

600
FORMULA
PROGRAMMING
(FP) ROM
REFERENCE
MANUAL

M-Labor

WANG

600 FORMULA PROGRAMMING (FP) ROM REFERENCE MANUAL

© Wang Laboratories, Inc., 1972



LABORATORIES, INC.

836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL. (617) 851-7311, TWX 710 343-6769, TELEX 94-7421

Congratulations! Your decision to purchase a Formula ROM for your Model 600 Calculator was a wise move. You'll find that the Formula ROM adds a new problem-solving ease to the most powerful calculator on the market — The Model 600. No other calculator offers you the combined problem-solving ease of an algebraic language, together with the proven efficiency of the 600's calculator language and standard keyboard functions.

With the Formula ROM, you enter equations and evaluate algebraic expressions as easily as you write them. This is because the Formula ROM allows you to enter expressions in true algebraic notation, including parentheses, exponentials, and two-dimensional subscripting. As a result, you don't have to rewrite your expressions to conform to the calculator.

By using subscripting, you can work with matrix elements as easily as you work with regular variables. And, if you add a 618 Extended Memory Unit to the System, the 600 Calculator-Formula ROM combination will enable you to use as many as five separate matrices. These can be entered either manually via the 600 Keyboard, or under program control. After all the elements have been entered, you can then easily manipulate the elements of each matrix. All you have to do is specify the subscript of the particular "element" you want of the matrix, identifying it. You can then use it in your calculations just as you would if you were using the FORTRAN or BASIC language. The Model 600 with Formula ROM is the only calculator on the market with this capability.

Programming with the Formula ROM is just as easy. You'll find yourself solving complex equations, operating on matrices, and setting up "looping" programs faster than you imagined: All because of the Formula ROM. And, if you have a writer such as the Model 601, 602, or 611, the Formula ROM will give you a printout of your expression either as you enter it from the keyboard, or execute it under program control.

We are sure that you will find your investment in the Model 600 with Formula ROM was well made.

"WE DO MORE, SO THAT YOU CAN"

WANG LABORATORIES, INC.
TEWKSBURY, MASS.

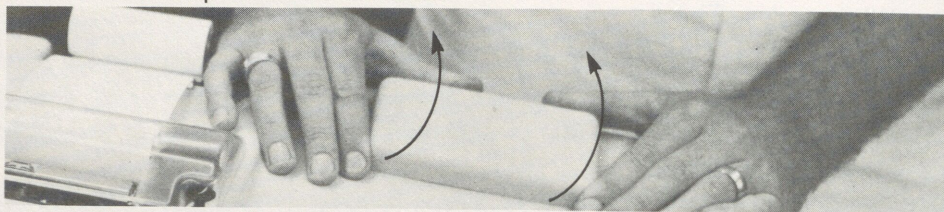
TABLE OF CONTENTS

	PAGE
Installation	vii
The New Special Function Keys	1
Solving Simple Numeric Expressions	1
Parentheses and When to Use Them	2
Balancing Parentheses and the 03 ERROR Indicator	3
How Many Parentheses Can Be Used?	4
The Error Indicators and What They Tell You	5
Solving Complex Numeric Expressions	6
The Algebraic Constants	7
Defining and Using the Constants	7
Programming the Model 600 Calculator with the Formula ROM	11
The NEXT Command — A Programming Aid in Looping	13
Subscripts, Matrices, and Vectors	19
What is a Matrix?	19
Matrix Operations With the Formula ROM	20
Entering a Matrix into the Calculator	20
R & S Matrix Size	21
Checking the Elements of a Matrix in Memory	22
Using the Matrix OUT PAUSE Feature	22
Using the Matrix OUT Feature	23
The U(, V(, and W(Keys and the Optional Model 618 Extended Memory	24
Defining the U, V, and W Matrices	25
Addressing the Extended Memory	25
Manipulating the Elements of the U, V, and W Matrices	25
Programming with Matrix Operations	26
APPENDIX A — Additional Notes and Hints	27
How to Handle Large Expressions with Many Nested Parentheses	27
Sharing the Keyboard Registers with the Formula ROM	27
APPENDIX B — Subroutine Levels and the Formula ROM	29
APPENDIX C — Optional Column Printer Codes	30

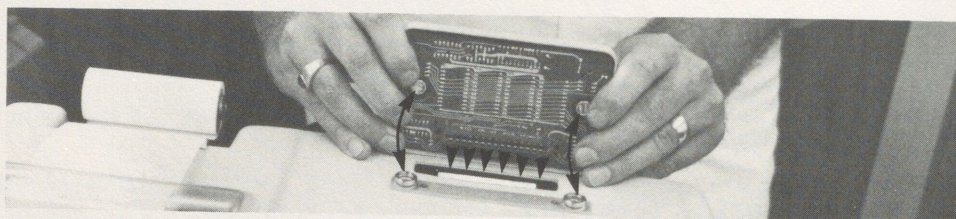
INSTALLATION

To use your new Formula ROM,

- 1.) With the Calculator Power switch in the OFF position, unsnap the protective cover on top of the Model 600.



- 2.) Position the Formula ROM over the connector snaps so that when the Formula ROM is snapped into place, the "connector tongue" of the calculator will fit into the receptacle on the bottom side of the Formula ROM.



- 3.) Now, snap the Formula ROM into position.
- 4.) Remove the 16 Special Register keytops, and replace them with the new set of keytops provided. This new set of keytops will show you which key is used for each algebraic function.

PRINT BEGIN	(	)	,	OUT PAUSE	+	-	*	/	IN	OUT	NEXT	DEFINE	R(	S(	U(	V(	W(
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15		

- 5.) Depress the POWER ON Switch on the back of the calculator, and the (X) and (Sp) Selector Switches on the calculator keyboard. Your Formula ROM is now ready to use.

NOTE:

In order to utilize the 600 and Formula ROM to their full capacity, it is suggested that the first four chapters (pages 1 - 27) of the 600 Manual be reviewed. This material will serve as a general introduction to the 600 calculator as well as enable you to familiarize yourself with all of the keys and switches you will use in solving your problems.

THE NEW SPECIAL FUNCTION KEYS

PRINT BEGIN	(	)	+	OUT PAUSE —	*	/	IN ↑	OUT →	NEXT I	DEFINE J	R(K	S(L	U(X	V(Y	W(Z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15

BEG	"Do The Following"	Used whenever you begin a new formula or equation	*	"multiply"
(	left parenthesis		/	"divided by"
)	right parenthesis		↑	"raised to the power"
+	"plus"		→	"equals" or "becomes" —
—	"minus"			(used to obtain results and/or define the constants)

NOTE:

To use these functions: 1) be sure that both the (X) and (Sp) Selector switches are down; 2) depress the CLEAR switch or store the number 25 in Register 04; and 3) key **PRIME** and **00** before entering an expression.

When the power to the calculator is first turned on, it is suggested that the red CLEAR switch be pressed, regardless.

SOLVING SIMPLE NUMERIC EXPRESSIONS

EXAMPLE 1:

Solve the expression $“(3 + 4)^2 + 5 \times (7 - 6)”$ using the Formula ROM.
(Key **PRIME**, and depress the CLEAR switch if you have just turned the calculator on.)
The keystrokes would be:

BEG (3 + 4) ↑ 2 + 5 * (7 — 6) → The display will show **54.00000000** .

See how easy it is?

NOTE:

If you have an output device such as the Model 601, 602, or 611, you can obtain a printout of your formula as you key it in; if, before you key in the expression, you key **SHIFT**, **00** (**PRINT**), instead of just keying **00** (**BEG**). The output writer will then print out the entire formula (with numeric values), until you press the $\rightarrow$ symbol. At that point, the equal sign (=) and the result of the formula will be printed out.

EXAMPLE 2:

Solve the expression $\sqrt[3]{17.3} / 2.27^2 + \sqrt{(15.85 - 9.36)}$ using the Formula ROM, and print out the expression as you key it in.

