

33.5103

4. For a new calculation of V return to step 1.

Here is a table showing the volume of a sphere as a function of the radius. The table was made by running the program several times with different radii.

Radius	Volume
1	4.1887
2	33.5103
3	113.0973
4	268.0825
4.125	294.0088

*If you make a mistake and want to get back to the beginning of the program, simply put the run/load switch in LOAD and then back to RUN. Start running the program again from step 1.

PROGRAMMING HYPERBOLIC FUNCTIONS

These same techniques are applicable to our previous problem of calculating the hyperbolic functions. Recalling the equations:

$$\text{Sinh } x = \frac{e^x - e^{-x}}{2}$$

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

$$\text{Tanh } x = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{\text{Sinh } x}{\text{Cosh } x}$$

Here's how they can be written into a program. RESET RUN/LOAD

Step	Key	
1	1	Identifier
2	SET DP	
3	0	
4	START/STOP	Enter x
5	e^x / 10^x	$e^x - e^{-x}$
6	ST_n	
7	1	
8	−	
9	$\frac{1}{x}$	
10	=	

Step	Key	
11	÷	$\frac{e^x - e^{-x}}{2}$
12	2	
13	=	
14	ST_n	
15	2	
16	RCL_n	$e^x + e^{-x}$
17	1	
18	+	
19	$\frac{1}{x}$	
20	=	

Step	*Key*	
21	÷	$\frac{e^x + e^{-x}}{2}$
22	2	
23	=	
24	EXCH$_n$	*Stores Sinh(x) in register 1, Cosh(x) in register 2*
25	2	
26	ST$_n$	
27	1	

Step	*Key*	
28	RCL$_n$	*Calculates tanh(x) and stores it in register 3*
29	÷	
30	2	
31	ST$_n$	
32	3	
33	SET DP	*Resets the decimal point to four places, clears the display and stops the program*
34	4	
35	0	
36	START/STOP	

RUN / LOAD

RUNNING THE PROGRAM

1. START/STOP — *I*· displays
2. Enter x — (0.6 for example)
3. START/STOP — The program runs until 0.0000 displays. The hyperbolic functions are stored in registers 1, 2, and 3
4. For sinh(x), RCL$_n$ 1 — sinh(0.6) — 0.6366
   For cosh(x), RCL$_n$ 2 — cosh(0.6) — 1.1854
   For tanh(x), RCL$_n$ 3 — tanh(0.6) — 0.5370

5. For another calculation return to step 1.

With a little thought and planning a similar program could be written for the calculation of the inverse hyperbolic functions. As an exercise, write a program for the inverse hyperbolic sine using this equation.

$$\sinh^{-1} x = \ln(x + \sqrt{x^2 + 1}) \qquad -\infty < x < \infty$$

If you get $\sinh^{-1}(1) = 0.8813$ when you run your program, you're on your way to being an expert Micro programmer. If you didn't get the correct answer, try using the program shown below. RESET RUN / LOAD

Step	*Key*	
1	START/STOP	*Enter X (1)*
2	+	
3	(	
4	(	
5	×	
6	=	
7	+	

Step	*Key*
8	1
9	)
10	√
11	)
12	=
13	Ln LOG
14	START/STOP

RUN / LOAD

MORE ABOUT PROGRAMMING

As you become more familiar with your Micro Scientist, you can begin to realize its full potential. We have already shown the basics needed to start writing programs using one 80 step memory. In this section we show some more programming concepts and introduce two memory scratch pad programming. This section will help you get the most out of your machine because we have included some useful tricks and techniques.

As a kind of quick review, let's take our very simple program for finding the volume of a sphere and make it a little more complex. Suppose that you are doing some rough calculations in designing a spherical pressure vessel. You wish to find the actual volume of material used in making vessels of various diameters and thicknesses. We already know that the volume of a sphere is given by:

$$V = \frac{4}{3}\pi r^3$$

To find the material volume of a hollow sphere of thickness t, all that is needed is to calculate the volume of the non hollow sphere of radius r and subtract the volume of a sphere of radius (r – t). The equations look like this:

$V_S = \frac{4}{3}\pi r^3$ — Volume of the solid sphere

$V_H = \frac{4}{3}\pi(r-t)^3$ — Volume of the hollow space in the pressure vessel

$V_m = V_S - V_H$

$V_m = \frac{4}{3}\pi[r^3 - (r-t)^3]$

} Material volume of a hollow sphere

If

$r = 10$

and

$t = 0.1$

then we can find V_m using the following sequence of steps.

