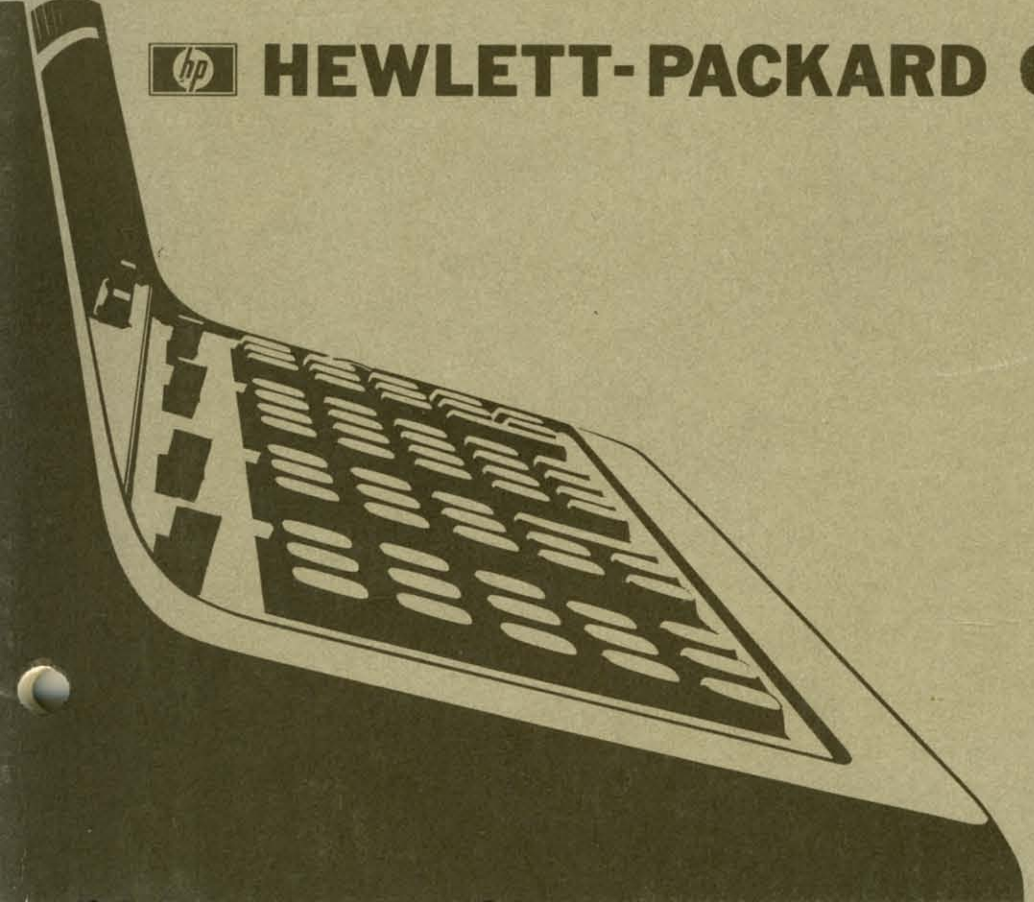
HEWLETT-PACKARD CALCULATOR

Workbook



CALCULATOR WORKBOOK

A Self Teaching Guide to the HP-9100A

HEWLETT hp PACKARD

Hewlett-Packard Company
Loveland, Colorado 80537

Table of Contents

UNIT 1

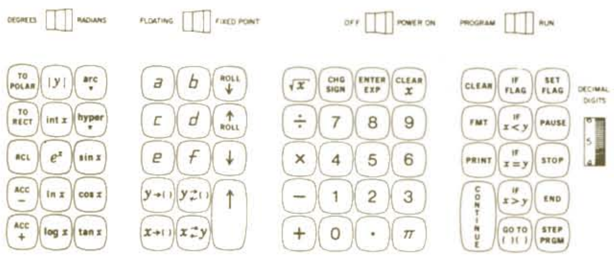
GETTING STARTED

Section 1-1 Introduction

The Hewlett-Packard 9100A is a computer. It can be operated manually, in which case it serves as a calculator, or it can be programmed (given a set of instructions) and serve as a computer.

FIGURE 1

Note the CRT (cathode ray tube) display screen and the keyboard in Figure 1. Results of keys pressed on the keyboard are shown on the display screen. There are three rows of numbers. The bottom row is referred to as the X register or keyboard register. When numbers are first entered into the machine, they appear in the X register. The middle row is called the Y register, or Y accumulator since the results of operations performed are shown in this register. The upper row is the Z register or Z temporary. Arithmetic operations cannot be performed on numbers in the Z register.



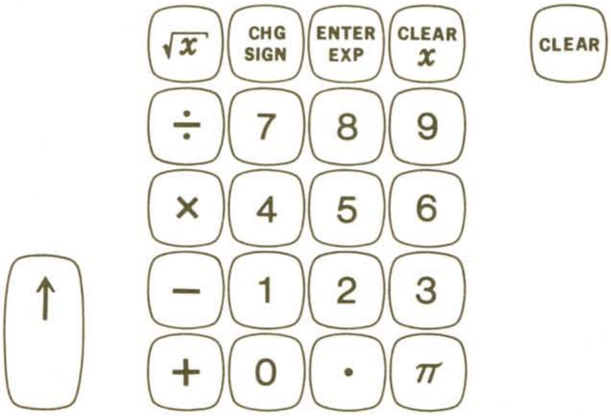
When first operating the machine, the switches along the top of the keyboard should be in the position shown. That is, the POWER switch ON, the PROGRAM-RUN switch to RUN, the FLOATING-FIXED POINT switch to FIXED POINT. (The DEGREES-RADIANS switch will be discussed later and may be in either position for now.)

The decimal digit wheel on the right side of the keyboard will set the number of places to the right of the decimal point. The least significant figure is rounded. A number which is too large for a particular setting will automatically go to floating decimal point (similar to scientific notation). This is discussed in Section 1-3.

In the following exercises, the decimal wheel should be set at five.

Section 1-2 Entry and Arithmetic Keys

In the next few sections, we will use the following section of the keyboard.



Let's try adding two numbers using the calculator. Follow the instructions listed below, pressing the keys from left to right as they occur.

Find s $s = a + b$ $a = 521$ $b = 273$

$\boxed{s = a + b}$ DECIMAL WHEEL AT 5

STEP	INSTRUCTION	KEYS	CRT DISPLAY	
0	CLEAR	CLEAR	0. 0. 0.	Z Y X
1	ENTER a	5 2 1	0. 0. 521.	Z Y X
2	↑	↑	0. 521. 521.	Z Y X
3	ENTER b	2 7 3	0. 521. 273.	Z Y X
4	+	+	0. 794. 273.	Z Y X

Note the answer is located in the Y register. Why must step 2 be included in the instructions? ______________

____

____

____

What would be on the display screen if step 2 was omitted? Show how the display screen would appear after step 3.

Z

Y

X

Would you obtain the same sum omitting step 2? ______

____

The set of instructions listed above to carry out this computation is an example of a program. Using this program, compute $s = a + b$ where $a = 7.32$ and $b = 12.15$. Show the final results on the display screen.

$s = a + b$

$a = 7.32$ $b = 12.15$

Z

Y final result

X

Let's change the above program so that we can find the product of two numbers.

$p = ab$ $a = 42.3$ $b = 37$

$\boxed{p = ab}$ DECIMAL WHEEL AT 5

STEP	INSTRUCTION	KEYS	CRT DISPLAY	
0	CLEAR	CLEAR	0. 0. 0.	Z X X
1	ENTER a	4 2 · 3	0. 0. 42.3	Z Y X
2	↑	↑	0. 42.3 42.3	Z Y X
3	ENTER b	3 7	0. 42.3 37.	Z Y X
4	×	×	0. 1565.1 37.	Z Y X

Suppose we wanted to change the above program so that we could divide a by b. Can you suggest what change(s) must be made?

____

____

Try the new program for division using $a = 42.3$ and $b = 37$. Your final display should look like this:

0.	Z
1.14324	Y
37.	X

Execute the program for multiplication on the following numbers and show the final display.

$p = ab$

DECIMAL WHEEL AT 5

1. $a = 5$ $b = 6$

Z
Y final result
X

2. $a = 73$ $b = 1.2$

Z
Y final result
X

3. $a = 17$ $b = -2$ *

Z
Y final result
X

*To obtain −2, press CHG SIGN 2 or 2 CHG SIGN

SO FAR:

1. Numbers must be entered from left to right.
2. The result of an arithmetic operation appears in the Y register.
3. ↑ Duplicates what is in X register into the Y register.
4. CHG SIGN Changes the sign of the contents of the X register.
5. + Adds the contents of the X and Y registers and places the result in the Y register.
6. − Subtracts the contents of the X register from the contents of the Y register and places the result in the Y register.
7. × Multiplies the contents of the X and Y registers and places the result in the Y register.
8. ÷ Divides the contents of the Y register by the contents of the X register and places the result in the Y register.

We should be ready for some problems:

1. Find the area of a rectangle.
   First we need a formula: $A = lw$
   What operation is indicated?

Follow through the program below filling in what you think will appear on the CRT display.

Let $l = 52$ $w = 13$

$A = lw$

DECIMAL WHEEL AT 5

STEP	INSTRUCTION	KEYS	CRT DISPLAY
0	CLEAR	CLEAR	Z Y X
1	ENTER *l*	5 2	Z Y X
2	↑	↑	Z Y X
3	ENTER *w*	1 3	Z Y X
4	×	×	Z Y X

Your answer should be 676.

2. Area of a triangle.

Formula: $A = \frac{1}{2} bh$

This time fill in the keys you would press and the CRT display.

