

© 1976 Compuser Design Corporation. All rights reserved. Printed in the U.S.A.
Computer Design Corporation

COMPUCORP MICRO STATISTICIAN

SUPPLEMENTARY OPERATING INSTRUCTIONS

**COMPUCORP
MICRO STATISTICIAN**

**SUPPLEMENTARY
OPERATING
INSTRUCTIONS**

INTRODUCTION

This supplement for users of the Compucorp 344 Micro Statistician gives instructions for using its unique dual 80-step program feature.

Scratchpad programming is a powerful tool when efficient use is made of the Micro's twin 80-step storage capacity. In this supplement we show, through examples how efficient programs are written.

Examples are not overly complex, but reflect realistic statistical problems. Efficient programming often requires imagination and some educated trickery. We can't give you imagination but we can show you some useful tricks and some good common sense methods which allow you to get the most out of the world's first hand-held programmable statistical computer.

REVIEW

Before getting into scratchpad programming methods, let's review the operation of the Micro Statistician.

The on/off switch is on the back. Fully charged batteries will operate the machine for about 3-1/2 hours. Batteries may be recharged while you calculate, if desired. About 15 hours are required for full recharge.

After turning on the machine, clear the display by pressing **RESET**. The machine is ready for use - all storage and arithmetic registers, and both program memories are cleared.

Arithmetic operations are performed the same way that you write them on paper (**2** **+** **2** **=**). Be sure to end an arithmetic operation with **=**. Answers are displayed with zero to nine decimal places. Set the decimal point by pressing **SET DP** and the number of places that you want (**0** to **9**). All the examples in this supplement have the decimal set to four places (**SET DP** **4**).

PROGRAMMING

Scratchpad programming is essentially automatic key pressing. If you know how to solve a particular problem using the keyboard, you can then program the Micro Statistician to do the same problem for you automatically. Once your program is working, you are free to study and analyze several different sets of data without having to repeat long key sequences for each new data set.

EXAMPLE:

Suppose that you wish to find the harmonic mean M_H of the numbers 1, 2, 3, 6, 6, 7, 9, 12.

$$M_H = \frac{N}{\sum \frac{1}{X}}$$

where

N = the number of data items

X = a particular data item

Using the keyboard the solution works like this.

GROUP					See This
1	CLEAR GROUP	1	$\frac{1}{X}$	$\Sigma \frac{1}{X}$	1.0000
2		2	$\frac{1}{X}$	$\Sigma \frac{1}{X}$	0.5000
		3	$\frac{1}{X}$	$\Sigma \frac{1}{X}$	0.3333
		6	$\frac{1}{X}$	$\Sigma \frac{1}{X}$	0.1666
		6	$\frac{1}{X}$	$\Sigma \frac{1}{X}$	0.1666
		7	$\frac{1}{X}$	$\Sigma \frac{1}{X}$	0.1428
		9	$\frac{1}{X}$	$\Sigma \frac{1}{X}$	0.1111
		12	$\frac{1}{X}$	$\Sigma \frac{1}{X}$	0.0833

This operation gives $\Sigma \frac{1}{X}$ and N

Remember that:

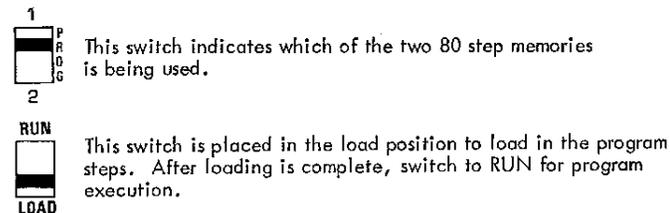
(N in Register 1)
 ΣX in Register 2
 ΣX^2 in Register 3)

	} Calculates M_H	8.0000
		$M_H = 3.1949$
	To clear the display	0.0000

The procedure isn't particularly difficult, but if it must be done for a large number of data or several sets of data, it could become quite tedious.

ENTER DOUBLE MEMORY SCRATCHPAD PROGRAMMING:

This same problem becomes quite easy with scratchpad programming. Let's program the same problem using both of the 80-step memories to show some basic programming procedures.



The number at the far right of the display tells you which step of the 80-step memory is ready for loading. On the left side of the display is the mantissa and between the mantissa and the step number is the two digit exponent to the mantissa. When in LOAD, all numbers are displayed in scientific notation. Your main concern is with the step numbers. The mantissa could be anything depending on what you have done prior to loading your program.