1 0 STπ 0 a^x 3 — r^3

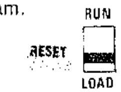
– ((RCLπ 0 – . 1) a^x 3) = — $-(r-t)^3$

× 4 × π ÷ 3 — $x(\frac{4}{3}\pi)$

= — V_m *displayed* 124.4112

To generalize the calculation to work for any r and any t, load the sequence as a program, replacing the steps where we entered 10 for r and 0.1 for t with START/STOP. We should also put a START/STOP at the end to hold the calculation for V_m in the display. To keep track of where we are, we will display 1. at the point where r is to be entered and 2. at the point were t is to be entered.

Here's the program.



Step	*Key*		*Step*	*Key*	
1	1	*This displays* 1. *as an identification for the entry of r*	5	STπ	r^3
2	SET DP		6	0	
3	0		7	a^x	
4	START STOP	*Enter r*	8	3	

MORE ABOUT PROGRAMMING
HOLLOW SPHERE

Step	Key	
9	−	
10	(	
11	RCL$_n$	
12	0	
13	−	
14	2	*Identifier 2 to enter t*
15	START/STOP	*Enter t*
16	ST$_n$	
17	1	
18	a^x	
19	3	

(Steps 13–19: $r^3 - (r-t)^3$)

Step	Key	
20	)	
21	×	
22	4	
23	×	
24	π	
25	÷	
26	3	
27	=	V_m
28	SET DP	*Change decimal point for answer*
29	4	
30	START/STOP	*Display V_m*

(Steps 20–26: $\frac{4}{3}\pi\,[r^3 - (r-t)^3]$; steps 28–29: *Change decimal point for answer*)

Running the program

1. START/STOP — *1. displays*
2. Enter radius r — *(10 for example)*
3. START/STOP — *2. displays*
4. Enter thickness t — *(0.1 for example)*
5. START/STOP — V_m displays `124.4112`
6. To run the program again (using any r and any t) return to step 1.

MORE ABOUT PROGRAMMING
SURFACE AREA OF A HOLLOW SPHERE

You can see that scratchpad programming adds a whole new dimension to the Micro Scientist's capacities. It frees your mind from the tedium of pencil and paper doodling and lets you experiment. Equations become ideas rather than complicated manipulations. The machine itself becomes a scratchpad and a real world extension of your mathematical thinking.

With the hollow sphere program loaded into program memory 1, suppose now that there is a need to make a *separate* calculation for the inside and outside surface areas of the hollow sphere. With the two program capacities of the 324 Micro Scientist, we can easily load the new program into Memory 2.

The surface area equations are:

$S_O = 4\pi r^2$ — Surface area outside

$S_I = 4\pi(r - t)^2$ — Surface area inside

Using the fact that r is stored in register 0 and t is stored in register 1, the program sequence is the following: RESET, switches set to 2 (PROG) and LOAD (switch labels: 1 / 2, PROG, RUN / LOAD)

Step	Key	
1	RCL$_n$	
2	0	r
3	×	
4	=	r^2
5	×	
6	4	$4\pi r^2$
7	×	
8	π	
9	=	
10	START/STOP	*Displays S_O*
11	RCL$_n$	
12	0	
13	RCL$_n$	$r - t$
14	−	

Step	Key	
15	1	
16	×	
17	=	$4(r - t)^2$
18	×	
19	4	

Step	Key	
20	×	
21	π	$4\pi(r - t)^2$
22	=	
23	START/STOP	Displays S_I
	RUN / LOAD	

What we have now is two programs that can be used either together or separately. If used together they run like this:

1. PROG 1 / 2

2. START/STOP *1. displays*

3. Enter r (10)

4. START/STOP *2. displays*

5. Enter t (0.1)

6. START/STOP *V_m displays* 124.4112

7. Return to step 1 for a new calculation of V_m or go to step 8 to find S_O and S_I.

8. PROG 1 / 2

9. START/STOP *S_O* 1256.6370

10. START/STOP *S_I* 1231.6299

11. Return to step 1 for a new calculation.

Used separately, program 1 would operate as previously shown. Using program 2 as a separate program requires that r be prestored in register 0 and t be prestored in register 1. Program 2 would then operate like this:

1. PROG 1 / 2
2. Enter r STn 0 (10)
3. Enter t STn 1 (0.1)
4. START/STOP S_O displays 1256.6370
5. START/STOP S_I displays 1231.6299
6. Return to step 2 for a new problem.