Find A when $b = 12$, $h = 4$

$\boxed{A = \frac{1}{2} bh}$

DECIMAL WHEEL AT 5

STEP	INSTRUCTION	KEYS	CRT DISPLAY	
0	CLEAR			Z Y X
1	ENTER *b*			Z Y X
2	↑			Z Y X
3	ENTER *h*			Z Y X
4	×			Z Y X
5	ENTER ½ *			Z Y X
6	×		0. 24.0 .5	Z Y X

*Remember ½ = 0.5

Check your final display against the final display in the program. Can you suggest another way of entering ½?

____

____

____

Locate the $\boxed{\pi}$ key. Pressing this key, with the decimal wheel at 5 will give an approximation of π to 5 decimal places. Change the decimal wheel and note how the value of π changes. Place the decimal wheel back at 5, and let's try the following program using π.

3. Area of a circle.
Formula: $A = \pi r^2$ Let $r = 7$
Follow through this program.

$\boxed{A = \pi r^2}$ DECIMAL WHEEL AT 5

STEP	INSTRUCTION	KEYS	CRT DISPLAY	
0	CLEAR	CLEAR	0. 0. 0.	Z Y X
1	ENTER r	7	0. 0. 7.	Z Y X
2	↑	↑	0. 7. 7.	Z Y X
3	×	×	0. 49. 7.	Z Y X
4	ENTER π	π	0. 49. 3.14159	Z Y X
5	×	×	0. 153.93804 3.14159	Z Y X

Rotate the decimal wheel to 4, 3, 2, and record your answers.

D. W. at 4

Z
Y
X

D. W. at 3

Z
Y
X

D. W. at 2

Z
Y
X

What is the function of the decimal wheel?

Try some programs of your own.

---

4. Circumference of a circle.
Formula: $C = \pi d$ where $d = 13$

$\boxed{C = \pi d}$ DECIMAL WHEEL AT 5

STEP	INSTRUCTION	KEYS	CRT DISPLAY
0			
1			
2			
3			
4			

C = ____________________

5. Area of a trapezoid.
Formula: $A = \frac{1}{2} h(b + B)$
If $h = 4$, $b = 6$, $B = 7$, fill in the keys and the CRT display.

$\boxed{A = \frac{1}{2} h(b + B)}$

DECIMAL WHEEL AT 5

STEP	INSTRUCTION	KEYS	CRT DISPLAY
0	CLEAR		Z Y X
1	ENTER *b*		Z Y X
2	↑		Z Y X
3	ENTER B		Z Y X
4	+		Z Y X
5	ENTER *h*		Z Y X
6	×		Z Y X
7	ENTER 2		Z Y X
8	÷		Z Y X

A = ____________________

There are a few more keys which we should discuss in this first section.

The key CLEAR x does just what it says. When this key is pressed, whatever number is in the X register is cleared.

Before Pressing CLEAR x

521.	Z
12.2	Y
5.	X

After Pressing CLEAR x

521.	Z
12.2	Y
0.	X

The √x key takes the square root of the number which is in the X register and places the answer in the X register.

Before Pressing √x		After Pressing √x	
15.	Z	15.	Z
972.	Y	972.	Y
25.	X	5.00000	X

Try the following:

1. Find $\sqrt{29}$

DECIMAL WHEEL AT 5

STEP	INSTRUCTION	KEYS	CRT DISPLAY
0	CLEAR	CLEAR	0. Z 0. Y 0. X
1	ENTER 29	2 9	0. Z 0. Y 29. X
2	$\sqrt{29}$	√x	Z Y X

Can you suggest a different program which might give us a better display?

2. Find $\sqrt{-3}$

STEP	INSTRUCTION	KEYS	CRT DISPLAY
0			
1			
2			

What happened?

Let's try something else. Divide 5 by 0, fill in the program.

STEP	INSTRUCTION	KEYS	CRT DISPLAY
0			
1			
2			
3			
4			

What happened? ______________________________

____

Can you explain what this means? ______________________________

____

____

SO FAR:

1. Decimal wheel sets the number of decimal places to the right of the decimal point.
2. The calculator recognizes illegal operations (such as dividing by zero, taking the square root of a negative number) by means of a red light in the display screen.
3. π Approximates the value of π and places it in the X register.
4. CLEAR x Clears the X register. Does not affect the Y or Z registers.
5. $\sqrt{x}$ Takes the square root of the number in the X register. Does not affect the Y or Z registers.

The following problems are not a program. They are each individual problems. Predict each of the final displays and then check your results using the machine.

Example

ORIGINAL DISPLAY		KEY PRESSED
5.	Z	CLEAR x
7.	Y	
9.	X	

FINAL DISPLAY

5.	Z
7.	Y
0.	X

1.

ORIGINAL DISPLAY		KEY PRESSED
5.	Z	×
7.	Y	
9.	X	

FINAL DISPLAY

	Z
	Y
	X

2. ORIGINAL DISPLAY KEYS PRESSED

5.	Z
7.	Y
9.	X

$\sqrt{x}$

FINAL DISPLAY

	Z
	Y
	X

3. ORIGINAL DISPLAY KEYS PRESSED

5.	Z
7.	Y
9.	X

× 2

FINAL DISPLAY

	Z
	Y
	X

4. ORIGINAL DISPLAY KEYS PRESSED

5.	Z
7.	Y
9.	X

$\sqrt{x}$ ×

FINAL DISPLAY

	Z
	Y
	X

The int x key takes the integer part of the number in the X register. That is, it clears the decimal part.

Before pressing int x

7.51	Z
3.4	Y
1.2	X

After pressing int x

7.51	Z
3.4	Y
1.	X

The $|y|$ key takes the absolute value of the contents of the Y register.

Before pressing $|y|$

7.	Z
−3.1	Y
5.	X

After pressing $|y|$

7.	Z
3.1	Y
5.	X

Section 1-3 Floating Point Notation

Actually, numbers are represented in the computer as *floating point numbers*. Floating point notation used in computers is similar to the scientific notation used in many mathematics and science textbooks. Some examples are shown here.

FIXED POINT NOTATION (Standard Notation)	SCIENTIFIC NOTATION	FLOATING POINT	
123.45	1.2345×10^2	1.2345	02
10,000,000	10^7 or 1×10^7	1.	07
0.000001	10^{-6} or 1×10^{-6}	1.	−06
3.14159	3.14159×10^0	3.14159	00
−987.65	-9.8765×10^2	−9.8765	02
−0.000597	-5.97×10^{-4}	−5.97	−04
*	6.02×10^{23}	6.02	23
*	1.67339×10^{-24}	1.67339	−24

*Why do we hesitate to write these numbers in standard notation? ______________________________

A floating point number has two parts called the precision part and the exponent.

1.2345	02
↑	↑
Precision part	Exponent

Some additional examples are shown below.

FLOATING POINT NUMBER	PRECISION PART	EXPONENT
−1. 06	−1.	06
3.14159 00	3.14159	00
1.67339 −24	1.67339	−24
1.2345 02	1.2345	02

Write each floating point numeral in fixed point notation and scientific notation. The first two are completed for you.

FLOATING POINT NOTATION	SCIENTIFIC NOTATION	FIXED POINT NOTATION
1.23456 −05	1.23456×10^{-5}	0.0000123456
1.23456 08	1.23456×10^{8}	123456000.
2.71828 00		
6.93147 −01		
3.10062 01		
6.07610333 03		
1.0 −07		
5.893 −05		

Write each number in floating point.

NUMBER	FLOATING POINT NUMBER
186000	1.86 05
3959	
6.6254×10^{-27}	
0.001118	
3.1416	

In the HP 9100A, all numbers are represented internally in floating point notation with a 10 digit precision part and a two digit exponent. The precision part is always in the range:

$$-10 < \text{precision part} < 10$$

The exponent is always an integer in the range:

$$-99 \leqq \text{exponent} \leqq 99.$$

Therefore, the calculator can operate on numbers in the range 10^{-99} to 10^{99}. These numbers can have up to 10 significant digits.

If the FLOATING - FIXED POINT switch is in the FLOATING position, the numbers represented in the X, Y, and Z registers are displayed on the CRT in floating point form. Each number is displayed with up to 10 digits in the precision part. For example,

1.673 4	−24	Z
6.02	23	Y
3.141 592 653	00	X
precision part	exponent	

When the switch is in FLOATING position, the DECIMAL DIGITS wheel has no effect on the display. If the switch is in the FIXED POINT position, the numbers in the X, Y, and Z registers are displayed in standard notation, rounded to the number of decimal places specified by the DECIMAL DIGITS wheel.

The number of digits that can be displayed in fixed point notation is limited by the following relationship:

> Number of digits to the left of the decimal point plus the decimal wheel setting is less than or equal to ten.

If a number is too large to be displayed in fixed point notation, the computer displays it as a floating point number. For example, the CRT behaves like this when 35600.45 is entered:

D. W. at 2

```
     0.        Z
     0.        Y
 35600.45      X
```

$5 + 2 \leq 10$

D. W. at 6

```
 0.                Z
 0.                Y
 3.560 045   04    X
```

But $5 + 6 > 10$, so it switches to floating point.

The speed of light could be represented as follows:

```
 2.997 93   05    Z    kilometers per second
 9.835 72   08    Y    feet per second
 1.862 83   05    X    miles per second
```

Let's flip the switch to FIXED POINT. Here is how the display looks for DECIMAL WHEEL settings of 0, 3, and 6.

D. W. at 0

```
     299793.     Z
  983572000.     Y
     186283.     X
```

D. W. at 3

```
     299793.          Z
 9.835 72      08     Y
     186283.          X
```

D. W. at 6

```
 2.997 93    05    Z
 9.835 72    08    Y
 1.862 83    05    X
```

You can set up the CRT as we have shown it by setting the FLOATING - FIXED POINT switch to FLOATING and executing the following program.