First, we would rewrite the formula something like this:

$$17.3^{1/3} / 2.27^2 + (15.85 - 9.36)^{1/2}$$

Now, key **PRIME**. The keystrokes would be:

PRINT 17.3 $\uparrow$ (1 / 3) / 2.27 $\uparrow$ 2 + (15.85 - 9.36) $\uparrow$.5 $\rightarrow$ (Read **3.049463008** in the display.)

The output writer would have printed the following:

17.3 ** (1 / 3) / 2.27 ** 2 + (15.85 - 9.36) ** .5 = 3.049463008

You now have a permanent copy of your expression, with numeric values and answer.

PARENTHESES, AND WHEN TO USE THEM

As the two examples above have shown, parentheses are used to group numbers and operators together, so that the proper result is obtained.

If, in Example 1, we had not used the parentheses in "5 x (7 - 6)" which equals 5, the expression would have looked something like this:

"5 x 7 - 6", and would equal 29 instead of 5.

This is because the *order of execution* used by the Formula ROM is the same as the algebraic order of execution:

Operation	Order of Execution
Power ($\uparrow$)	1st
Multiplication (*) and Division (/)	2nd
Addition (+) and Subtraction (-)	3rd

By using the parentheses, we are assured of having an expression executed the way we want it to be.

EXAMPLE 3:

Evaluate the following expression, using the Formula ROM;
(Be sure to put the parentheses in where indicated.)

$$((3 + 4)^{1/2} - 15) / 17 + 4^{3/2}$$

KEYSTROKES:

PRINT ((3 + 4) ↑ .5 - 15) / 17 + 4 ↑ (3 / 2) →

(read 7.273279488 in the display)

The output writer would print the following:

$$((3 + 4) ** .5 - 15) / 17 + 4 ** (3 / 2) = 7.273279488$$

Now, evaluate the same expression, but leave out *all* parentheses:

$$\text{That is, evaluate } 3 + 4^{1/2} - 15 / 17 + 4^{3/2}$$

KEYSTROKES:

PRINT 3 + 4 ↑ .5 - 15 / 17 + 4 ↑ 3 / 2 → (read 36.117647059 in the display)

The output writer would print the following:

$$3 + 4 ** .5 - 15 / 17 + 4 ** 3 / 2 = 36.117647059$$

Can you see how important the parentheses are?

BALANCING PARENTHESES AND THE 03 ERROR INDICATOR

You've seen the importance of parentheses and how they are used to group numbers and algebraic operations. There is another important point about parentheses which should be mentioned. Whenever parentheses are used with the Formula ROM, there should always be as many right-hand parentheses as left-hand parentheses — just as in algebra. This is called balanced parentheses. If parentheses are not balanced, the calculator display will tell you so, by displaying a flashing 03 in the display.

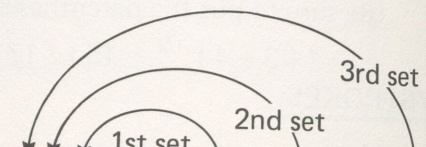
This condition is caused by either of the two following situations:

1. Too many left-hand parentheses (the display will flash after you press "→").
2. Too many right-hand parentheses (the display will flash as soon as you press the extra right parenthesis).

In either case, you should check your formula, correct the problem, press **BEG**, and start over.

HOW MANY PARENTHESES CAN BE USED?

In Example 3, you were asked to key in an expression with three pairs of parentheses, one pair of which was *nested* inside of another. The expression below shows three sets of parentheses, all nested:



$(\sqrt{ (7.3 + 4.2)^2 + 6 } + 17) / 22$ would be rewritten as $(((7.3 + 4.2)^2 + 6)^{1/2} + 17) / 22$

(When keying in the expression, it would be important to make sure that all the parentheses were entered.)

The 600 Formula ROM allows you to nest up to 99 sets of parentheses, even though few problems require more than 10 sets. This assures that you can solve virtually any problem.

THE ERROR INDICATORS AND WHAT THEY TELL YOU

Just as the Formula ROM tells you when you have entered an expression with unbalanced parentheses, the Formula ROM also tells you when you perform some illegal operation. As with the case of unbalanced parentheses (03), the display will flash on and off if any of the following occur:

Number Displayed	Meaning	Examples
+01	Expression too big	Expression entered is too large for ROM to evaluate. See page 27 "How to Handle Large Expressions."
+02	Expect operator	Keying 3 I — implied multiplication is not allowed.
+03	Unbalanced parentheses	Too few or too many right parentheses.
+04	Invalid operator	6 (3 + 8) should be entered as 6 * (3 + 8); or using the "," (comma) without using R(, S(, U(, V(, or W(.

NOTES:

Flashing 01

If the expression entered is larger than the Formula ROM can handle, rewrite and re-enter. One way of doing this is to evaluate part of the expression, and then store it as constant I, J, K, L, X, Y or Z. See page 7, "Defining and Using the Constants", as well as Appendix A, "How to Handle Large Expressions" for an explanation.

Flashing 02

3 I is not allowed.

As soon as the I is keyed in, the display will flash an 02 indicating that the user is trying to perform implied multiplication. An operator should have been inserted between the 3 and the I. Also, if the operator failed to key BEGIN or PRINT at the onset of keying in an expression.

Flashing 03

See page 3, "Balancing Parentheses", for an explanation.

Flashing 04

The display will flash 04 if a math operator (+, -, x, ÷) has been omitted between and number and a left parenthesis. This indicates that the left parenthesis was an invalid operator. For example: 6 (3 + 8) would cause the display to flash 04 because a "x" (multiply sign) operator was omitted between the 6 and the left parenthesis.

Flashing Anything Else

If the display flashes anything else, a program or mathematical error is indicated.

SOLVING COMPLEX NUMERIC EXPRESSIONS

EXAMPLE 4:

Solve the expression $\sqrt{5.2^2 + 6.1 \sqrt{74.38 + 7.62}} / 2.7^2$ and print out the formula as you key it in.

First, you would rewrite the formula something like this:

$$(5.1^2 + 6.1 * (74.38 + 7.62)^.5)^.5 / 2.7^2$$

Now, key PRIME

KEYSTROKES:

PRINT (5.1 ↑ 2 + 6.1 * (74.38 + 7.62) ↑ .5) ↑ .5 / 2.7 ↑ 2 →
(Read **1.236455266** in the display)

The output writer would type the following:

(5.1 ** 2 + 6.1 * (74.38 + 7.62) ** .5) ** .5 / 2.7 ** 2 = 1.236455266

EXAMPLE 5:

Sometimes you will need to use a combination of the ROM, and regular keyboard functions.

Solve the expression

$$\frac{\sin^4(.994)}{\sqrt{\cos^3 8.1 + \sin 4.3}}$$

and print out the formula as you key it in.

First, rewrite the expression something like this:

$$\sin^4(.994) / (\cos^3 8.1 + \sin 4.3)^{1/2}$$

Now, key PRIME,

KEYSTROKES:

PRINT .994 SIN ↑ 4 / ((8.1 COS) ↑ 3 + 4.3 SIN ↑ .5 →
on keyboard on keyboard on keyboard (Read **8.8580385889-08** in display)

(The SINE and COSINE functions are generated on the keyboard by keying the number, and then keying SIN, COS, etc.)

The output writer would then type the following:

.017347702 ** 4 / ((.990023657) ** 3 + .074978726) ** .5 = 8.8580385889-08

NOTE:

The DEG/RAD switch must be in the UP position in this example.

THE ALGEBRAIC CONSTANTS

Suppose you have an expression like the following which you want to evaluate for $X = 3.72$, $Y = 1.43$:

$$3X^2 - 4XY + 5Y^2$$

You could save a lot of effort by first assigning the values 3.76 and 1.42 to the X and Y constants respectively. Then, you could evaluate the expression just as it stands, by keying the X and Y constants in the appropriate sequence.

PRINT BEGIN	(	)	,	+	OUT PAUSE —	*	/	IN ↑	OUT →	NEXT I	DEFINE J	R(K	S(L	U(X	V(Y	W(Z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	

DEFINING AND USING THE CONSTANTS

As you can see, the ROM Function key caps have seven letters on them associated with register keys **09** through **15**.