Both of these programs can be considered to be independent. The fact that they share registers 0 and 1 keeps them from being completely independent, but in principal either of the programs can be changed without affecting the other. One important caution. Switch the program selector BEFORE you switch from RUN/LOAD to RUN/LOAD. The program selector has no effect in the RUN/LOAD position.

These programs are simple examples of two related problems which must be solved from time to time. Now let's look at another situation where two separate memories are a big help. Suppose your work requires you to compute the mean, standard deviation, and upper and lower 95% confidence limits of a group of things, like sampling the weights of small parts from a casting machine.

The formulas are not complex, but they aren't much fun when you do them by hand.

Here they are:

$$\text{Mean} = \overline{X} = \frac{\Sigma X_i}{n}$$

$$\text{Standard Deviation} = \text{S.D.} = \sqrt{\frac{\Sigma X^2 - n\overline{X}^2}{n}}$$

Upper 95% Limit = X + 2(S.D.)

Lower 95% Limit = X – 2(S.D.)

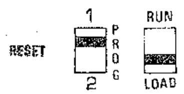
n is the number of entries

ΣX is the sum of all the entries

ΣX^2 is the sum of the squares of all the entries

The idea is to enter each weight measurement once, sum it, square it, sum the squares and count the entries. When all the weights are entered, the other calculations can be done. So, we can use two programs—one for data entry and one for the desired results. Data entry will be program 1.

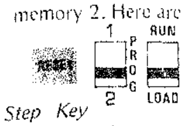
Here's what to do.



Step	Key	
1	ST_n	*Direct register arithmetic ΣX*
2	+	
3	1	
4	×	ΣX^2
5	=	
6	ST_n	
7	+	
8	2	
9	1	*n count*
10	ST_n	
11	+	
12	3	
13	CLEAR ENTRY	*Data entry point*
14	START STOP	

The program is now loaded.

With the data entry program in memory 1 we can now load the cleanup program into memory 2. Here are the steps for program 2.



Step	Key	
1	RCL_n	*Calculates the mean, $\overline{X} = \frac{\Sigma X}{n}$, stores it in register 4, and holds it in the display*
2	1	
3	RCL_n	
4	÷	
5	3	
6	ST_n	
7	4	
8	START STOP	
9	×	*Calculates Standard Deviation $S.D. = \sqrt{\frac{\Sigma X^2 - n\overline{X}^2}{n}}$*
10	×	
11	RCL_n	
12	3	
13	CHG SIGN	
14	+	
15	RCL_n	
16	2	
17	÷	*Calculation of S.D. (cont'd)*
18	RCL_n	
19	3	
20	=	
21	$\sqrt{\ }$	
22	START STOP	*Displays S.D.*
23	×	*2 x S.D.*
24	2	
25	=	
26	ST_n	
27	5	
28	RCL_n	*Upper Limit X +(2 x S.D.)*
29	4	
30	RCL_n	
31	+	
32	5	

Step	*Key*		*Step*	*Key*	
33	[START/STOP]	*Displays Upper Limit*	*38*	[5]	
34	[RCLₙ]	*Lower Limit*	*39*	[START/STOP]	*Displays Lower Limit*
35	[4]	*X − 2 x S.D.*		RUN / LOAD	
36	[RCLₙ]				
37	[−]				

Now that both programs are loaded, we are ready to use them with real data. Since we will be using registers 1, 2, and 3 as summation accumulators, it's important that we clear them before starting.

[0] [STₙ] [1]
[STₙ] [2]
[STₙ] [3]

Here's the data we'll be working with.