STEP	INSTRUCTION	KEYS
0	CLEAR	CLEAR
1	ENTER 2.99793 x 10^5	2 9 9 7 9 3 ENTER EXP 5
2	↑	↑
3	ENTER 9.83572 x 10^8	9 8 3 5 7 2 ENTER EXP 8
4	↑	↑
5	ENTER 1.86283 x 10^5	1 8 6 2 8 3 ENTER EXP 5

Since 1.86283 x 10^5 = 186283, we could also execute Step 5 by pressing only the keys:

1 8 6 2 8 3

The ENTER EXP key is used to enter numbers in SCIENTIFIC or FLOATING POINT notation.

ENTER EXPONENT

ENTER EXP The ENTER EXP key is used when entering a floating point number into the X register. Pressing ENTER EXP clears the exponent (in X) and causes the next digit entries (0 to 99) and CHANGE SIGN to affect only the exponent.

For example,

To enter 6.02 x 10^{23} press

6 0 2 ENTER EXP 2 3

To enter 7 million press

7 ENTER EXP 6

Experiment with the ENTER EXP key and see how it has to be used.

Set the FLOATING - FIXED POINT switch to FIXED POINT.

DECIMAL WHEEL AT 5

Enter 4500 using the ENTER EXP key.

Which of the following methods would be correct?

KEYS

(1) 4 5 ENTER EXP 2

(2) 4 5 ENTER EXP 3

(3) 4 · 5 ENTER EXP 3

Methods (2) and (3) would both give you a value of 4500. But method (1) gives 450.

As a precaution, it would be best to change numbers in standard notation to scientific or floating notation when you want to enter numbers using an exponent.

Section 1-4 Control Keys

Often we want to keep some of the numbers we have in the machine, but want them in different positions. The following control keys take care of this problem.

↑ When the ↑ key is pressed, the computer copies the content of the Y register into the Z register, and copies the content of the X register into the Y register. The X register is not changed. The previous content of Z is erased from the display.

Before pressing ↑

333.	Z
22.	Y
1.	X

After pressing ↑

22.	Z
1.	Y
1.	X

You can probably guess what the ↓ key does!

↓ When the ↓ key is pressed, the computer copies the content of the Y register into the X register, and copies the content of the Z register into the Y register. The Z register is not changed. The previous content of X is erased from the display.

Before pressing ↓

333.	Z
22.	Y
1.	X

After pressing ↓

333.	Z
333.	Y
22.	X

Notice, however, that when either of these keys are used, you are losing some of the numbers which were in the machine.

The following two keys change the order of the display, but all of the numbers are retained.

The first key is the ↑ ROLL key. Study the example below.

Before pressing ↑ ROLL

333.	Z
22.	Y
1.	X

After pressing ↑ ROLL

22.	Z
1.	Y
333.	X

Note that all the numbers are still displayed. Only their order has been changed.

Try completing the box below.

Before pressing ↑ ROLL

4.	Z
6.	Y
7.	X

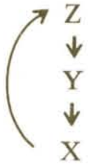
After pressing ↑ ROLL

	Z
	Y
	X

This key rotates the contents of the X, Y, and Z registers as follows.

The ROLL ↓ key performs the rotation in the opposite direction to that of the ↑ ROLL key.

ROLL ↓ Rotates the contents of the X, Y, and Z registers as follows.



If we operate the ROLL ↓ key, the display will be shown as follows:

Before pressing ROLL ↓

333.	Z
22.	Y
1.	X

After pressing ROLL ↓

1.	Z
333.	Y
22.	X

Complete the display box below.

Before pressing ROLL ↓

4.	Z
6.	Y
7.	X

After pressing ROLL ↓

	Z
	Y
	X

There is another key which changes the order of the numbers on the display, and retains all the same numbers. This is the $x \rightleftarrows y$ key.

Frequently we may wish to interchange the contents of the X and Y registers without disturbing the content of the Z register. This is accomplished by using the $x \rightleftarrows y$ key.

$x \rightleftarrows y$ This key exchanges the content of the X and Y registers. The Z register is not affected.

To illustrate this operation, we show the following example.

Before pressing $x \rightleftarrows y$

333.	Z
22.	Y
1.	X

After pressing $x \rightleftarrows y$

333.	Z
1.	Y
22.	X

Try this operation by completing the display box on the right.

Before pressing $x \rightleftarrows y$

5.	Z
3.	Y
7.	X

After pressing $x \rightleftarrows y$

	Z
	Y
	X

We will use the following abbreviations for the keys we have just defined.

KEY	ABBREVIATION
↑ ROLL	R ↑
ROLL ↓	R ↓
$x \rightleftarrows y$	X ⇄ Y

Section 1-5 Traces

In order to clarify what happens when we execute a program on the HP 9100A, we will trace several programs. That is, for selected values of input variables, we will show a step-by-step description of what happens in the computer. Although you may find these traces to be quite obvious . . . carry on anyway! We are trying to develop a technique here that will become essential as you begin writing and analyzing more complicated programs.

We begin by numerically tracing a program that is already familiar to you. Here is a trace of our program to evaluate $s = a + b$; it shows the contents of the X, Y, and Z registers after each step is carried out.

$\boxed{s = a + b}$ $a = 57$ $b = 29$

STEP	INSTRUCTION	X	Y	Z
0	CLEAR	0.	0.	0.
1	ENTER a	57.	0.	0.
2	↑	57.	57.	0.
3	ENTER b	29.	57.	0.
4	+	29.	86.	0.
5	CLEAR X	0.	86.	0.

Now it's your turn. Trace the program for $s = a + b$, for $a = 857$ and $b = 296$.

$\boxed{s = a + b}$ $a = 857$ $b = 296$

STEP	INSTRUCTION	X	Y	Z
0	CLEAR			
1	ENTER a			
2	↑			
3	ENTER b			
4	+			
5	CLEAR X			

Next we show a trace of a program to evaluate $q = a \div b$, for $a = 29$ and $b = 32$. Execute this program on the computer.

$\boxed{q = a \div b}$

D. W. at 5

STEP	INSTRUCTION	X	Y	Z
0	CLEAR	0.	0.	0.
1	ENTER a	29.	0.	0.
2	↑	29.	29.	0.
3	ENTER b	32.	29.	0.
4	÷	32.	.90625	0.
5	CLEAR X	0.	.90625	0.

Without using the HP 9100A, complete the numeric trace of this program using $a = 75$ and $b = 3$.

$\boxed{q = a \div b}$

STEP	INSTRUCTION	X	Y	Z
0	CLEAR	0.	0.	0.
1	ENTER a			
2	↑			
3	ENTER b			
4	÷			
5	CLEAR X			

Here is a "do nothing" program. We have partially traced it numerically. It should help you understand the previous keys. Complete the trace.

STEP	INSTRUCTION	X	Y	Z
0	CLEAR	0	0	0
1	ENTER 3	3	0	0
2	R ↑	0	3	0
3	ENTER 2	2	3	0
4	R ↑	0	2	3
5	ENTER 1			
6	R ↑			
7	X ⇄ Y			
8	R ↑			
9	X ⇄ Y			
a	R ↓			
b	X ⇄ Y			
c	CLEAR			

We can also show a trace in terms of variables and algebraic expressions rather than specific numerical values. For example, here is the algebraic trace of a program to evaluate $s = a + b$.

$s = a + b$

STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR	0	0	0	
1	ENTER *a*	*a*	0	0	Enter value of *a*.
2	↑	*a*	*a*	0	Copy value of *a* into Y.
3	ENTER *b*	*b*	*a*	0	Enter value of *b*.
4	+	*b*	$a + b$	0	The value of $a + b$ is in *Y*.

In Step 1, we show the letter "*a*" in X. We mean, of course, that the numerical value of *a* is in the X register. Similarly in Step 3, we mean that the numerical value of *b* is in X; the numerical value of $a + b$ is in Y after execution of Step 4.

Now do a numeric trace for $a = 5.76$ and $b = 8.39$; show the trace.

$\boxed{s = a + b}$

STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR				
1	ENTER *a*				
2	↑				
3	ENTER *b*				
4	+				

Next we trace a program to compute the area **A** of a circle of radius *r*.

$\boxed{A = \pi r^2}$

STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR	0	0	0	
1	ENTER *r*	r	0	0	Enter value of *r*.
2	↑	r	r	0	
3	×	r	r^2	0	
4	ENTER π	π	r^2	0	$\pi \cong 3.141592654$
5	×	π	πr^2	0	The value of πr^2 is in **Y**.

Trace the same program numerically for $r = 5$

$\boxed{A = \pi r^2}$

STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR				
1	ENTER *r*				
2	↑				
3	×				
4	ENTER π				
5	×				

Notice how the final display appears in the two programs above. What is the content of X?__________ Is this necessary in the final display? ______________________________

____

______________________________ . In the future, clear the display screen so that only information pertinent to the result is displayed.

How could we change the above two programs so the screen appears as follows?

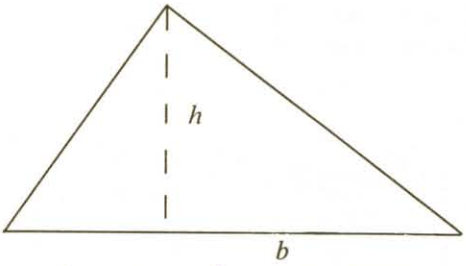
0.	Z
A	Y
0.	X

____

____

Consider the area of a triangle of base b and height h. Examine the algebraic trace of the program below and complete the REMARKS column.