Each of these letters can be used to represent a constant, or an expression.

For example, in our above problem, we would do the following:

BEG 3.72 → X 1.43 → Y

This would enter the values we need for X and Y. We would then evaluate the expression, $3X^2 - 4XY + 5Y^2$, by keying X and Y as needed:

KEYSTROKES:

BEG 3 * X ↑ 2 - 4 * X * Y + 5 * Y ↑ 2 → (Read **30.46129999** in the display)

Similarly, suppose you have an expression like the following:

$$[\sin^2(30) + 4] + 5\sqrt{\sin^2(30) + 4} - \frac{1}{(\sin^2(30) + 4)^3}$$

The expression $(\sin^2(30)+4)$ appears three times in the formula. To save time and effort, the Formula ROM enables you to enter this expression once, and then use it many times as you need to. This is accomplished by using the constant keys.

That is, if we set $\sin^2(30) + 4 = X$, our formula simply becomes:

$$X + 5\sqrt{X} - 1/X^3, \text{ or } X + 5X^{1/2} - 1/X^3$$

To find the answer to this expression, we would first find the result of $\sin^2(30)+4$, enter the value into the calculator as X, and then evaluate the expression $X+5X^{1/2}-1/X^3$.

Therefore, key PRIME,

KEYSTROKES:

BEG 30 SIN ↑ 2 + 4 → (Read **4.250000000** in the display)
on keyboard

Now, to enter this result as X, simply depress 13 .

To finish the problem, (Do not depress the CLEAR Switch):

KEYSTROKES:

BEG X + 5 * X ↑ .5 - 1 / X ↑ 3 → (Read **14.54473740** in the display)
Simple, isn't it?

EXAMPLE 6:

Using the Formula ROM on your Model 600, solve the following formula,

$$\frac{17.32}{\sqrt{(e^{3.4} + 7 \tan 23.6)}} - 3(e^{3.4} + 7 \tan 23.6)^2 + \log_e 5 (e^{3.4} + 7 \tan 23.6)^{1/3}$$

by first assigning the expression " $e^{3.4} + 7 \tan 23.6$ " to the ROM constant K.

Substituting $K = (e^{3.4} + \tan 23.6)$ we have

$$\frac{17.32}{\sqrt{K}} - 3K^2 + \log_e 5 K^{1/3} \text{ or } 17.32 / K^{1/2} - 3K^2 + \log_e 5 K^{1/3}$$

Key PRIME, and

KEYSTROKES:

PRINT 3.4 e^x + 7 * 23.6 TAN → K **33.02232485** in the display
on keyboard on keyboard

The expression has now been evaluated and entered as K.

The printout would be:

$$29.964100047 + 7 * .436889257 = 33.02232485$$

Now,

KEYSTROKES:

PRINT 17.32 / K ↑ .5 - 3 * K ↑ 2 + 5 log_eX * K ↑ (1 / 3) →

 on keyboard

(read **-- 3263.244318** in the display)

The output writer would type the following:

17.32 / 33.02232485 ** .5 - 3 * 33.02232485 ** 2

$$+ 1.609437912 * 33.02232485 ** (1 / 3) = -3263.24431832$$

NOTES

PROGRAMMING THE MODEL 600 CALCULATOR WITH THE FORMULA ROM

NOTE:

In order to familiarize yourself with the techniques used in programming the Model 600 calculator, chapters 6 through 9 of the 600 Manual should be reviewed. This review will enable you to better utilize the Formula ROM for programming your problem solutions.

As you have seen, using the Formula ROM to solve formulas makes problem-solving a snap. You will find this same high-powered problem-solving capability invaluable when programming the 600 calculator as well. The reasons are easy to see. This example will show you some of them.

EXAMPLE 7:

Using the Formula ROM, write a program to solve the following equation for K, given any set of values for X and Y:

$$K = (4 * X + 12 * X * Y + 9 * Y)^2$$

Operating Instructions (These tell you how to instruct the calculator to execute the program you have entered into the memory area. See page 60 of the 600 Manual for a fuller explanation.)

In RUN mode, PRIME, depress CLEAR Switch.

BEG (Enter X Value into the display) → X (Enter Y Value into the display) → Y
Key GO. (The answer appears in the display.)

— Keystrokes entered into calculator in LEARN mode — (This is the actual program.)

BEG (4 * X + 12 * X * Y + 9 * Y) ↑ 2 → END PROG

NOTE:

All underlined steps are regular calculator keystrokes; all numbers are entered via the numeric keyboard.

As you can see, programming with the Formula ROM is as easy as using the Formula ROM directly from the keyboard.

EXAMPLE 8:

Using the Formula ROM, write a program to calculate the real roots of the quadratic equation:

$$i x^2 + j x + \kappa = 0, \text{ and print out the values with three decimal places.}$$

That is, solve for the two X values

$$\frac{-j + \sqrt{j^2 - 4 * i * \kappa}}{2 * i} \quad \text{and} \quad \frac{-j - \sqrt{j^2 - 4 * i * \kappa}}{2 * i}$$

where the set of values i, j, and κ are keyed into the program.

Operating Instructions: (Column Printer "ON") — in RUN mode — (Again, these instructions tell you how to execute the program you have entered into the memory area of the calculator.),
PRIME, depress the CLEAR Switch, key GO,
key in "i" value, key GO, key "j" value, key GO,
key in " κ " value, key GO.

—KEYSTROKES entered into the calculator in the LEARN mode —

BEG STOP → I STOP → J STOP → K } Entering values for i, j, and κ

BEG (- J + (J ↑ 2 - 4 * I * K) ↑ .5) / (2 * I) → } calculates
the first
value

WRITE 0003 — Prints the first value to 3 decimal places

BEG - J - (J ↑ 2 - 4 * I * K) ↑ .5) / (2 * I) → } calculates
the first
value

WRITE 0003 — Prints the second value to 3 decimal places

END PROG

Notice how easy it is to follow the program flow. The Formula ROM enables you to key in a program in the same logical order you would use in writing the problem on a piece of paper.

THE NEXT COMMAND — A PROGRAMMING AID IN LOOPING

As you are well aware, the power of a programmable calculator lies in its ability to give fast, accurate results to problems entered into its memory area. Quite often these problems require the calculator to evaluate a particular expression or routine over and over again. Rather than enter the expression into the calculator a number of times, it's easier just to enter it once, and just "loop" through it as many times as necessary.

To control the number of times you loop, a register must be set aside as a counter. Each time you loop, a number would be either added to or subtracted from the counter register, after which a decision would be made as to whether or not another loop would be required.

The Formula ROM simplifies this procedure, through the use of the NEXT Command and one of the seven pre-defined constants, I, J, K, L, X, Y, or Z. All that is necessary is to enter the initial and final values of the counter in one of the constant registers. Each time the NEXT command is encountered in the program, the value in the counter register will increase by 1, until the final value is obtained, after which the contents of the counter are set to zero.

NOTE:

To generate the NEXT command, key **SHIFT**, **09**.

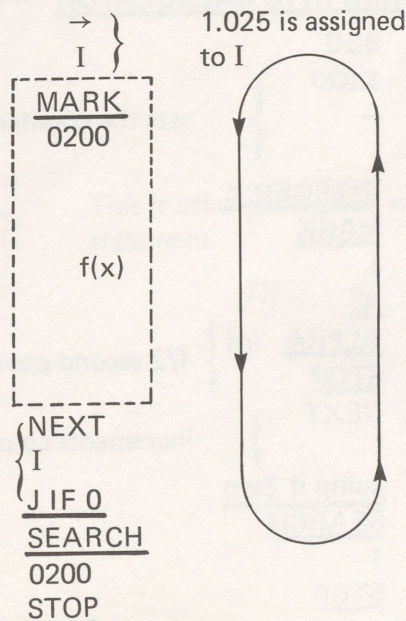
Consider the following diagram, and the accompanying description:

Suppose it was decided that the I Register would be used as the counter, and you wanted to loop 25 times. You would enter the number 1.025 → I. You would then put in the set of instructions you wanted to loop through, and end the instruction with **NEXT I**, and a **Jump if 0** test.