Weight of casting	3.5	3.6
	3.7	3.5
	3.8	3.8
	3.5	3.4

Running the Program

1. PROG 1/2

2. Enter the data, pressing [START/STOP] after each entry. When all the data has been entered, go to step 3.

If you want the summations at any time:

[RCLₙ] [1] for ΣX
[RCLₙ] [2] for ΣX^2
[RCLₙ] [3] for n

For the above data: $\Sigma X = 28.8$
$\Sigma X^2 = 103.84$
$n = 8$

3. PROG 1/2

[START/STOP]	$\bar{x}$ displays	3.6000
[START/STOP]	SD displays	0.1414
[START/STOP]	Upper Limit	3.8828
[START/STOP]	Lower Limit	3.3171

4. You can switch to program 1 and return to step 1 to *add* more data *or* you can switch to program 1, clear registers 1, 2, and 3 and return to step 1 to start a new problem with all new data.

DECISION MAKING

The 324 can make decisions. There are two types of "decisions" which can be made. One is a sort of counting decision; the other a quantitative decision. Let's take the counting type first. Suppose you want to be able to take factorials. You know that n Factorial = n! = n(n - 1) (n - 2) . . . (2)(1). So obviously, you want to enter a number n and let the 324 do the work. It can. Here's the program:

RUN
RESET
LOAD

Step	Key	
1	ST_n	Register 1 is used for the current value of N
2	1	
3	ST_n	Register 2 is used for building the factorial
4	×	
5	2	
6	1	
7	ST_n	
8	−	

Step	Key	
9	1	*This decrements N*
10	RCL_n	*When N = 0, E···· will display (1/0 gives error)*
11	1	
12	$\frac{1}{x}$	
13	RCL_n	*New values of N*
14	1	*Program will return to step 1, looping until N = 0*

RUN
LOAD

Running the Program

Here's how to run the program:

1. 1 ST_n 2 Prestore the decrementing value.
2. Enter N START/STOP (e.g., N = 5)

   Program runs until E···· displays.
3. RESET RCL_n 2 displays N!

   (5! = 120)

Try it again with another number, 47 for example. This will take longer, but finally E···· displays.

RESET RCL_n 2 gives 47! 2.586232415 59

You can see that 1/x is an effective means of providing loop control when decrementing to zero. Dividing by a register which contains zero might also be used.

Many of the mathematical processes we use don't ever give zero results, and thus the divide by zero technique is not applicable. Let's look at another method. Suppose, for example, we want to find a maximum value of a function:

$$Y = aX^2 + bX + c$$

where

a = −2

b = 3

c = 4

The easiest way is to evaluate the equation at various points and compare subsequent results. As soon as a result becomes less than the previous one, the maximum value has been reached. Let's try it.

1. Store the values of a, b, and c in registers 1, 2, and 3.

CHG SIGN 2 ST_n 1

3 ST_n 2

4 ST_n 3

2. Load the following program, the current value of X is assumed to be in register 9.

RUN / LOAD

Step	*Key*	
1	RCL n	
2	1	
3	×	*aX*
4	RCL n	
5	9	
6	+	
7	RCL n	*aX + b*
8	2	
9	×	
10	RCL n	*aX² + bX*
11	9	
12	+	
13	RCL n	*aX² + bX + c*
14	3	
15	=	
16	EXCH n	*Puts the new value of Y in register 8 and the previous value in register 7*
17	8	
18	ST n	
19	7	
20	RCL n	
21	8	
22	RCL n	*Put the difference between the new and old value in the display*
23	−	
24	7	
25	√	*This is the test. The first time the difference between the new and old value is negative* E . . . . *displays. Recall that √ of a negative number is an illegal operation*
26	RCL n	
27	0	
28	ST n	*This increments the value of X by ΔX.*
29	+	
30	9	

RUN / LOAD

Running the Program

1. Verify that a, b, and c are correctly stored in registers 1, 2, and 3.
2. Store a starting value for X in register 9. Let's start with $X_0 = -5$.
3. Store the increment to X in register 0. Use 0.25 for this increment.
4. START/STOP The program loops until E . . . . is displayed.
5. RESET RCL n 9
6. If your starting value for X is displayed, press START/STOP. If not, go to step 8.
7. The program loops again until E . . . . is displayed.
8. RESET RCL n 7 The *computed* maximum value for Y is displayed. (5.1250).
9. RCL n 9 − RCL n 0 = Displays the X value (0.7500) that corresponds to the computed Y_{max}. Of course, what you have is the nearest value to Y_{max} allowed by the 0.25 increment. In this example X = 0.75.
10. For a new program store new values for a, b, and c in registers 1, 2, and 3. Return to step 1.