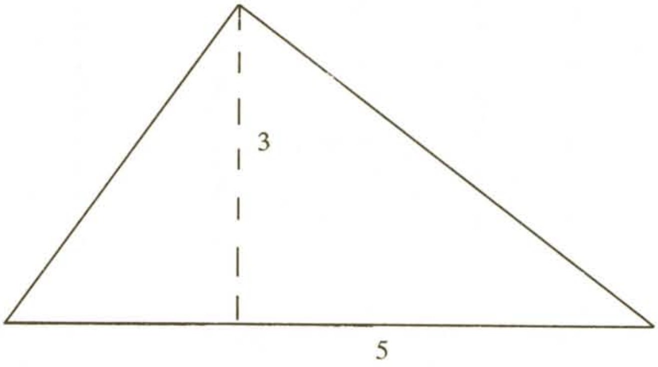
$$\boxed{A = \tfrac{1}{2}bh}$$



STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR	0	0	0	
1	ENTER b	b	0	0	
2	↑	b	b	0	
3	ENTER h	h	b	0	
4	×	h	bh	0	
5	ENTER 2	2	bh	0	
6	÷	2	$\frac{1}{2}bh$	0	
7	CLEAR X	0	$\frac{1}{2}bh$	0	

Trace the same program numerically for a triangle with $b = 5$ and $h = 3$.

$$\boxed{A = \tfrac{1}{2}bh}$$



STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR				
1	ENTER b				
2	↑				
3	ENTER h				
4	×				
5	ENTER 2				
6	÷				
7	CLEAR X				

Complete the following algebraic trace of a program to evaluate A = P + P*r*.

$\boxed{A = P + Pr}$

STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR	0	0	0	
1	ENTER *r*	*r*	0	0	
2	↑				
3	ENTER P				
4	×				
5	+				
6	CLEAR X				

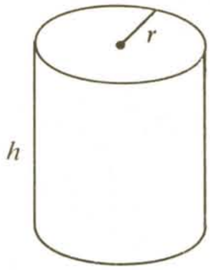
Now trace the program numerically for $r = 0.06$ and $P = 1000$.

$\boxed{A = P + Pr}$

STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR				
1	ENTER *r*				
2	↑				
3	ENTER P				
4	×				
5	+				
6	CLEAR X				

Therefore, A = ______________________________

Below is a program to compute the volume V of a right circular cylinder with radius *r* and height *h*. Write the algebraic trace.



$$\boxed{V = \pi r^2 h}$$

STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR				
1	ENTER *r*				Enter a value for *r*.
2	↑				
3	×				
4	ENTER π				$\pi \cong 3.141598654$
5	×				
6	ENTER *h*				Enter a value for *h*.
7	×				
8	CLEAR X				

Execute this program on the computer for $r = 2$, $h = 5$ and record your results. V = ____________________

Here is a program that uses the Z register. This program computes the value of $(a + b)\ (c + d)$. Look it over carefully.

$R = (a + b)\ (c + d)$

STEP	INSTRUCTION	X	Y	Z	REMARKS
0	CLEAR	0	0	0	
1	ENTER *a*	a	0	0	
2	↑	a	a	0	
3	ENTER *b*	b	a	0	
4	+	b	$a + b$	0	
5	ENTER *c*	c	$a + b$	0	
6	↑	c	c	$a + b$	
7	ENTER *d*	d	c	$a + b$	
8	+	d	$c + d$	$a + b$	
9	↓	$c + d$	$a + b$	$a + b$	
a	×	$c + d$	R	$a + b$	$R = (a + b)\ (c + d)$

Why is the Z register necessary for finding the product of the sums? ____________________

____

Execute the preceding program for $a = 4$, $b = 6$, $c = 13$, and $d = 5$.

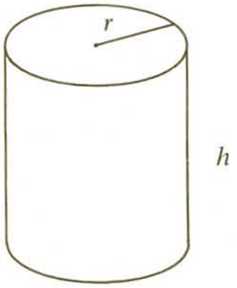
Record the results from the CRT below:

$a + b$ = __________________ Z register

$(a + b)\ (c + d)$ = __________________ Y register

$c + d$ = __________________ X register

Write a program to compute the surface area of a (closed) right circular cylinder with radius r and height h. As you write each step, also show the algebraic expression whose value is in each register.



$$S = 2\pi rh + 2\pi r^2$$

STEP	INSTRUCTION	X	Y	Z	REMARKS

If your program involved more than 9 steps, you are working too hard! Hint: $2\pi rh + \pi r^2 = 2\pi r(h + r)$.

Section 1-6 Vector Keys

There are a few other keys which may prove useful. They are referred to as vector keys since they deal with ordered pairs of numbers.

ACC + Adds the ordered pair of numbers in the X and Y registers to the ordered pair of numbers in the *f* and *e* registers. The result is stored in the *f* and *e* registers. The contents of X, Y and Z do not change.

For example,

$$e + Y \rightarrow e$$
$$f + X \rightarrow f$$

BEFORE PRESSING ACC +		AFTER PRESSING ACC +	
1.	Z	1.	Z
2.	Y	2.	Y
4.	X	4.	X
6.	*e*	8.	*e*
3.	*f*	7.	*f*

Complete the following:

BEFORE PRESSING ACC +

0.	Z
4.	Y
3.	X
5.	*e*
1.	*f*

AFTER PRESSING ACC +

	Z
	Y
	X
	e
	f

ACC − Subtracts the ordered pair of numbers in the X and Y registers from the ordered pair of numbers in the *f* and *e* registers. This does not affect the contents of the X, Y, and Z registers.

For example,

$$f - X \rightarrow f$$
$$e - Y \rightarrow e$$

BEFORE PRESSING ACC −		AFTER PRESSING ACC −	
1.	Z	1.	Z
2.	Y	2.	Y
5.	X	5.	X
6.	*e*	4.	*e*
3.	*f*	−2.	*f*

Complete the following.

BEFORE PRESSING ACC −

0.	Z
−4.	Y
3.	X
0.	*e*
0.	*f*

AFTER PRESSING ACC −

	Z
	Y
	X
	e
	f

UNIT 2

PROGRAMMING

So far we have been using the HP 9100A as a calculator. Although we have been writing programs, we have been using them manually. Let's make the HP 9100A do some of the work. We can make it behave like a computer, under control of a program that is stored in its memory.

Section 2-1 Storing a Program

Each key on the HP 9100A has a code. When a key is pressed, the machine reacts to the particular code attached to the key. Refer to the inside front cover for a list of each key and its code.

What do you notice about these codes? ____________

____

____

You should have noted that each code is a two-digit number and that the digits are not greater than 7.

Let's actually run through a program we've already done, but this time we'll put the computer to work.

First, you need a mark sense card. Study it for a moment. Look down the step column first. The first step would be step 00. (You must supply the first 0.) The second step is 01, next 02, and so on. Note that after step 09, the next step is 0a. After reaching step 0d, the calculator then goes to step 10, 11, 12, and so on.

Now look at the next column. It's divided into two parts. The left portion is used to write the key you want, and the right portion its code. For instance, suppose we wanted to press the CLEAR key. We would write:

KEY	
CLEAR	20

Should you make an error, use the "Skip" column. Filling in this box will prevent the card reader from reading your error. Be sure to renumber your next program step after using the "skip" column.

The last two columns are used to change the codes into a language the machine can understand. The first column is headed 40, 20, 10 and the second column is headed 4, 2, 1. All codes can be written in terms of 40, 20, 10, 4, 2, and 1. Let's see how it's done.

Remember the program we did for finding the area of a circle? (Refer to Section 1 - 2, Problem 3.)

Rewriting part of it and inserting the codes:

STEP	INSTRUCTION	KEYS	CODE
00	CLEAR	CLEAR	20
01	ENTER r	7 *	07
02	↑	↑	27
03	×	×	36
04	ENTER π	π	56
05	×	×	36
06	END **	END	46

*In this case, r was 7.

**All stored programs should end with this instruction. We will define the END key later.

Now put this program on your calculator program card. Darken the appropriate spaces in the right columns corresponding to the key-codes. Then check your answers with the example below.

STEP	KEY		40	20	10	4	2	1
00	CLEAR	20		—				
01	7	07				—	—	—
02	↑	27		—		—	—	—
03	X	36		—	—	—	—	
04	π	56	—		—	—	—	
05	X	36		—	—	—	—	
06	END	46	—			—	—	

TO ENTER THE PROGRAM into the HP 9100A

1. Switch PROGRAM - RUN switch to RUN.
2. Press CLEAR, END.
3. Switch PROGRAM - RUN switch to PROGRAM.
4. Place program card into the card reader.
5. Return PROGRAM - RUN switch to RUN.

The display screen (before returning switch to RUN) should look like the display below:

0.	00	Z
0.	00	Y
1.d----	*	X

*These digits may vary.

TO RUN THE PROGRAM:

1. With PROGRAM - RUN switch in RUN position, press END, CONTINUE.

Your answer should appear in the Y register, as it did in our original program, Section 1 - 2, Problem 3.

TO CHECK YOUR PROGRAM:

1. PROGRAM - RUN switch to RUN.
2. Press CLEAR, END.
3. PROGRAM - RUN switch to PROGRAM.
4. Press STEP PROGRAM.

The screen should look like this:

0.	00	Z
0.	00	Y
0.0----20		X

Step Code

As you continue to press STEP PRGM, you can run through your program, step by step, checking to see that you have the correct codes.

SHOULD YOU FIND AN ERROR

1. PROGRAM - RUN switch to RUN.
2. Press GO TO, then press the two digits which denote the step which has an error.
3. Return PROGRAM - RUN switch to PROGRAM, press the correct key.

To illustrate this: Suppose, in the program above, at step 03 we had put in ÷ instead of × . Naturally, when we ran the program, we would get an incorrect answer. In step-checking our program when we reached step 03, the display screen would appear as follows.

0.	00	Z
0.	00	Y
0.3----35		X

As you can see, the code number is not the one we originally wanted. To correct this step, turn switch to RUN, Press GO TO 0 3 , switch to PROGRAM, press × . Then step through the program again.

This method of step-checking your program is referred to as "debugging" your program.

Here is what you should observe when checking through your program.