"1" indicates that the count is to start at 1.

→ 1
.
0 } these three digits
2 } indicate the last
5 } value of the count

This says increment the integer portion of I by 1. If display is zero, Jump 2 steps.



The two-step NEXT I command will cause the integer portion of the content of the I Register, as well as the display to be incremented by 1 each time this two-step command is encountered. The **JIF 0** test (Jump two steps if the display shows zero) will check the content of the display to see if it is zero. If the content of the display is not zero, the program will execute the next two program steps, which in this case say to **SEARCH** for **MARK 0200**, the beginning of the loop again. After the NEXT I command has been encountered twenty-five times, the content of the display will automatically be set to zero. Therefore, the **JIF 0** command will cause the program to jump out of the looping procedure, and continue executing the rest of the program. In this case, the program would stop.

EXAMPLE 9:

Write a program using the Formula ROM with a loop and the NEXT command to square the numbers 1, 2, . . . 20 and display each for a half second.

To satisfy the requirements of this example, several things will have to be done. First, we will need to set up a "counter" (let's use I) which can be used to keep track of the number of times we loop through the program. In this case, because we want to square all the integers from 1 through 20, we would store the number 1.020 as the constant I. We will then need a set of program steps we can loop through which will square the number and display it for half a second. Lastly, we will need a set of steps to increment the I constant and check the content of the display for a zero value. If the content of the display is not yet zero (i.e., if we haven't looped for all the 20 integers), we will want to execute the "loop" again. Therefore, we will use a **SEARCH** command to bring the program back through the loop again. Once we have looped the proper number of times, we will want the program to stop.

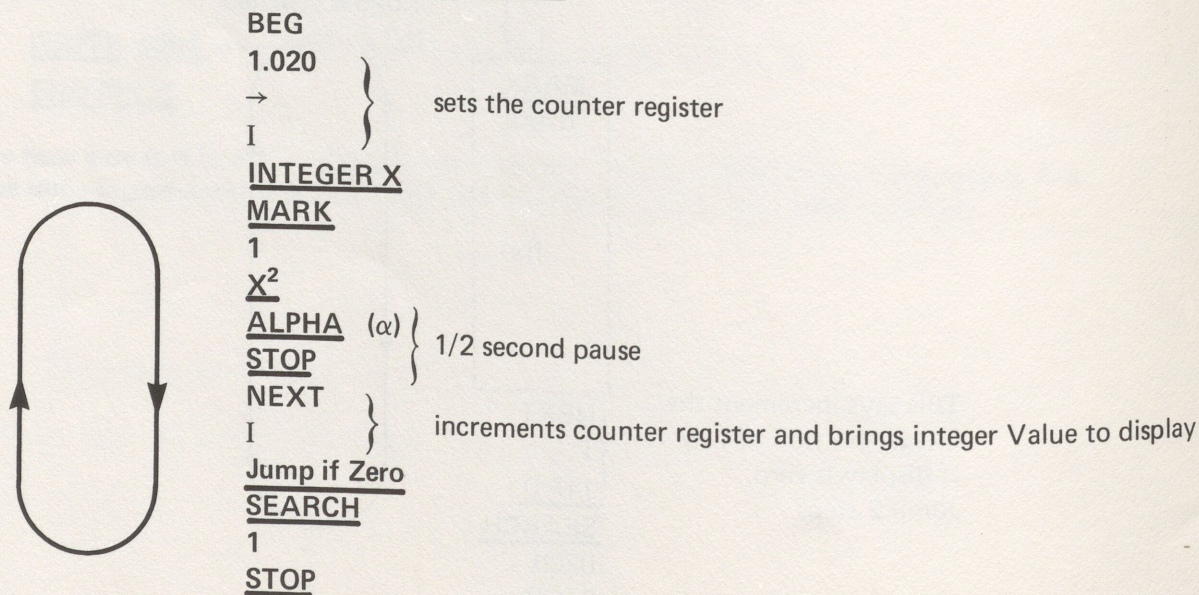
The flow diagram on the following page represents the program just described.

The program needed is as follows:

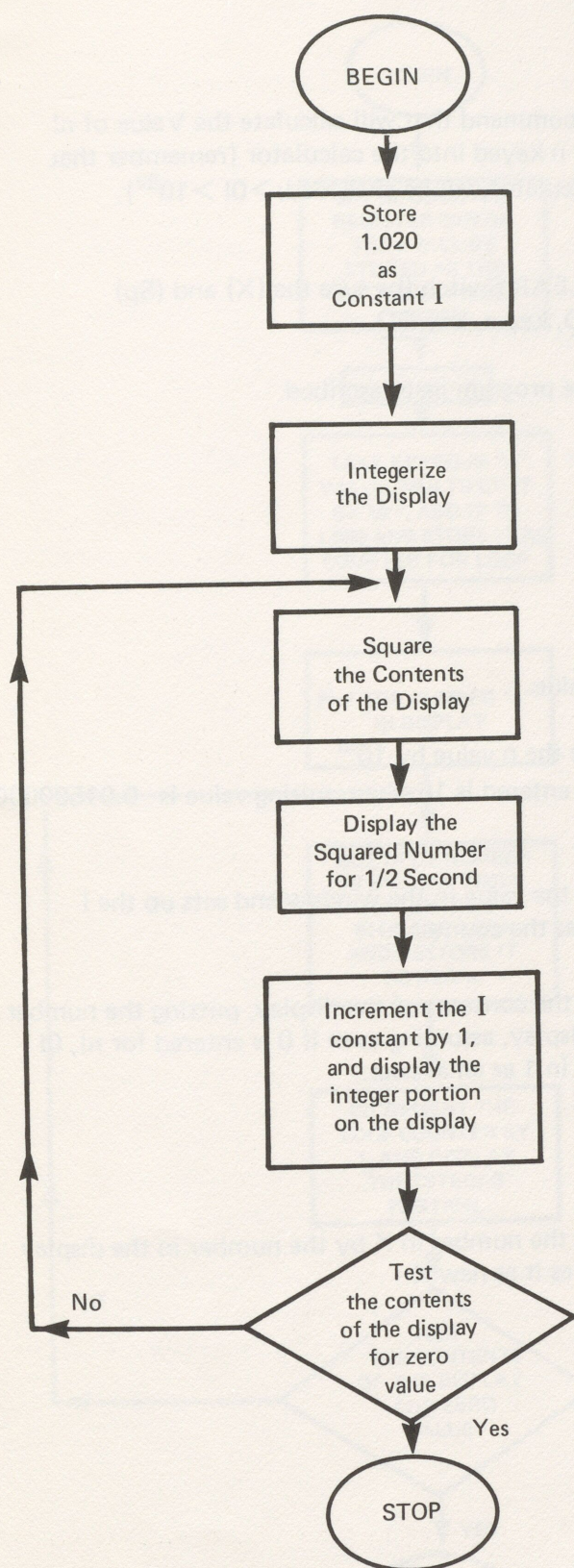
Operating Instructions: — In RUN Mode —

Key **PRIME**, depress the CLEAR Switch, (Be sure the (X) and (Sp) Switches are down), key **GO**.

KEYSTROKES ENTERED INTO CALCULATOR — IN LEARN MODE —



THIS IS THE ASSOCIATED FLOW DIAGRAM



This will change the contents of the display from 1.020 to 1, thus generating the first number in our program for us, without looping.

All this is accomplished by using the NEXT I command.

This is accomplished by using a Jump if Zero decision statement.