We will use Program 1 for the $\sum \frac{1}{X}$ portion of our example problem. The program steps do what we want with a minimum of confusion.

Step Number	Key	Display
1.		
2.		1.000000 00 02
3.		This stops the program to enter a data item 1.000000 00 03
4.		1.000000 00 04
5.		
6.		Reciprocal 1.000000 00 05
7.		Summation 1.000000 00 06

Program 1 is loaded. Notice that we are summing both the raw data as well as the reciprocals. Also, the change of the GROUP switch will be remembered by the 344, and it doesn't even take a step in memory. We are ready to load Program 2. Program 2 calculates the harmonic mean ($M_H = \frac{N}{\sum \frac{1}{X}}$) for the data entered with Program 1.



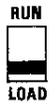
Step Number	Key	Display	
1.		} Puts N in the display 1.000000 00 02	
2.			1.000000 00 03
3.		} Calculates $\frac{1}{N} \sum \frac{1}{X}$ 1.000000 00 04	
4.			1.000000 00 05
5.			1.000000 00 06
6.		1.000000 00 07	
7.		Harmonic Mean 1.000000 00 08	

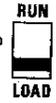
The second program is now loaded. Put the RUN/LOAD switch in the RUN position

To run the program, follow these steps:

1.    . The GROUP switch can be in either position.
2.  1.0000 displays
3. Enter X from the data 1, 2, 3, 6, 6, 7, 9, 12, touching  after each entry. After all the data has been entered, go to step 4.
4.   . The harmonic mean is displayed 3.1949
5. The arithmetic Mean and SD of the raw data can be obtained by   read SD (3.6936);  , read X (5.7500).
6. For a new problem, return to step 1.

The principles used in this program are applicable to any program that you wish to write. These are the basic rules.

- To load the program, switch to 
- Do the calculation sequence one time in  . After that, the Micro-Statistician repeats the sequence as many times as you need it.
- Use  at the places in your program where you want the program to stop for your data entry. During the loading sequence you may enter an input value immediately after a  . Sometimes you must enter a particular value after  to avoid an error condition which can occur with a divide by zero, reciprocal of zero, or \ln of zero. Also use  to display answers.
- Once the program is loaded, switch to  for execution.
- In general, press  to start the program and to take you to the point where the first input value or data item is to be entered.
- The GROUP switch position can be programmed, without even using a program step.

- When you switch from  to  the program will automatically loop from the last keystroke you entered, back to step 1.
- If you are in the middle of running a program and you want to get back to step 1, simply switch from  to  and then back to  . The program won't be changed but you will be back at step 1 again.
- Don't change the program switch  when in LOAD. It is nonfunctional in that mode.

Let's apply these basic principles to solve a more complex problem. Suppose that you are working with X and Y data pairs and you are looking for the curve that gives the best least squares fit to your data. It could be a linear fit but you're not sure. It could also be a power curve or an exponential curve. With the linear regression capability that is built into the Micro Statistician plus the two program memories, it is simple to do a least squares fit with each of the three curves. A comparison of the correlation coefficients tells you which curve gives the best fit. The equation of the curve that you want is also calculated.

The generalized formulas for the three curves are as follows:

Linear $Y = mX + b$

Power curve: $Y = bX^m$

Exponential curve: $Y = be^{mX}$

To perform a regression analysis for the power curve and the exponential curve all we do is make a transformation on the equations to put them in the linear form. The transformation for the power curve looks like this:

$$Y = bX^m \text{ transforms to } \ln Y = \ln b + m \ln X$$

The input values for Y and X are converted to $\ln X$ and $\ln Y$. The calculated value for m will not be transformed but to find b we must take the antilog_e of $\ln b$.

The transformation for the exponential curve is similar.

$$Y = be^{mX} \text{ transforms to } \ln Y = \ln b + mX$$

The data to be used in our analysis is as follows:

X	Y
1	1
2	4
3.8	8
7	9
9	10
14	13

The procedure for the keyboard linear regression is as follows:

1. **CLEAR GROUP** **XY**
2. Enter X **XY**
3. Enter Y **=**
4. Repeat steps 2 and 3 until all data pairs are entered
5. **LIN REG** displays the correlation coefficient (r)

For the example data r = 0.9299

6. **2ND FUNC** displays the slope (m)

For the example data m = 0.8218

7. **0** **LINE** displays the Y intercept (b)

For the example data b = 2.4594

The equation for the least squares line that gives the best fit for the example data:

$$Y = 0.8218X + 2.4594$$

8. For a new problem return to step 1.

Now let's write the program for the power curve regression using Memory I.