PRESS THE KEY	OBSERVE THE DISPLAY	EXPLANATION
STEP PRGM	0. 00 0. 00 0.0---20	Key code 20 is in location 00
STEP PRGM	0. 00 0. 00 0.1---07	Key code 07 is in location 01
STEP PRGM	0. 00 0. 00 0.2---27	Key code 27 is in location 02
STEP PRGM	0. 00 0. 00 0.3---36	Key code 36 is in location 03
STEP PRGM	0. 00 0. 00 0.4---56	Key code 56 is in location 04
STEP PRGM	0. 00 0. 00 0.5---36	Key code 36 is in location 05
STEP PRGM	0. 00 0. 00 0.6---46	Key code 46 is in location 06

Corrections could also be made quickly on your calculator program card. Be sure to erase your error completely, write in the correct key and code, and enter the card in the card reader again.

Before going any further, let's list the definitions of some of the new keys we've been using.

STOP This instruction automatically halts execution of the program. The STOP instruction provides an unconditional halt of the program so that the operator can enter data or record results that appear on the display. Pressing CONTINUE will begin automatic execution again with the next program step.

END In a stored program, the END instruction halts execution of the program in the same way that a STOP instruction does. However, if the CONTINUE key is then pressed, the computer will begin at Step 00 instead of the step following END. In manual operation, pressing the END key has the same effect as pressing the three keys, GO TO . 0, 0.

CONTINUE With the PROGRAM - RUN switch in the RUN position, pressing CONTINUE starts the HP 9100A executing the stored program.

GO TO ( )( ) Directs the HP 9100A to a specific location in the program.

We have defined the STOP key, but, up to this point, we have not used it.

Look back at the program we just completed, finding the area of a circle. The program we wrote was for finding the area of a circle whose radius is 7. Clearly, you can see where this would be impractical. Revise the program so that it can be used to find the area of all circles.

In our definition of STOP, it states: "The STOP instruction provides an unconditional halt of the program so the operator can enter data . . .". We would like to use the STOP instruction when we want to enter new data.

In our program written on the computer program card described at the beginning of the Section, step 01 enters data. We must put a STOP in here instead of ENTER *r*. (Our code will change too.)

Where else are we entering data? ____________________

Why don't we have to put a STOP here? ____________________

____

So our new program should look like this:

STEP	INSTRUCTION	KEYS	CODE
00	CLEAR	CLEAR	20
01	STOP	STOP	41
02	↑	↑	27
03	×	×	36
04	ENTER π	π	56
05	×	×	36
06	CLEAR X	CLEAR x	37
07	END	END	46

Now write up a new program card, filling in the correct codes on the right and enter the program.

Since our program was altered, our instructions TO RUN THE PROGRAM will also change.

TO RUN THE PROGRAM

1. With PROGRAM - RUN switch in RUN position, press END, CONTINUE. (Pressing CONTINUE brings the program up to the first stop.)
2. Enter data, press CONTINUE.
   (If there is only one stop, the program will now continue to the end. If there is more than one stop, you must press CONTINUE after each data entry.)

Review what happened when you executed the program.

(1) When you pressed the END key, the computer was ready to begin the program. When you pressed the CONTINUE key, the CLEAR instruction, which was stored in step 00 was executed.

(2) The HP 9100A then went to Step 01 and executed the STOP instruction. It stopped.

(3) You then entered the data. (The value of *r*.)

(4) Then you told the computer to go ahead with the program by pressing the CONTINUE key again.

(5) The computer executed Steps 02, 03, 04, 05, 06 and 07 automatically. Step 07 is an END which tells the machine to stop—it did and displayed the answer.

All of these calculations took place in a very small fraction of a second so that we could not see them taking place. At the END instruction the computer stopped and displayed the answer.

Now go back to the first program we wrote in Unit 1, finding the sum of two numbers. Try to revise the program, supplying the STOP and END instructions. Write a computer program card and instructions to run the program. Then check your work with the following example.

STEP	KEY		40	20	10	4	2	1
00	CLEAR	20		—				
01	STOP	41	—					—
02	↑	27		—		—	—	—
03	STOP	41	—					—
04	+	33		—	—		—	—
05	CLEAR X	37		—	—	—	—	—
06	END	46	—			—	—	

TO RUN THE PROGRAM

1. Press END, CONTINUE.
2. Enter *a*, press CONTINUE.
3. Enter *b*, press CONTINUE.

A program need not start at Step 00. Suppose after entering the previous program for finding the area of a circle, you wanted to enter another program. But you don't want to erase the area of a circle program. You could start your new program at step 07—or at any step after 06.

Start a new program at step 07. Write your program on a program card, using step 07 as your first step. The instructions to enter the program will not change, but note below how the instructions to run the program do change.

TO RUN THE PROGRAM

1. Press GO TO ( )( ), Press 0, 7, then Press CONTINUE. (Remember, if there are stops in your program, the CONTINUE key must be pressed after entering the data at each STOP.)

Try the problems below.

1. Make a program to multiply two numbers, try the three problems in Section 1-2 to see if your program works.
2. Make a program for finding the area of a rectangle. Test your program for at least three different values of length and width.
3. Make a program for finding the area of a trapezoid.
4. Program $m = (a - b)(a - c)$. Fill in the chart below first.

STEP	INSTRUCTION	KEYS	X	Y	Z

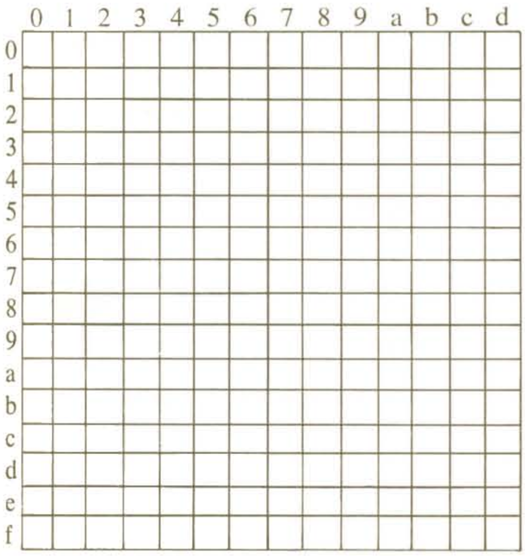
Section 2-2 The Memory

Perhaps you wonder how the instructions of a program are stored in the HP 9100A. The memory of the computer consists of 16 storage registers. They are labeled:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f

Registers e and f may be used only for storage of data (numbers). Registers 0 through 9 and a through d, however, can be used for storing data or for storing instructions.

When used for data storage, each register can store one floating - point number, with a 10 digit precision part and a two digit exponent. When used for instruction storage, each register can hold 14 steps. A step is equivalent to one keystroke. Since each of the 14 registers which store instructions can store 14 steps, a total of 196 program steps can be stored at any one time. Here is a diagram of the memory.



Registers 0 through d can store either data or instructions. Each register can hold one floating point number or 14 instructions. Only data in registers e and f. Each register can hold one floating point number.

Each register is subdivided into 14 locations. In each register, the locations are "numbered" 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, and d. In registers 0 through d, each location can contain one decimal digit or one program step. Program steps cannot be stored in registers e and f.

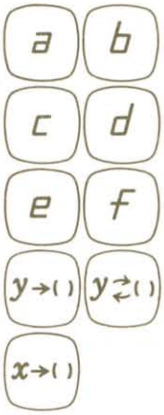
We will assign an address to each location. The address of a location is a two digit symbol such as 00, 01, 53, 7c, a2, ... The first digit designates the register and the second digit the character position within the register. Addresses range from 00 to dd, as follows:

00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0a, 0b, 0c, 0d, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1a, 1b, 1c, 1d, 20, 21, 22, etc.

Addresses correspond to step numbers in a program. Step 00 is stored in location 00, step 01 in location 01, and so on. In the computer a program counter keeps track of the location into which a program step is to be stored. When we set the PROGRAM - RUN switch to RUN and press END, we set the program counter to 00. Then we set the switch to program and begin entering program steps. As each step is entered, the address in the program counter is increased by one.

Section 2-3 Storing and Recalling Numbers

In this section we describe the use of the following keys:



We use the x→( ) and y→( ) keys when we want to copy numbers from the X and Y registers into the memory registers. We can store the content of X or the content of Y into any of the following registers:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f.

x→( ) Copy the content of the X register into the storage register indicated by the next keystroke. The previous content of the storage register is erased. The content of the X register remains the same.

For example:

x→( ) a Copy the content of X into register a.

x→( ) e Copy the content of X into register e.

x→( ) 7 Copy the content of X into register 7.

Here are some before and after examples.

BEFORE		KEYS	AFTER	
333.	Z		333.	Z
22.	Y	x→( ) c	22.	Y
1.	X		1.	X
0.	c		1.	c
13.	Z		13.	Z
2.	Y	x→( ) b	2.	Y
−3.	X		−3.	X
0.	b		−3.	b

y→( ) Copy the content of the Y register into the storage register indicated by the next keystroke. The previous content of the indicated storage register is erased. The content of the Y register remains the same.

For example:

y→( ) a Copy the content of Y into register a.

y→( ) f Copy the content of Y into register f.

y→( ) 9 Copy the content of Y into register 9.

Complete the following before and after examples.

BEFORE		KEY	AFTER	
333.	Z			Z
22.	Y	y→( ) c		Y
1.	X			X
6.2	c			c
1.	Z	x→( ) f		Z
2.	Y	y→( ) e		Y
3.	X			X
42.3	e			e
37.	f			f

To recall a number from any of the registers a, b, c, d, e, or f, we simply press the key with the letter of the register. The content of the register is copied into X. Here is an example.

a Recall a. Copy the content of register a into X. The previous content of X is erased. The content of register a remains unchanged.

BEFORE		KEY	AFTER	
3.	Z		3.	Z
2.	Y	a	2.	Y
1.	X		−.37	X
−.37	a		−.37	a

Let's use the memory registers in a program to add two fractions, a/b and c/d.

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd}$$

Given a, b, c, and d as input data, we want to compute the values of $ad + bc$ and bd. The procedure to run the program is on the following page.