EXAMPLE 10:

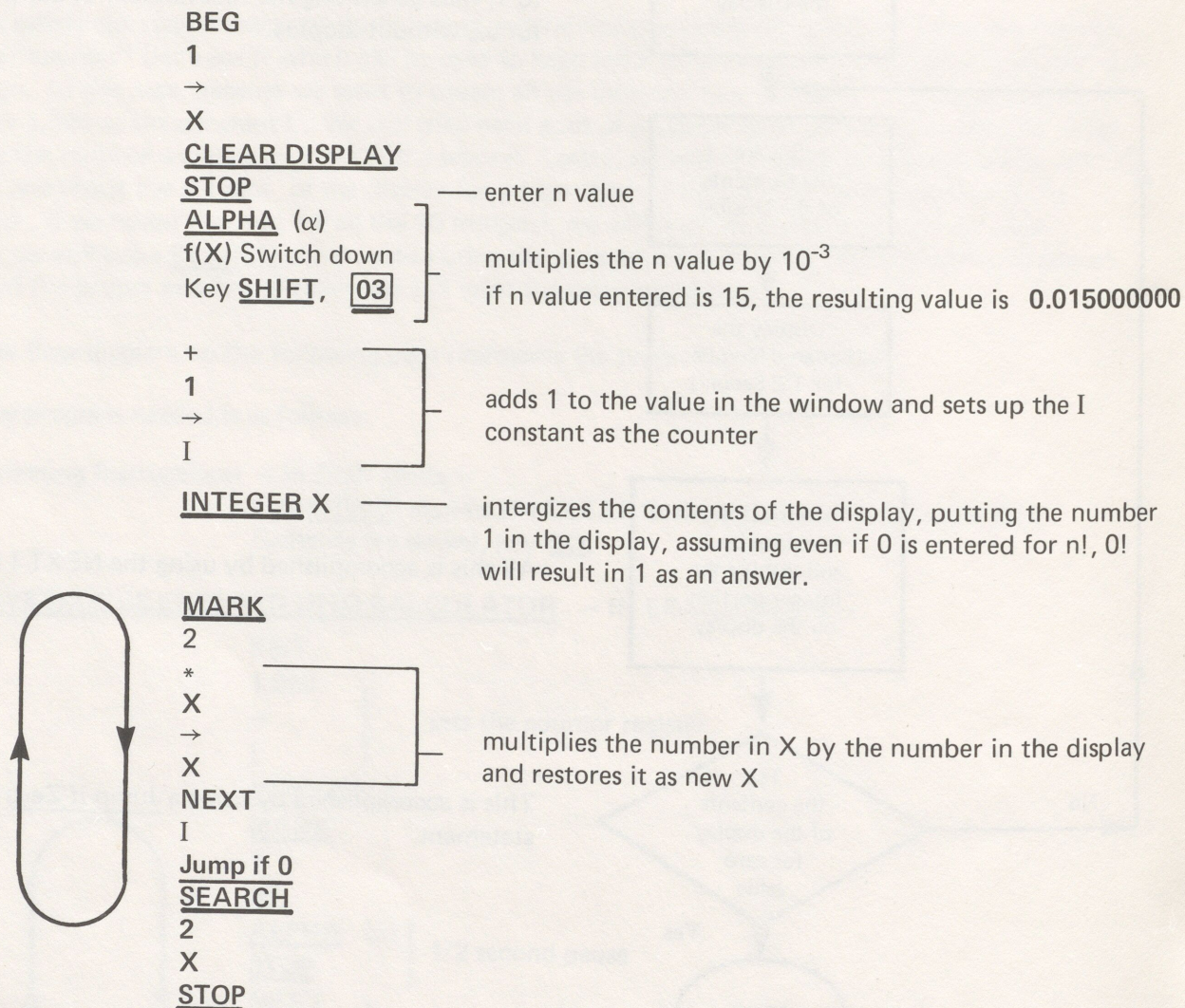
Write a program using the Formula ROM and NEXT command that will calculate the Value of $n!$ (i.e., $n \times (n-1) \times (n-2) \dots 3 \times 2 \times 1$) for any integer n keyed into the calculator (remember that $0! = 1$, and n must be less than >0 , or the capacity of the calculator will be exceeded; $>0! > 10^{99}$).

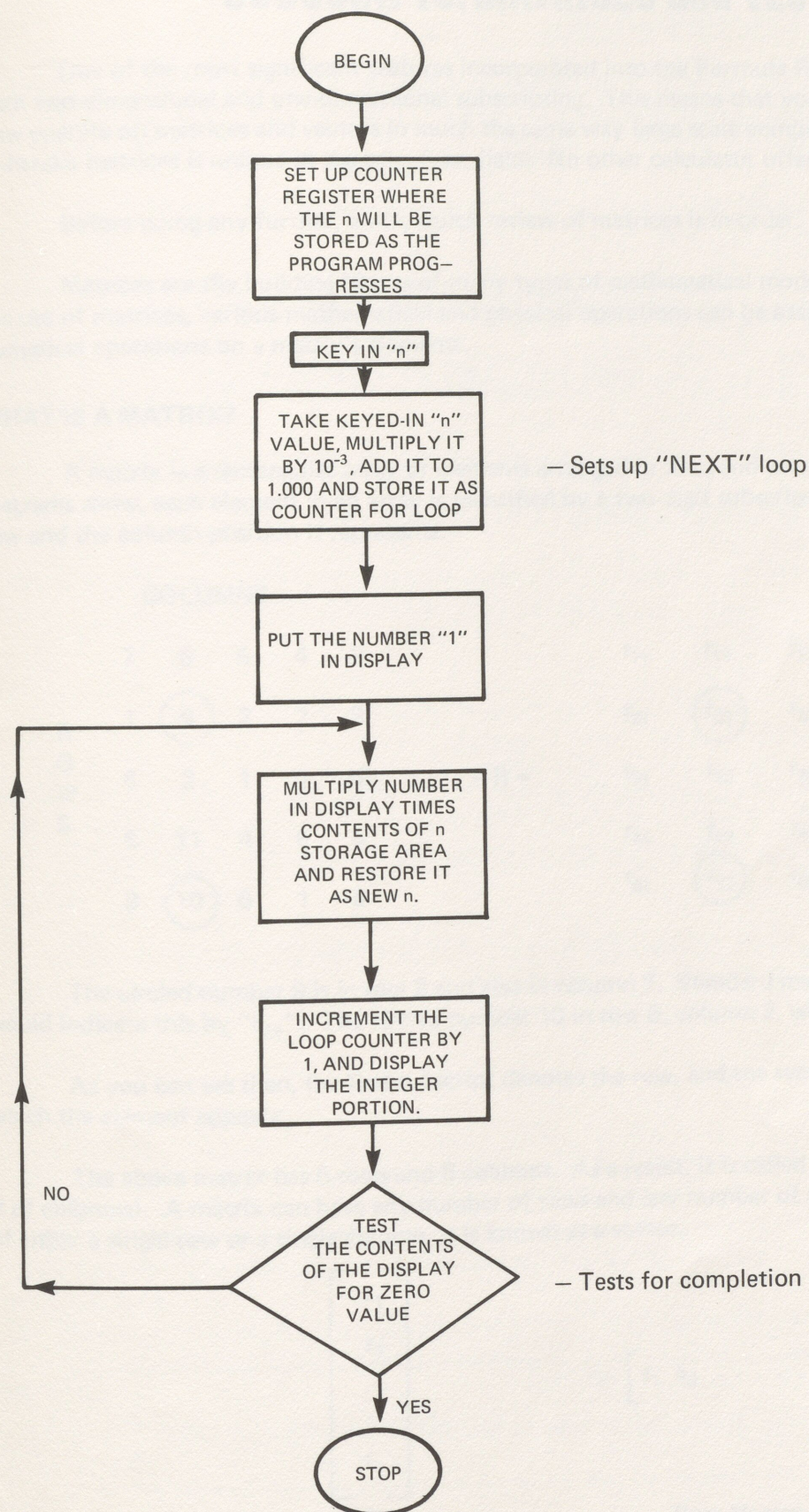
Operating Instructions: — In RUN Mode —

Key PRIME, depress the CLEAR Switch (be sure the (X) and (Sp) Switches are down), key GO, key n , key GO.

The flow diagram on the following page represents the program just described.