Loading Program I

- | | | | |
|------|----|-------------------|----------------------------|
| Step | 1. | 1 | Identification for X entry |
| | 2. | START STOP | Entry of X |
| | 3. | Ln LOG | Transformation of X |
| | 4. | XY | Paired data key (X) |
| | 5. | 2 | Identification for Y |
| | 6. | START STOP | Entry of Y |
| | 7. | Ln LOG | Transformation of Y |
| | 8. | = | Paired data key (Y) |

Display should show step 09 .



The second program for the regression on the exponential curve is loaded in Memory 2.

- 

1
P
R
O
G



RUN
LOAD
- Loading Program II
- Step
1.  Identification for X entry
 2.  Entry of X
 3.  Paired data key (X)
 4.  Identification for Y entry
 5.  Entry of Y
 6.  Transformation on Y
 7.  Paired data key (Y)
- Display should show step "08".



With both programs now in memory, we can compare the linear fit to the power curve fit and the exponential curve fit. Here's our data again:

X	Y
1	1
2	4
3.8	8
7	9
9	10
14	13

Running the Programs:

1.  for the power curve fit
2.  
3.  1.0000 displays
4. Enter X
5.  2.0000 displays
6. Enter Y
7. Repeat steps 3 through 6 until all data pairs are entered, then  and go to step 8.
8.  The correlation coefficient (r) is displayed
9.  m is displayed
10.    b is displayed
11. For the exponential curve fit, switch to program 2  start at step 2.
If you wish to run program 1 again, switch to  then switch from RUN to LOAD to RUN, and return to step 2.

Having performed the least squares fit for the three kinds of curves, we can compare the calculations and decide which curve gives the best fit.

Linear Fit	Power Curve Fit	Exponential Curve Fit
$r = 0.9299$	$r = 0.9347$	$r = 0.7821$
$m = 0.8218$	$m = 0.8926$	$m = 0.1511$
$b = 2.4594$	$b = 1.5594$	$b = 2.2884$
<u>Equation of the Line</u>	<u>Equation of the Line</u>	<u>Equation of the Line</u>
$Y = 0.8218X + 2.4594$	$Y = 1.5594X^{0.8926}$	$Y = 2.2884e^{0.1511X}$

The correlation coefficient for the exponential curve is small when compared to those for the linear and power curves. The high value for r in the linear fit is impressive but the power curve regression results in an even better fit.

TRICKS AND TECHNIQUES

We have already seen one useful trick in programming – to have a particular identification number display at each point in the program where a variable is to be entered. In some programs, you may not have enough room for the identification feature. In others it may be possible to have the identification numbers preceded by a  . This will help avoid confusing identification numbers with data.

Of course, you will want to change the decimal point to a larger setting at the points in your program where the output calculations are displayed.

Let's look at some other useful techniques. Suppose that you wish to have a feature in your program that allows you to increment or decrement a particular value each time the program is executed. This is easy enough, but at the same time you don't want the program to run forever, just a certain number of times. We will show how this can be done to solve Factorial of N .

Recall that Factorial of N is given by:

$$N! = N(N - 1)(N - 2) \dots 1$$

To program this we'll use register 1 for building the factorial and register 2 for the current value of N .

Here's the program:

Step	Operation	
1.		} Store the current value of N in register 1
2.		
3.		} Use direct register arithmetic here to multiply and store the current product
4.		
5.		
6.		
7.		} Direct register arithmetic for decrementing n by 1 on each run of the program
8.		
9.		
10.		} Test for n = 0. Error displays when n = 0, (1/0 = Error). Program execution stops.
11.		
12.		
13.		} The new value of n is in the display ready for the next pass through the program
14.		

To use the Program:

1.    Load the program as shown above.
2.  
3.    This puts an initial product of 1 in register 2
4. Enter N  (e.g., N = 5)
5. When E - - - - displays,   
N! is in the display (5! = 120)
6. For a new N! return to step 3.

Note that 69! is the largest factorial that can be calculated before the machine's capacity of $9.999999999999999 \times 10^{99}$ is exceeded.