STEP	KEY	X	Y	Z	b	d	REMARKS
00	CLEAR						
01	STOP	a					Enter a
02	↑	a	a				
03	STOP	b	a				Enter b
04	$x\rightarrow( )$	b	a				Store b
05	b	b	a		b		
06	↑	b	b	a	b		
07	STOP	c	b	a	b		Enter c
08	×	c	bc	a	b		
09	STOP	d	bc	a	b		Enter d
0a	$x\rightarrow( )$	d	bc	a	b		Store d
0b	d	d	bc	a	b	d	
0c	↑ ROLL	a	d	bc	b	d	
0d	×	a	ad	bc	b	d	
10	↓	ad	bc	bc	b	d	
11	+	ad	$ad+bc$	bc	b	d	
12	b	b	$ad+bc$	bc	b	d	Recall b
13	↑	b	b	$ad+bc$	b	d	
14	d	d	b	$ad+bc$	b	d	Recall d
15	×	d	bd	$ad+bc$	b	d	
16	CLEAR x	0	bd	$ad+bc$	b	d	Clear X
17	ROLL ↓	bd	$ad+bc$	0	b	d	
18	END	bd	$ad+bc$	0	b	d	Answers

TO RUN THE PROGRAM

1. Press END, Press CONTINUE
2. Enter *a*, press CONTINUE
3. Enter *b*, press CONTINUE
4. Enter *c*, press CONTINUE
5. Enter *d*, press CONTINUE

The final display looks like this.

```
0.
ad + bc
bd
```

Examples:

1. $\frac{2}{3} + \frac{4}{5}$

```
0.
22.
15.
```

Therefore: $\frac{2}{3} + \frac{4}{5} = \frac{22}{15}$

2. $\frac{3}{8} + \frac{1}{6}$

```
0.
26.
48.
```

Therefore: $\frac{3}{8} + \frac{1}{6} = \frac{26}{48}$

In writing this program, we used 7 of the 19 registers in the computer. We used registers 0, 1, b, d, X, Y, and Z. We stored our program in registers 0 and 1, then used registers b, d, X, Y, and Z for input data, intermediate results and final results.

Note our use of registers b and d. In steps 04 and 05, we stored the value of *b* in register b and in steps 0a and 0b, we stored the value of *d* into register d. Thus, we "saved" these numbers so we could recall them in steps 12 and 14 and compute the denominator, *bd*, of the result.

The following program is a different approach to the problem of adding two fractions. Complete the algebraic trace.

STEP	KEY	X	Y	Z	f	REMARKS
00	CLEAR					
01	STOP	*a*				Enter *a*
02	↑	*a*	*a*			
03	STOP	*b*	*a*			Enter *b*
04	↑	*b*	*b*	*a*		
05	STOP	*c*	*b*	*a*		Enter *c*
06	$x \rightleftarrows y$					

07	×					
08	y→( )					Store *bc*
09	f					
0a	x⇄y					
0b	STOP					Enter *d*
0c	×					
0d	↑ ROLL					
10	×					
11	f					Recall *bc*
12	+					
13	CLEAR x					
14	ROLL ↓					
15	x⇄y					
16	END	*bd*	*ad+bc*	0		

The procedure to run this program is the same as the procedure to run the preceding program.

The y⇄( ) key.

It is not possible to copy a number from one of the storage registers directly into the Y register. It is possible, however, to exchange the content of Y and the content of one of the storage registers. For example:

KEYS	DESCRIPTION
y⇄( ) a	Exchange the content of Y and the content of a.
y⇄( ) b	Exchange the content of Y and the content of b.
y⇄( ) 8	Exchange the content of Y and the content of register 8.

Here is another way to look at it.

BEFORE		KEYS	AFTER	
333.	Z		333.	Z
22.	Y	y⇄( ) c	6.2	Y
1.	X		1.	X
6.2	c		22.	c

Complete the following diagram.

BEFORE		KEYS	AFTER	
0.	Z			Z
3.7	Y	y⇄( ) a		Y
.68	X			X
235.	a			a

We will use the y⇄( ) key in still another program to add two fractions.

STEP	KEY	X	Y	X	f	REMARKS
00	CLEAR					
01	STOP	*a*				Enter *a*
02	↑	*a*	*a*			
03	STOP	*b*	*a*			Enter *b*
04	x→( )	*b*	*a*			Store *b*
05	f	*b*	*a*		*b*	
06	↑	*b*	*b*	*a*	*b*	
07	STOP	*c*	*b*	*a*	*b*	Enter *c*
08	×	*c*	*bc*	*a*	*b*	
09	y⇄( )	*c*	*bc*	*a*	*b*	
0a	f	*c*	*b*	*a*	*bc*	
0b	STOP	*d*	*b*	*a*	*bc*	Enter *d*
0c	×	*d*	*bd*	*a*	*bc*	
0d	↑ ROLL	*a*	*d*	*bd*	*bc*	
10	×	*a*	*ad*	*bd*	*bc*	
11	f	*bc*	*ad*	*bd*	*bc*	Recall *bc*
12	+	*bc*	*ad+bc*	*bd*	*bc*	
13	CLEAR x	0	*ad+bc*	*bd*	*bc*	
14	ROLL ↓	*ad+bc*	*bd*	0	*bc*	
15	x⇄y	*bd*	*ad+bc*	0	*bc*	
16	END	*bd*	*ad+bc*	0	*bc*	Answers

Each of the following is a program segment rather than a complete program. Therefore, we omit the step numbers. We have traced the first segment.

KEY	X	Y	Z	a	b
y⇄( )	1	2	3	25	−36
a	1	25	3	2	−36
y⇄( )	1	25	3	2	−36
b	1	−36	3	2	25
y⇄( )	1	−36	3	2	25
a	1	2	3	−36	25

The net result of the six steps is to exchange the content of a and the content of b, leaving X, Y, and Z unchanged. Now you try one. Complete the following trace.

KEY	X	Y	Z	a	REMARKS
	1	2	3	4	Initial contents
ROLL ↓					
y⇄( )					
a					
↑ ROLL					
x⇄y					
↑ ROLL	4	2	1	3	Final contents

The following definitions will prove useful.

CLEAR This key clears the X, Y, Z, e, and f registers to zero.

The e and f registers are not displayed. If you want to see the contents of the e and f registers, press the RCL key.

RECALL

RCL Recalls the content of the e and f registers. The content of e is copied into Y and the content of f is copied into X. The original contents of X and Y are destroyed.

BEFORE PRESSING		AFTER PRESSING	
1.	Z	1.	Z
2.	Y	33.	Y
3.	X	25.	X
33.	e	33.	e
25.	f	25.	f

Remember these things:

1. We can copy the content of X into any memory register 0 through f.
2. We can copy the content of Y into any memory register 0 through f.
3. We can *recall* the content of any of the registers a through f into X. We *cannot,* however, recall the content of any of the registers 0 through 9. Pressing any of these keys causes the number itself to be put into X.
4. We can *exchange* the content of Y with the content of any memory register. In fact, this is the only way to retrieve the content of one of the registers 0 through 9.
5. Be careful not to store a number into a register that contains program steps. If you do this, you will erase the program steps.
6. Pressing the CLEAR key will clear not only the X, Y, and Z registers, but also will clear whatever is stored in the e and f registers.

Section 2-4 Loops

The following program causes the computer to generate and momentarily display consecutive positive integers:

1, 2, 3, etc.

Each integer is displayed for approximately 1/8 second.

STEP	KEY	REMARKS
00	CLEAR	
01	1	
02	↑	
03	PAUSE	The PAUSE key is on the right side of the keyboard.
04	+	
05	GO TO ( )( )	
06	0	Go around again.
07	3	
08	END	

Under control of this program, the computer will count rapidly. Each counting number is displayed in the Y register.

TO RUN THE PROGRAM

1. Press END, then press CONTINUE. The computer will count rapidly, displaying the consecutive numbers in the Y register.
2. To stop the computer, press STOP.
3. To continue the count, press CONTINUE.
4. To restart the count at 1, press STOP, then press END, and then press CONTINUE.

We will describe this program in detail. But first, perhaps you would like to try it. If the computer is available, store the program, then press END and CONTINUE. The display will begin blinking rapidly.

How good are your reflexes? Can you stop the computer exactly on 100? Try it—press END, then press CONTINUE. Watch the count in the Y register increase and press STOP at just the right moment, thus stopping the computer with 100 in the Y register.

Here is a modification of the preceding program, we have changed the PAUSE instruction to a CONTINUE.

STEP	KEY	TO RUN THE PROGRAM
00	CLEAR	1. PROGRAM - RUN switch RUN.
01	1	
02	↑	2. Press END, then press CONTINUE. The screen will go blank and the computer will count like crazy. To see the count, press STOP. To continue the count, press CONTINUE. To restart the program following a STOP, press END, then CONTINUE.
03	CONTINUE	
04	+	
05	GO TO ( )( )	
06	0	
07	3	
08	END	

Each of the preceding programs contains a loop. A loop is simply a set of instructions that is repeated. A loop has a beginning and an end. In each of the programs the loop began at step 03 and terminated with the GO TO 0, 3, instruction in steps 05, 06, 07.

This loop is an example of an unconditional loop. Once we store the program and direct the computer to begin at step 00, it will execute the instructions in steps 00, 01, and 02 once, then repeat steps 03 through 07 until we intervene by manually pressing STOP. That is, there is no way for the computer to terminate the loop. In Section 2 - 6, we will describe conditional loops, which are self-terminating.

Before we go on, here is a definition of the PAUSE instruction.

PAUSE The PAUSE instruction causes the calculator to pause for ⅛ second and then go on.

To help you understand exactly what is happening, we are going to show you a numeric trace in which we "unwrap" the loop. Our trace shows you what happens in X, Y, and Z, during the first three times through the loop.