— KEYSTROKES Entered into calculator in LEARN mode —





SUBSCRIPTS, MATRICES and VECTORS

One of the most significant features incorporated into the Formula ROM is its ability to handle both two-dimensional and one-dimensional subscripting. This means that your Model 600 calculator can now operate on matrices and vectors in much the same way large scale computer systems do. This ability to handle matrices is unique in the calculator field: No other calculator offers this capability.

Before going any further, a very quick review of matrices is in order.

Matrices are the building blocks of many types of mathematical models. The reason is that through the use of matrices, various mathematical and physical operations can be easily represented in terms of numerical operations on a matrix's elements.

WHAT IS A MATRIX?

A matrix is a rectangular array of elements arranged in rows and columns. As the following diagrams show, each element in an array is identified by a two-digit subscript that is a function of the row and the column position it represents.

COLUMNS											
	7	6	5	4	9		r_{11}	r_{12}	r_{13}	r_{14}	r_{15}
R O W S	1	9	2	2	3		r_{21}	r_{22}	r_{23}	r_{24}	r_{25}
	6	2	1	4	10	= R =	r_{31}	r_{32}	r_{33}	r_{34}	r_{35}
	5	11	4	9	5		r_{41}	r_{42}	r_{43}	r_{44}	r_{45}
	9	10	6	1	5		r_{51}	r_{52}	r_{53}	r_{54}	r_{55}

The circled number 9 is in row 2 and also in column 2. Standard mathematical subscript notation would indicate this by " r_{22} ". The circled number 10 in row 5, column 2, would be identified as r_{52} .

As you can see then, the first subscript denotes the row, and the second subscript the column, in which the element appears.

The above matrix has 5 rows and 5 columns. As a result, it is called a square matrix (# of rows = # of columns). A matrix can have any number of rows and any number of columns. If a matrix consists of either a single row or a single column, it is known as a vector.

$$\begin{array}{c}
 \left[\begin{array}{c} r_1 \\ r_2 \\ \cdot \\ \cdot \\ r_i \end{array} \right] \\
 \text{Column Vector } R_i
 \end{array}
 \qquad
 \begin{array}{c}
 \left[s_1 \ s_2 \ \cdot \ \cdot \ \cdot \ \cdot \ s_j \right] \\
 \text{Row Vector } S_j
 \end{array}$$

MATRIX OPERATIONS WITH THE FORMULA ROM

Getting back to the Formula ROM, we find that there are 8 keys associated with Matrix Operations in the 600 calculator. (If you have a 618 Memory Module, the U(, V(, and W(keys are used as well — we'll discuss these keys later in the text):

To use these functions, key SHIFT before touching the key you want to use. The SHIFT key enables you to use all the uppercase functions.

PRINT BEGIN	(	)	+	OUT PAUSE —	*	/	IN ↑	OUT →	NEXT I	DEFINE J	R(K	S(L	U(X	V(Y	W(Z
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15

The Formula ROM enables you to enter two separate matrices, by using the R(and S(keys.

ENTERING A MATRIX INTO THE CALCULATOR

Suppose you wished to enter a 3 X 3 matrix into your 600 calculator. To do this, you would first *define* the matrix as follows:

KEYSTROKES

BEG DEF R(3 , 3)

This will *partition*, or set aside, in the memory area of the 600 calculator, an area where the elements of the matrix can be stored. Since your matrix has 3 X 3 or 9 elements, the Formula ROM will automatically set aside 9 storage registers in your 600 calculator, beginning with register **26** , and ending in register **34** .

In a similar manner, if a second matrix of dimension (4 X 5) is needed internally, in your 600 calculator, you would do the following:

KEYSTROKES:

BEG DEF S(4 , 5)

Again, this will partition a section of the 600 calculator's memory area where the elements can be stored. Since your matrix has 4 X 5 or 20 elements, the Formula ROM will automatically set aside 20 storage registers in your 600 calculator, beginning with the first register following the R(partition, and ending 19 registers away. If your R matrix is 3 X 3, the S matrix will be stored beginning in register **35** and ending in register **54** .

NOTE:

For an explanation of data storage in your Model 600, refer to Chapter 10 of your 600 Manual.

When defining the R and S matrices, it is necessary to first define the R(partition. This is because the Formula ROM always partitions the calculator memory area for the S matrix after the R matrix partition. For the same reasons, the R matrix should always be used when only one matrix is needed. Also, if you wish to define only a vector, all that is necessary is to key the name of the vector, followed by its size and a right parenthesis; e.g., R(3).

R & S MATRIX SIZE

The maximum size matrix your Model 600 can handle internally is determined by the number of steps in the memory area of your Model 600.

- If you have a Model 600-2 (312 steps), the maximum number of elements you can store is 29.*
- If you have a Model 600-6 (824 steps), the maximum number of elements you can store is 93.*
- If you have a Model 600-14 (1848 steps), the maximum number of elements you can store is 221.*

Remember: The memory area of your 600 calculator is used for storing both data and program steps. The larger the matrix or matrices stored in the calculator, the less the number of program steps that can be stored in the remaining area.

Once you have defined the size of your matrix, it's just a matter of keying in the elements, row by row.

To do this, you use the R(or S(key, depending on the matrix you wish to enter, the IN key, and the 600 calculator's "numeric keyboard" as follows:

$$R = \begin{matrix} & r_{11} & r_{12} & r_{13} \\ r_{21} & & & \\ r_{31} & & & \end{matrix} \begin{matrix} r_{22} & r_{23} \\ r_{32} & r_{33} \end{matrix}$$

- 1.) Key BEG
- 2.) Key R(or S(if the S matrix were defined as well, and you wish to enter the S matrix elements.
- 3.) Key in the first matrix element, element r_{11} .
- 4.) Key IN

The display will now show the matrix position of the next element to enter. In this case, if we used R(, the display will show **1.020000000** . (Remember — The R matrix was defined as a 3 X 3 matrix.) The number in the display means "enter the second element of Row No. 1", or in matrix notation, "enter element r_{12} ."

1.020000000 = ROW # 1. COLUMN # 2

- 4.) Key in matrix element r_{12} .
- 5.) Key IN. (the display will show **1.030000000**).
- 6.) Key in matrix element r_{13} .
- 7.) Key IN (the display will show **2.010000000**).

Continue entering all elements, by using the same procedure. After all elements have been entered, the display will show **0.000000000** .

Once all the elements have been entered, any manipulating of elements can be handled by simply referring to the subscripts. You never have to worry about register numbers or addresses.

Suppose, for example, you wanted to multiply element r_{12} by r_{11} and assign the results to r_{32} . The steps would be:

KEYSTROKES:

BEG R(1 , 2) * R(1 , 1) → R(3 , 2)

After the right parenthesis of each matrix element is keyed, the value of the element is displayed: That is, you can view the elements at random. All that is necessary is to enter the particular subscript. Similarly, to add the R matrix elements, r_{21} , r_{22} , r_{23} together and assign the result to S matrix element s_{24} , (the S matrix having been previously defined), the following steps would be performed:

KEYSTROKES:

BEG R(2 , 1) + R(2 , 2) + R(2 , 3) → S(2 , 4)

Notice that in both examples, data was recalled, operated upon, and then stored without getting involved with any data register addresses.

CHECKING THE ELEMENTS OF A MATRIX IN MEMORY

Using the Matrix OUT PAUSE Feature

Once the elements of a matrix have been entered, you will probably want to check them, to assure yourself that you have entered the elements properly. The Formula ROM will display the position of each element (1.01, 1.02, etc.) sequentially in the display, followed by the associated element, automatically for 1 second. All you have to do is key the following:

BEG R(OUT PAUSE

The display will show the matrix position of each element for 1/2 second followed by the element itself, for 1 second.

Using the Matrix OUT Feature

In addition to being able to call out the entire matrix for display, each element can be called out individually in succession, by using either the R(key (for the R matrix) or the S(key (for the S matrix), together with the **OUT** key. (When you are using this feature, Storage Register **06** will always contain the matrix element position of the *next* element to be displayed — 1.02 after the 1st element, 1.03 after the second element, etc.)