ROOTS OF QUADRATIC EQUATIONS

Here's an example that combines the decision techniques with the two memory capability. We wish to find the roots of the quadratic equation:

$$ax^2 + bx + c = 0$$

We know from elementary algebra that a solution is given by:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The roots are real if the term $b^2 - 4ac \geq 0$ and they are complex if $b^2 - 4ac < 0$. With two memories we can solve for both the real and complex roots. Here's how it's done. To make sure that the program loads correctly prestore these values:

1. 1 STn 0
   2 STn 1
   1 STn 2

2. RESET — PROG switch to 1 (1 PROG 2); RUN/LOAD switch to LOAD

3. Load the following program steps:

Step	Key	
1	RCLn	-4ac
2	×	
3	0	
4	×	
5	4	
6	CHG SIGN	
7	+	$b^2 - 4ac$
8	(	
9	RCLn	
10	1	
11	×	
12	=	
13	)	
14	=	
15	STn	Stores $b^2 - 4ac$
16	3	
17	√	Test is made at this point. If [E------], use the second program
18	STn	
19	3	
20	RCLn	-b
21	1	
22	CHG SIGN	

Step	Key	
23	RCLn	$\frac{-b - \sqrt{b^2 - 4ac}}{2a}$
24	+	
25	3	
26	÷	
27	2	
28	RCLn	
29	×	
30	0	
31	=	
32	START STOP	Displays 1st root (x_1)

Step	Key	
33	RCLn	$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$
34	1	
35	CHG SIGN	
36	RCLn	
37	−	
38	3	
39	÷	
40	2	
41	RCLn	
42	×	
43	0	
44	=	
45	START STOP	Displays 2nd root (x_2)

RUN/LOAD switch to RUN

The first program solves for real roots but goes into error if the roots are complex. The second program is then run to find the complex roots. Here's how to work the second program.

0 STn 3 — switch: 1 / PROG / 2; RUN / LOAD

Step	Key	
1	RCLn	$b^2 - 4ac < 0$
2	3	
3	CHG SIGN	$b^2 - 4ac > 0$
4	$\sqrt{\ }$	$\sqrt{b^2 - 4ac}$
5	STn	
6	4	
7	RCLn	b
8	1	
9	÷	$\frac{b}{2a}$
10	2	
11	RCLn	
12	×	
13	0	
14	=	
15	CHG SIGN	$-\frac{b}{2a}$
16	START STOP	*Displays real part of complex root*
17	RCLn	$\sqrt{b^2 - 4ac}$
18	4	
19	÷	$\frac{\sqrt{b^2 - 4ac}}{2a}$
20	2	
21	RCLn	
22	×	
23	0	
24	=	
25	START STOP	*Displays imaginary part of complex root*

RUN / LOAD

Running the Program

1. Switch: 1 / PROG / 2; RUN / LOAD

2. Prestore the coefficients a, b, and c in registers 0, 1, and 2. For the equation

$$x^2 - 2x + 1 = 0$$

1 STn 0
2 CHG SIGN STn 1
1 STn 2

3. START STOP the first root displays — 1.0000
4. START STOP the second root displays — 1.0000

The roots of the equation $x^2 - 2x + 1 = 0$ are $x_1 = 1$ and $x_2 = 1$.

Let's try the equation

$$x^2 + x + 1 = 0$$

5. 1 STn 0
STn 1
STn 2

6. START STOP E---- displays indicating roots are complex conjugates

7. RESET — switch 1 / PROG / 2 — Shift to program 2

8. START STOP The real portion of the complex roots displays -0.5000
9. START STOP The absolute value of the imaginary portion of the roots displays 0.8660

Therefore, the roots of the equation $x^2 + x + 1 = 0$ are:

$x_1 = -0.5 + 0.866i$
$x_2 = -0.5 - 0.866i$ $\quad (i = \sqrt{-1})$

MORE ABOUT PROGRAMMING
DECISIONS: ROOTS OF QUADRATIC EQUATIONS

In case you found that all confusing, here is a generalized procedure for running the programs:

1. Prestore the coefficients a, b, c in registers 0, 1, and 2.
2. [1 PROG 2 switch] The coefficient c must be in the display.
3. [START STOP] The real root x_1 is displayed.
   If the program goes into error, the roots are complex. Go to step 6 below. Otherwise
4. [START STOP] The real root x_2 is displayed.
5. Return to step 1 for a new problem.
6. [1 PROG 2 switch] RESET to find the complex roots.
7. [START STOP] The real portion (R) of the complex root is displayed.
8. [START STOP] The absolute value of the imaginary portion | C | of the complex root is displayed (i.e., the coefficient of i).
9. Combine the real and imaginary portions to get the complex conjugate pair.

$$x_1 = R + Ci$$
$$x_2 = R - Ci$$

10. Return to step 1 for a new problem.

ACCURACY AND NUMBER SIZES

All numbers in the Scientist have 13 significant digits. Angles are carried internally as decimal degrees or grads to the same significance. A two-digit exponent with each number gives the magnitude. This allows the range to go from 10^{-99} to 10^{99}. Significance is not lost when numbers are displayed to any number of decimal places, since the number retained internally is not modified.

Add, subtract, multiply and divide are accurate to the full 13 digits. Divide is actually rounded in the 13th digit, to gain a bit more accuracy.

[a^x], [ln/LOG], [e^x/10^x], [SIN/COS], [SIN^{-1}/COS^{-1}], [TAN], [TAN^{-1}], and [θ→/R→] are almost always accurate to 1 part in the 12th digit and are never worse than 1 part in the 11th digit. Since the [ ] and [ ] conversions use the trig functions, they have similar accuracy.

[√] is good to three parts in the 13th digit. [1/x] is a divide so it has the same accuracy as divide.

That's a lot of accuracy--more than you will need in most cases. But when you do need it, it is there.

ERROR CONDITIONS

The following causes E. . . .

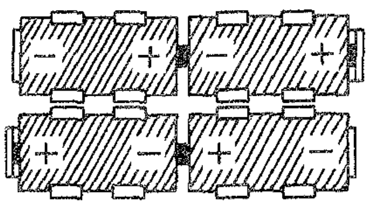
- Dividing by zero
- 1/x of zero
- $\sqrt{}$ of a negative number
- Log of zero or negative number
- Arc sine of a number greater than 1 or less than −1
- Entry of more than 13 digits (or a decimal point followed by more than 12 digits)
- Taking the tangent of 90° or 100 grads or any odd multiple of these angles
- Calculating 0^{-x} with the a^x key
- Calculating a^{-x} with negative a and noninterger x
- Calculations that produce numbers beyond 10^{+99} to 10^{-99}
- Pressing two or more keys simultaneously
- Pressing more than two keys in succession while the machine is busy calculating (it can remember two keystrokes)

BATTERIES AND RECHARGING

The Micro Scientist operates on batteries when not connected to an AC power source. You can connect it to the AC charger supplied with the Micro Scientist—the charger plugs into any standard 110 volt wall outlet.

The nickel-cadmium batteries supplied are rechargeable. Useful life is approximately 1000 charge/discharge cycles. When fully discharged, the batteries can be recharged overnight with the Micro Scientist turned off. When turned on and operating, recharging takes somewhat longer.

To remove the batteries, turn the Micro Scientist over, squeeze together the two round plungers at the top of the case and pull outward on the handle. Remove the batteries and replace with fresh ones—Nickel-Cadmium batteries; Monroe Part No. CD 3400017 are recommended.



NOTE: Observe polarity of batteries when installing. Improper battery installation may severely damage the calculator. Follow the diagram when installing batteries.

The AC adapter always supplies charging current to the batteries, whether the Micro Scientist is on or off. For this reason, you must not use any type battery other than nickel-cadmium. In an emergency, you may substitute mercury or alkaline batteries, but ONLY if the charger is disconnected. Discharged mercury or alkaline batteries must be removed before the AC adapter is reconnected. Failure to do this may damage the Micro Scientist or the batteries may explode.

KEY INDEX

KEY	PAGE
x□y / θ, r	20, 21
θ□r / x, y	21, 22
ST_n	24, 25, 26
RCL_n	24, 25, 26, 27, 28
$EXCH_n$	24, 25
GRAD / DEG	1, 16, 17, 18, 19
RUN / LOAD	33, 34
1 / PROG / 2	34, 43
START / STOP	33, 34

INDEX

INDEX

Litton MONROE

Monroe, The Calculator Company
Orange, New Jersey • Toronto, Canada • Zurich, Switzerland
Sales and Service Throughout the World

1807-S Printed in U.S.A.