STEP	KEY	X	Y	Z	REMARKS
00	CLEAR	0	0	0	
01	1	1	0	0	
02	↑	1	1	0	This sets the count to 1.
03	PAUSE	1	1	0	
04	+	1	2	0	
05	GO TO ( )( )	1	2	0	First time through the loop
06	0	1	2	0	
07	3	1	2	0	
03	PAUSE	1	2	0	
04	+	1	3	0	
05	GO TO ( )( )	1	3	0	Second time through the loop
06	0	1	3	0	
07	3	1	3	0	
03	PAUSE	1	3	0	
04	+	1	4	0	
05	GO TO ( )( )	1	4	0	
06	0	1	4	0	
07	3	1	4	0	Third time through the loop

The loop begins with a PAUSE and ends with a GO TO, 0, 3. At each PAUSE the computer pauses for ⅛ second, displaying the count. The counting numbers appear in the Y register and a 1 remains in the X register. For the first three times through the loop, the displays look like this:

FIRST TIME		SECOND TIME		THIRD TIME	
0.	Z	0.	Z	0.	Z
1.	Y	2.	Y	3.	Y
1.	X	1.	X	1.	X

Multiples

By changing one step in the counting program, we obtain a program to count by twos. That is, the following program generates consecutive even positive integers 2, 4, 6, ...

STEP	KEY	REMARKS
00	CLEAR	
01	2	
02	↑	
03	PAUSE	Pause for ¼ second.
04	PAUSE	
05	+	
06	GO TO ( )( )	
07	0	
08	3	
09	END	

What is the purpose of the following program? ______________________________

____

STEP	KEY
00	CLEAR
01	2
02	↑
03	PAUSE
04	×
05	GO TO ( )( )
06	0
07	3
08	END

TO RUN THE PROGRAM

Press END, then press CONTINUE

What are the first two displays?

	Z		Z
	Y		Y
	X		X

Trace the preceding program two times through the loop.

STEP	KEY	X	Y	Z	REMARKS
00	CLEAR				
01	2				
02	↑				
03	PAUSE				First time through the loop.
04	×				
05	GO TO ( )( )				
06	0				
07	3				
03	PAUSE				Second time through the loop.
04	×				
05	GO TO ( )( )				
06	0				
07	3				

If we run the following programs, what will the first four displays look like? Do not use the computer to find the results.

STEP	KEY
00	CLEAR
01	1
02	↑
03	2
04	PAUSE
05	+
06	GO TO ( )( )
07	0
08	4
09	END

FIRST DISPLAY
Z
Y
X

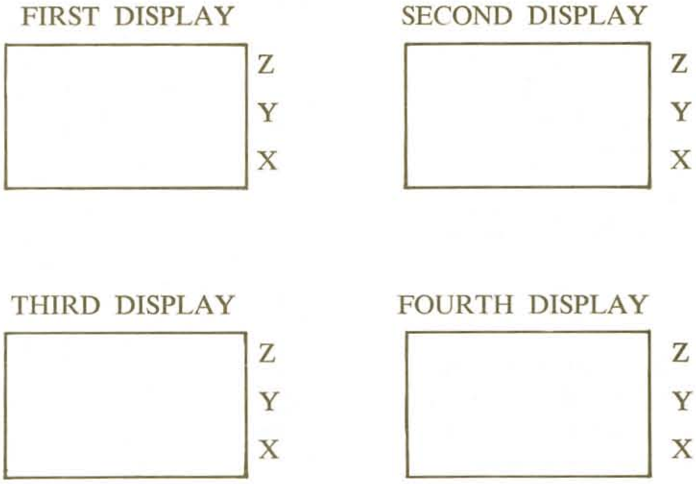
SECOND DISPLAY
Z
Y
X

THIRD DISPLAY
Z
Y
X

FOURTH DISPLAY
Z
Y
X

This one is more difficult. If we run the following program, what are the first four displays? Again, do not use the computer.

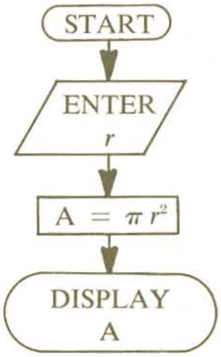
STEP	KEY
00	CLEAR
01	1
02	↑
03	×
04	PAUSE
05	$x \rightleftarrows y$
06	1
07	+
08	↓
09	GO TO ( )( )
0a	0
0b	2
0c	END



Section 2-5 Flow Charts

As our programs become more complicated, we may find flow charts useful in helping us think through the logic of our method. A flow chart is merely a graphic representation of a program or a procedure. Here is an example—it is a flow chart of a program to compute $A = \pi r^2$.

Example 1.



Every flow chart starts like this.

This is an input box.

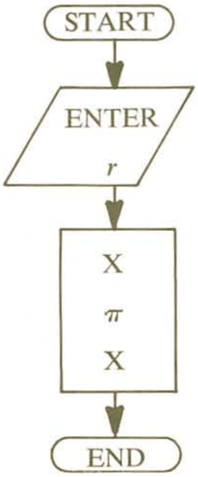
This is an assignment box. It says compute the value of πr^2.

This is an output box, or display box.

In some cases, we can relate flow chart boxes and instructions in a program for the computer. We have done this for the $A = \pi r^2$ program by actually putting the instructions in the appropriate flow chart box.

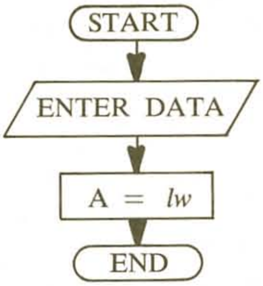
In this case our flow chart would look like this:

Example 2



STEP	KEY
00	CLEAR
01	STOP
02	↑
03	×
04	π
05	×
06	END

Check the following simplified flow chart for finding the area of a rectangle.



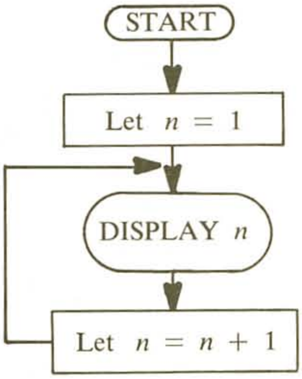
Remember flow charts should be an aid to the thinking process. They should not replace it. So keep your charts in a form which you will understand. In general though, keep the input, assignment, and output boxes the same shape as those above. We do want to keep the form so that others can understand your flow charts too.

Let's go back to the program we did in Section 2 - 4 which contained the loop and see how a flow chart for this would look.

The program was:

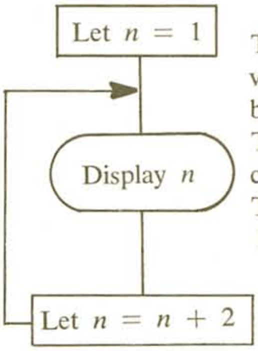
STEP	KEY
00	CLEAR
01	1
02	↑
03	PAUSE
04	+
05	GO TO ( )( )
06	0
07	3
08	END

Here's one possible flow chart.



The program actually began with a CLEAR instruction. This instruction is included simply to insure that there will be no leftover data in the Z register during the execution of the program. Therefore, we did not include it in our flow chart.

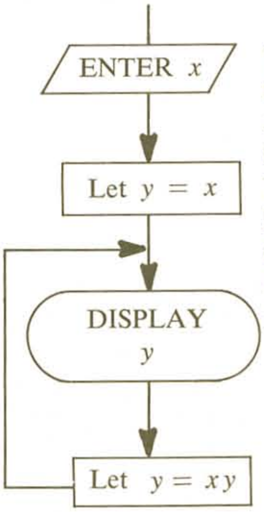
Here are some additional examples of flow charts. In each case, we have written a brief description of the process defined by the flow chart.



The initial value of n is one. This value is displayed, then increased by 2 each time through the loop. Therefore, this procedure generates consecutive odd positive integers. That is, n takes on the values 1, 3, 5, ...

How would you change this flow chart so that it displays even numbers? ______________________________

____



This is a flow chart of a procedure to generate and display powers of x, where the value of x is entered from the keyboard. We first assign y the value of x. Each time through the loop, we display the value of y, and then multiply it by the value of x. Thus, y will take on the values

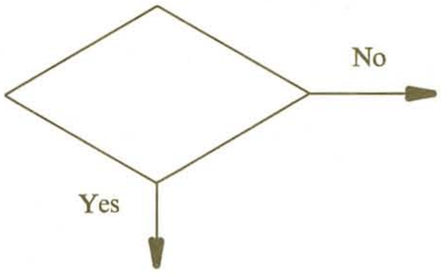
$$x, x^2, x^3 \ . \ . \ . \ .$$

A flow chart should be written before a program is written. Remember it is used to organize your thinking.

Section 2-6 Decision Making

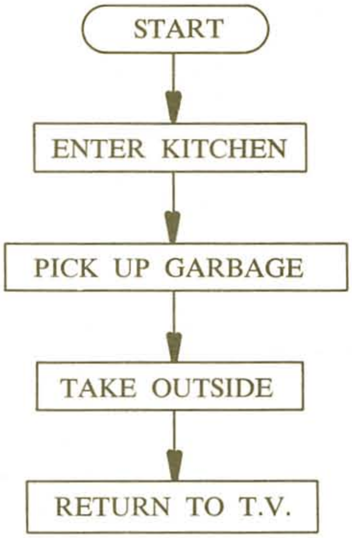
Decisions: Our day is full of decisions. When you wake up, it's "Do I really feel well enough to go to school?" Assuming the answer is yes, then "What do I wear?" After this decision is made, we continue with numerous others. There is one thing in common with all these decisions. That is, most of them must be answered before continuing. Such is the case when writing a program for the HP 9100A. We can ask the HP 9100A to make yes-no decisions.

Let's go back to the flow charts we were using in the last section and see how our decision-making is handled. The following is a decision symbol for a flow chart.