EXAMPLE 11:

Define a matrix R (3,4) in your Model 600 calculator, key in the following numbers in the proper location, and then recall them both individually (by using the **OUT** key), and as a group (by using the **OUT PAUSE** key):

10	11	12	13		r_{11}	r_{12}	r_{13}	r_{14}
14	15	16	17	$= R (3,4) =$	r_{21}	r_{22}	r_{23}	r_{24}
18	19	20	21		r_{31}	r_{32}	r_{33}	r_{34}

NOTE:

*When entering or recalling many matrix elements, it is easier to temporarily depress the (÷) Selector Switch instead of the (X) Selector Switch. When the (÷) Selector Switch is depressed, it is unnecessary to depress the **SHIFT** key before using **IN** or **OUT** each time you enter or recall an element. Remember to depress the (X) Selector Switch again after you have finished entering or recalling matrix elements.*

PART I: Defining the Matrix and Entering the Elements

KEYSTROKES:

	<u>PRIME</u> CLEAR SWITCH				BEG	DEF	R(3 , 4)	
R(	10	IN	11	IN	12	IN	13	IN
	14	IN	15	IN	16	IN	17	IN
	18	IN	19	IN	20	IN	21	IN

PART II: Reviewing the Elements by Using the OUT Key

KEYSTROKES:

BEG R(OUT (The number **10.00000000** now appears in the display, and the number 1.02 will be in the register **06**).

OUT (The number **11.00000000** will appear in the display, and the number 1.03 will be in register **06** .)

OUT (The number **12.00000000** will appear in the display, and the number 1.04 will be in register **06** .)

OUT (The number **13.00000000** will appear in the display, and the number 2.01 will be in register **06** .)

⋮
⋮
⋮
⋮
⋮

OUT (The number **21.00000000** will appear in the display, and the number 0.000 will be in register **06** .)

PART III: Reviewing the Elements Automatically by Using the OUT PAUSE Key

KEYSTROKES:

BEG R(PAUSE OUT The entire matrix will be displayed element by element, row by row, preceded by its matrix position (except for the first element). The calculator display will look like this:

10.00000000 , **1.020000000** , **11.00000000** , **1.030000000** ,
12.00000000 , **1.040000000** , **13.00000000** , **2.010000000** , etc.

THE U(, V(AND W(KEYS, AND THE OPTIONAL MODEL 618 EXTENDED MEMORY

If more storage space is needed for matrix elements than the stand-alone 600 calculator offers, the Model 618 Memory Module can be added to the system. This will add an additional 122 to 506 matrix element storage registers to the 600 Formula ROM System.

The Model 618-1 Extended Memory adds a matrix storage capability of 122 elements.

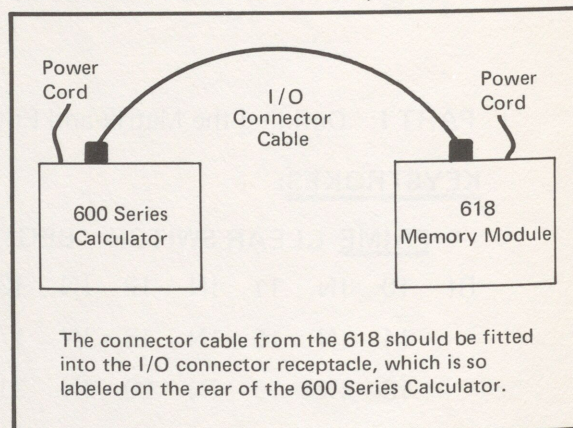
The Model 618-2 adds a matrix storage capability of 250 elements.

The Model 618-3 adds a matrix storage capability of 378 elements.

The Model 618-14 adds a matrix storage capability of 506 elements.

Equipment Setup

600 Series Calculator and 618 Memory Module



To use this added storage space, the Formula ROM offers you three additional matrices: U, V, and W, giving you a total of 5 possible matrices. Only the U, V, and W matrices are defined in the 618 Extended Memory.

Defining the U, V, and W Matrices

ADDRESSING THE EXTENDED MEMORY

In order to read or write information in the extended memory, it is necessary to first address the memory unit from the calculator. The procedure involves a two-step command; the first part is **GROUP 2**, while the second part is the **identification number** of the memory unit itself, and is normally generated on the calculator by using the numeric keyboard.

Unless otherwise identified, your 618 Extended Memory will respond to Code (00 00). To generate this code, depress the "0" key on the 10-key numeric keyboard.

GROUP 2
00 00

Once the extended memory has been addressed, it is unnecessary to address it again, unless the memory unit is turned off, or another peripheral is addressed.

GROUP
2

NUMERIC KEYBOARD

CHANGE SIGN	7	8	9
CLEAR DISPLAY	4	5	6
SET EXP	1	2	3
0	.		

Once the memory unit is addressed, the procedures used in defining and using the U, V, and W matrices are identical in every way to the procedures employed in using the R and S matrices.

To define U as a 10 X 12 matrix, the following keystrokes would be used:

BEG DEF U(10 , 12)

Elements would then be entered, recalled, and manipulated by the same procedures used with the R and S matrices.

As with the R and S matrices, the U matrix would be defined before the V matrix, and the V matrix before the W matrix. If one additional matrix is needed in the 618 Extended Memory, then only the U matrix would be defined; if two additional matrices were needed, then the U and V matrices, in that order, would be defined; and, if all three additional matrices were used, then the U, V, and W matrices, in that order would be defined. Once the matrix or matrices are defined, the elements are entered in the same way as the R and S matrix elements.

Manipulating the Elements of the U, V, and W Matrices

After all the elements of the U, V, and/or W matrices are entered, manipulating the elements is accomplished in the same manner as manipulating the elements of the R and S matrices. For example, if R and S are both defined as 3 X 3 matrices, and the U, V, and W are all defined as 5 X 5 matrices, any of the following operations would be valid:

To multiply element u_{15} of the U matrix by element v_{51} of the V matrix and store it in w_{55} of the W matrix:

KEYSTROKES:

BEG U(1 , 5) * V(5 , 1) → W(5 , 5)

To add element r_{11} of the R matrix to element s_{22} of the S matrix and store it as element u_{33} of the U matrix:

KEYSTROKES:

BEG R(1 , 1) + S(2 , 2) → U(3 , 3)

Likewise, we could add the first element of the second row of all 5 matrices together and store the result as element r_{21} of the R matrix:

KEYSTROKES:

BEG R(2 , 1) + S(2 , 1) + U(2 , 1) + V(2 , 1) + W(2 , 1) → R(2 , 1)

Programming with Matrix Operations

EXAMPLE 12:

Write a program which will add two 4 X 4 matrices together and store the result in the second matrix area. That is, add element r_{ij} of a (4 X 4) R matrix to the s_{ij} element of a (4 X 4) S matrix, and store it as element s_{ij} of the S matrix:

$$R(I, J) + S(I, J) \rightarrow S(I, J)$$

Since i and j will vary depending upon the row and column of the element, we will need two counter registers, and a loop.

Operating Instructions: —In RUN Mode—
Define R and S as 4 X 4 matrices,
enter their elements,
key PRIME, GO.

CALCULATOR KEY STORES — IN LEARN MODE —

BEG 1.004 → I → J → X } sets up I and J constants as counters
and saves 1.004 in X for later use in
resetting the J counter.

MARK

1
BEG R(I , J) + S(I , J) → S(I , J)

NEXT

J

Jump if 0

SEARCH

1

X → J

NEXT

I

Jump if 0

SEARCH

1

END PROG

To verify that the program works, use the OUT PAUSE feature to review the elements of the S matrix: i.e.,

KEYSTROKES

BEG S(OUT PAUSE

APPENDIX A

ADDITIONAL NOTES AND HINTS

HOW TO HANDLE LARGE EXPRESSIONS WITH MANY NESTED PARENTHESES

When entering an expression with many sets of nested parentheses, it is a good idea to enter the expression in the following manner:

(Figure 1) $((4 - X)^2 - 2)^2 + 4)^{1/2}$

instead of

(Figure 2) $(4 + (2 - (4 - X)^2)^2)^{1/2}$

That is, open as many parentheses as possible immediately. By doing this, the Formula ROM can evaluate each part of the expression as soon as each right (closed) parenthesis is entered. Notice that in (Figure 1), the first right-hand parenthesis is entered after fewer operators and constants (+, -, 4, 2) are entered than in (Figure 2): Yet both are equivalent expressions. An alternate way of approaching the problem would be to break it up into smaller parts and define each part as a constant (see page 7, DEFINING AND USING THE CONSTANTS.)