The method used in the factorial program can be applied to other problems where incrementing or decrementing is used. The principal involved is to perform a test that causes E - - - to display when the program has been executed the desired number of times. Dividing by zero using the $\frac{1}{x}$ key is probably the most efficient method.

Many of the mathematical processes we use don't ever give zero results, and thus the divide by zero technique is not sufficient. 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.

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

CHR SIGN	2	ST _n	1
3	ST _n	2	
4	ST _n	3	

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

Step	Operation	Step	Operation
1.	RCL _n	9.	×
2.	1	10.	RCL _n
3.	×	11.	9
4.	RCL _n	12.	+
5.	9	13.	RCL _n
6.	+	14.	3
7.	RCL _n	15.	=
8.	2		

aX (steps 1-3)
 $aX + b$ (steps 4-6)
 $aX^2 + bX$ (steps 9-11)
 $aX^2 + bX + c$ (steps 12-15)

Thus far we have evaluated the equation. The next steps in our program will find the maximum value of the function.

Step	Operation	
16.	EXCH _n	} Puts the new value of X in register 8 and the previous value of X in register 7
17.	8	
18.	ST _n	
19.	7	} Compares the new value to the old value and leaves the difference in the display
20.	RCL _n	
21.	8	
22.	RCL _n	} This is the test. As long as each new value is greater than the old value, the difference between the two is positive. When the difference becomes negative, the maximum has been passed, E - - - displays because taking square roots of negative numbers is an illegal operation. The Micro Statistician stops program execution.
23.	-	
24.	7	
25.	√	
26.	RCL _n	} These last steps increment the current value of X by the value stored in register zero.
27.	0	
28.	ST _n	
29.	+	} The program is loaded.
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 $X = -5$.
3. Store the increment to X in register 0. Use 0.25 for the increment.
4. **START STOP** The program loops until E - - - is displayed.
5. **RESET** **RCL_n** **9**
6. Your starting value for X is displayed, press **START STOP**.
7. The program will loop again until E - - - is displayed.
8. **RESET** **RCL_n** **7** The computed maximum value for X is displayed.
9. **RCL_n** **9** **-** **RCL_n** **0** **=** Displays the X value that corresponds to the compute X_{\max} (5.1250).
Of course, what you have is the nearest value to X_{\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.

Many of our customers have developed their own techniques for wringing the maximum power from their Micro Statisticians. You should experiment with different ideas on your problems. Since you know your needs best, if you come up with something great, let us know. There's a handy card in the back for you to use in explaining your unique solution.

BATTERIES AND RECHARGING

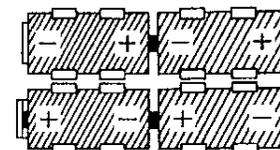
The Micro Statistician operates about 3-1/2 hours when not connected to an AC power source. You can connect it to the AC charger, supplied with the Micro Statistician -- the charger plugs into any standard 110 volt wall outlet.

The nickel cadmium batteries supplied are rechargeable. Useful life is approximately 1000 charge/discharge cycles. When fully discharged, the batteries require about 15 hours to recharge with the Micro Statistician turned off. When turned on and operating, recharging takes somewhat longer.

To remove the batteries, turn the Micro Statistician 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, Compucorp Part Number 3400017 are recommended.

NOTE

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



The AC adapter always supplies charging current to the batteries, whether the Micro Statistician 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 Statistician, or the batteries may explode.

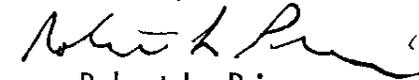
NOTES

Please use this handy reply card for any comments that you have about your Micro-Statistician or this manual. We are particularly interested in learning about customer applications, and problems.

Are there any programs that you have written or would like to see written? If you have a program that you've found useful, tell us about it. We would like to make it available to others.

We are developing a library of Micro-Statistician programs and useful additions are always welcome.

Thank you,



Robert L. Price
Product Manager

Name _____ Position _____

Company _____

Company Address _____

City _____ State _____ Zip _____

Phone Number _____

Micro-Statistician Model 340 ___ 342 ___ 344 ___

Serial Number _____

Comments _____

BUSINESS REPLY MAIL

FIRST CLASS
Permit No. 53488
Los Angeles, Calif.

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by—

Computer Design Corporation
12401 West Olympic Boulevard
Los Angeles, California 90064