(Figure 1) is the preferred way of entering this type of expression since it increases both the size of the expression that the Formula ROM can handle, and the speed at which it is calculated. This is because, in the case of Figure 1, the ROM can evaluate the expression as you enter it rather than waiting until you are finished. If, however, after following this procedure, you find that your expression is still too large (error code 01), you can increase the 600 + Formula ROM's capacity to evaluate expressions temporarily. To do this, store a number larger than 25 in register 04 just before keying 00 (either **BEG** or **PRINT**): (Note the available registers will now start at one register greater than the number keyed into register 04.) This will increase the number of operators and terms you will be able to evaluate, but at the same time lessen the number of matrix elements you can define internally in your 600 calculator.

For example, if the number "30" is entered in register 04, an additional five operators, variables, and constants ($30 - 25 = 5$) can be included in an expression before evaluation, without the Formula ROM giving an 01 error code. However, in this situation, the R(matrix would start at register 31 instead of register 26. This means that the calculator will be able to hold five matrix elements less than before ($26 - 31 = -5$).

To reset the calculator + Formula ROM to the regular setting of 25, depress the red CLEAR Switch, and 00 (either **BEG** or **PRINT**).

SHARING THE KEYBOARD REGISTERS WITH THE FORMULA ROM

Chapter 4 (pages 20 - 27) of the 600 Manual describes the usage of the 16 Basic Keyboard Registers found on your Model 600. The Formula ROM also makes use of these keyboard registers in order to operate efficiently. (In actual practice, the Formula ROM uses at least registers 00 through 25.) Registers 09 through 15 are used to store the values you define as I, J, K, L, X, Y, and Z. Therefore, you can Store, Recall, etc., data in these registers, defining or changing the values of these "constants" as you find it necessary during the solution of problems, either manually or under program

Similarly, registers **00** , **01** , **05** , **06** , and **07** are used by the Formula ROM as scratch registers; you can use these registers as you need (except when using the IN and OUT matrix operators, when only the **00** register will be available), for storing data, for programmed compare decisions (see pages 78 - 83 of the 600 Manual), or as pointer registers for indirect addressing (Chapter 11 of the 600 Manual). Just remember that the numbers stored in these five registers will change when you use the various Formula ROM functions. For this reason, these registers should be used only as temporary storage areas, if you intend to use the Formula ROM while you are using these registers.

NOTE:

*Although many of the keyboard registers can be used for other purposes while you are using the Formula ROM, registers **02** , **03** , **04** , and **08** should not be used for other purposes, while you are using the Formula ROM. Changing the contents of these registers will alter the Formula ROM functions, and may cause the Formula ROM to give incorrect results or error messages. Similarly, registers **16** through **25** should not be used while the Formula ROM is being utilized. (This number may be extended beyond register **25** as described in the previous section, "How to Handle Large Expressions").*

APPENDIX B

SUBROUTINE LEVELS AND THE FORMULA ROM*

As described in the 600 Manual, your 600 calculator has the ability to operate with nested or several-level subroutines. This increases the complexity of the problems your Model 600 can handle.

Altogether, your 600 calculator can handle up to 8 levels of subroutines, if the calculator is programmed from the regular keyboard. If the Formula ROM is used for programming, the number of subroutine levels which the 600 calculator can handle will vary, depending upon the functions called:

— SUBROUTINE LEVELS REMAINING —

NO PRINT		PRINT	NO PRINT		PRINT
BEG.	7	6	→Y	6	6
(	6	6	→Z	6	6
+	6	4	DEF	6	6
-	6	4	→R(	4	4
*	6	4	→S(	4	4
/	6	4	→U(	5	5
↑	6	4	→V(	5	5
)	5	4	→W(	5	5
→	5	5	I	7	7
→I	6	6	,	6	6
→J	6	6	J	7	7
→K	6	6	)	6	6
→L	6	6	NEXT	6	
→X	6	6	I	6	
		R(, S(		U(, V(, W(	
IN	4			3	
OUT	4			3	
OUT PAUSE	3			2	

*(For an explanation of Subroutine Levels, see pages 83 - 92 of the 600 Manual.)

APPENDIX C

OPTIONAL COLUMN PRINTER CODES

The following codes are all generated by depressing the (X) and (Sp) Selector Switches, and keying one of the special function keys, [00] through [15] .

Formula ROM Command	Programming Code	Column Printer Symbol	Generated By Keying
BEG	12 00	A0	[00]
(	12 01	A1	[01]
)	12 02	A2	[02]
+	12 03	A3	[03]
-	12 04	A4	[04]
*	12 05	A5	[05]
/	12 06	A6	[06]
↑	12 07	A7	[07]
→	12 08	A8	[08]
I	12 09	A9	[09]
J	12 10	A10	[10]
K	12 11	A11	[11]
L	12 12	A12	[12]
X	12 13	A13	[13]
Y	12 14	A14	[14]
Z	12 15	A15	[15]
PRINT	13 00	B0	SHIFT [00]
, (comma)	13 02	B2	SHIFT [02]
OUT PAUSE	13 04	B4	SHIFT [04]
IN	13 07	B7	SHIFT [07]
OUT	13 08	B8	SHIFT [08]
NEXT	13 09	B9	SHIFT [09]
DEF	13 10	B10	SHIFT [10]
R(	13 11	B11	SHIFT [11]
S(	13 12	B12	SHIFT [12]
U(	13 13	B13	SHIFT [13]
V(	13 14	B14	SHIFT [14]
W(	13 15	B15	SHIFT [15]

NOTES:

NOTES:



LABORATORIES, INC.

836 North Street
Tewksbury, Massachusetts 01876
TELEPHONE (617) 851-7311
TWX 710-343-6769
TELEX 94-7421

WANG LABORATORIES (CANADA) LTD.

180 Duncan Mill Road
Don Mills, Ontario
TELEPHONE (416) 449-7890
TELEX 06-21-7549

WANG EUROPE, S.A.

Rue Anatole France 115-121
1030 Brussels, Belgium
TELEPHONE 15.48.01/02/03

WANG ELECTRONICS LTD.

40-44 High Street
Northwood, Middlesex, England
TELEPHONE Northwood 27677

WANG FRANCE SARL

47, Rue de la Chapelle
Paris 18, France
TELEPHONE 203.27.94 or 203.25.94

WANG LABORATORIES GMBH

Moselstrasse No. 4
6000 Frankfurt am Main
West Germany
TELEPHONE (611) 23-00-40

WANG SKANDINAVISKA AB

Fredsgaten 17
S-172 33
Sundbyberg, Sweden
TELEPHONE 98-12-45

WANG NEDERLAND N.V.

Danstraat 2
Utrecht, Netherlands
TELEPHONE 030-930947

WANG PACIFIC LTD.

61, King Yip Street, 1st Floor
Kwun Tong, Kowloon, Hong Kong
TELEPHONE K-450528

WANG INDUSTRIAL CO., LTD.

9, Sec. 5, Nanking East Road
Taipei, Taiwan
TELEPHONE 77.96.19 or 71.35.78

WANG ELEKTRONIK GESELLSCHAFT MBH

Grinzing Allee 16
1190 Wien 19
Austria
TELEPHONE (0222) 32.42.43

WANG MEDICAL SYSTEMS, INC.

836 North Street
Tewksbury, Massachusetts 01876
TELEPHONE (617) 851-7311
TWX 710-343-6769
TELEX 94-7421

PHI COMPUTER SERVICES, INC.

800 Massachusetts Avenue
Arlington, Massachusetts 02174
TELEPHONE (617) 648-8550

Printed in U.S.A.
700-0664
7-72-2.5M
Price: \$5.00