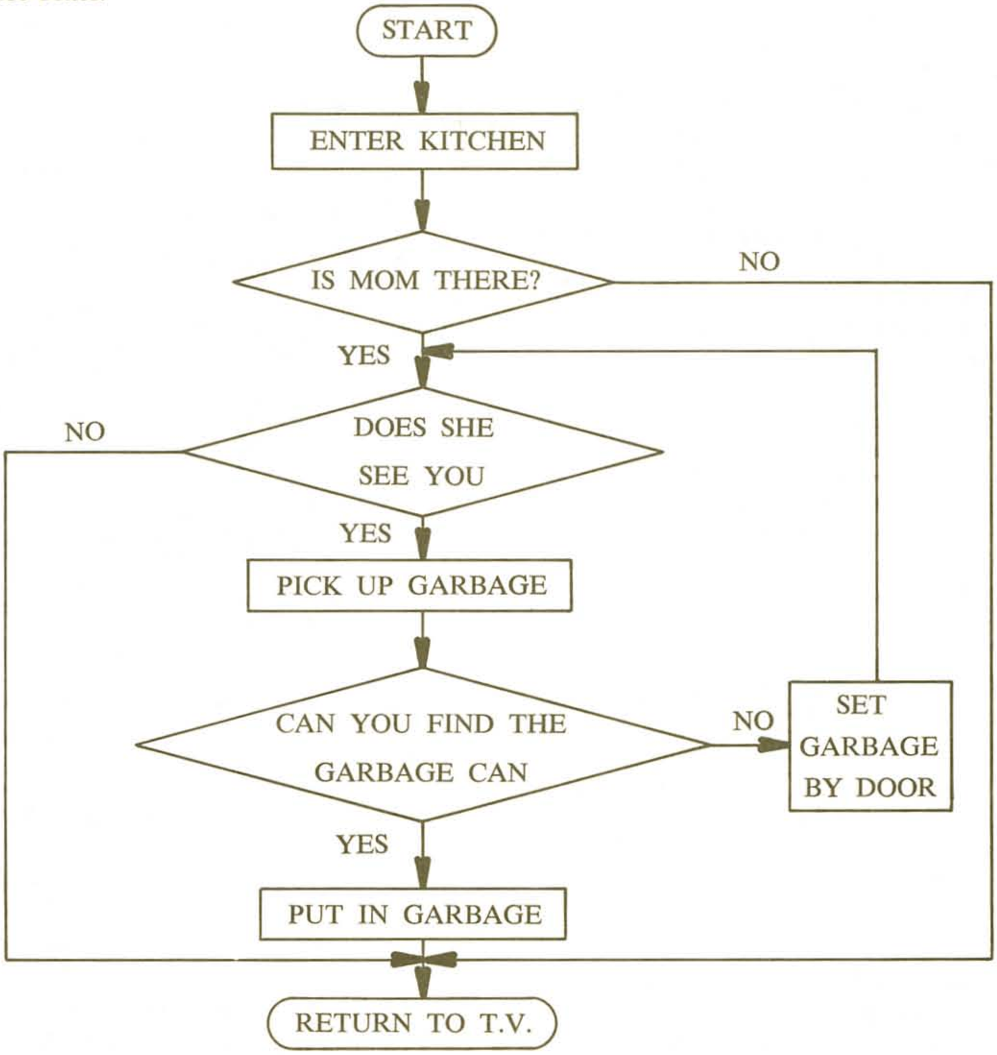


Note the two arrows coming from the decision symbol. If the question in the decision symbol is yes, the HP 9100A will follow along the path indicated by the "yes arrow." If the answer is no, the HP 9100A will follow the path of the "no arrow."

Let's make a flow chart for an everyday event, such as taking out the garbage. An initial flow chart should be as simplified as possible. It should be used to initially organize your thinking.



But, as you probably know, it's not so simple as the above flow chart indicates. Most of those boxes can be broken into many more boxes.

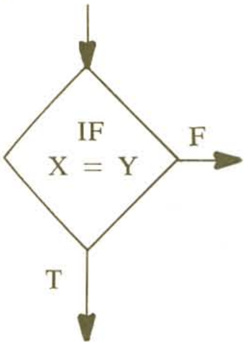


Note all the different paths which could be taken, depending on the answers to the decisions.

Look at the right section of the keyboard of the HP 9100A. We will describe the "IF" keys. Using these keys, we can write a program in which simple decisions are made automatically by the computer as it executes the program. The IF keys are shown below.

IF $x<y$ IF $x=y$ IF $x>y$

Each of these keys causes the computer to follow one of two possible paths, depending on whether a specified condition is TRUE or FALSE. Let's first illustrate this idea by means of a flow chart symbol. In flow charts, we have used diamond-shaped boxes to indicate a decision point in the flow chart.



This is a decision box. There is one arrow leading into the box and two arrows leading out of the box. If the condition expressed in the box is TRUE, follow the T path; if the condition is FALSE, follow the F path.

In other words, if the content of X is equal to the content of Y, follow the TRUE (T) path; otherwise ($X \neq Y$), follow the FALSE (F) path.

Each of the following displays shows the content of the X register and the content of the Y register. In each case, we have shown the path followed out of the $X = Y$ decision box.

```
-6.   Z
 5.   Y
 4.   X
```

Path F

```
7.   Z
3.   Y
3.   X
```

Path T

```
0.   Z
0.   Y
0.   X
```

Path T

Remember, the IF decision compares the content of the X register and the content of the Y register only! The content of Z is irrelevant.

For each of the following, which path will be followed out of an $X = Y$ decision box? (Answer T or F.)

```
        0.
6.02       23
6.02       23
```

Answer ______________

```
-7.
 7.
-7.
```

Answer ______________

```
2.
1.
 .999 999 999
```

Answer ______________

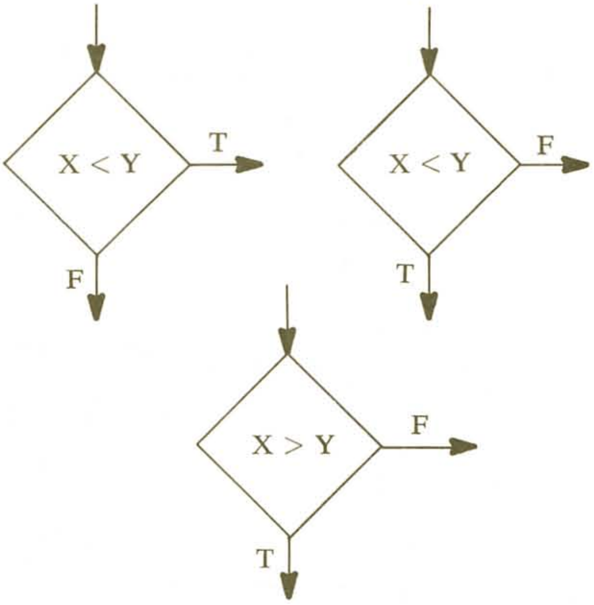
Which path will be followed under each of the following conditions?

Content of X	Content of Y	D. W. at 5 Path followed (T or F)
π *	3.14159	
0.	.00000	
$\sqrt{2}$ **	1.4142	

*Entered by pressing π

**Entered by pressing 2 $\sqrt{x}$

We use similar flow chart boxes for the $X < Y$ and $X > Y$ decisions. Here are some examples.



In each case, follow the T path if the condition is TRUE; follow the F path if the condition is FALSE.

For each of the following displays, which path will the program follow for an X < Y condition?

3.	Z
2.	Y
1.	X

Answer ______________

33.	Z
6.	Y
7.	X

Answer ______________

0.	Z
−5.	Y
−6.	X

Answer ______________

Answer the following for an X > Y condition.

0.	Z
0.	Y
2.	X

Answer ______________

0.	Z
0.	Y
−1.	X

Answer ______________

0.	Z
0.	Y
0.	X

Answer ______________

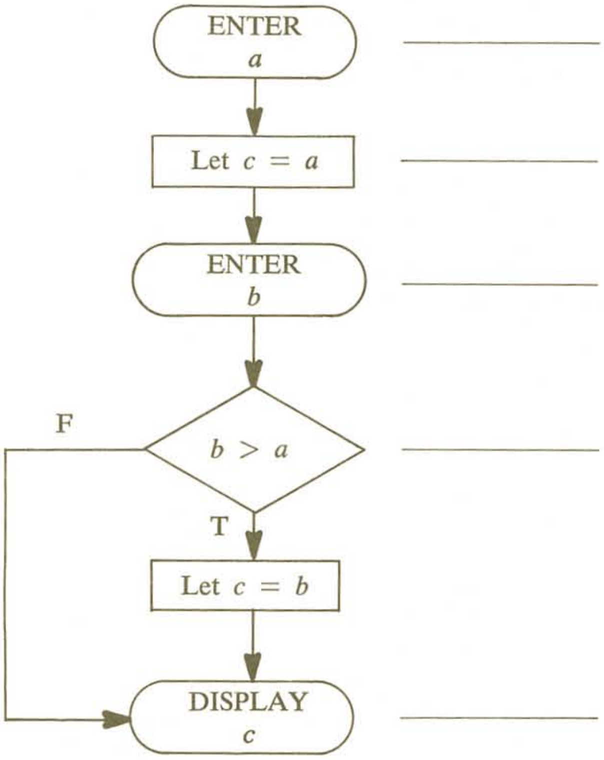
Now that you have learned something about the decisions that the computer can make, let's see what happens within a program when the conditions are met (TRUE) and when the conditions are not met (FALSE). Since it is difficult to describe the operations with complete clarity, we shall write a program in which we use the IF $x > y$ key.

The following program computes the value of c, where

$c = \text{MAX}(a, b)$ = the larger of the numbers a and b.

For example, MAX (3, 2) = 3; and MAX (−5, —2) = −2. The following flow chart illustrates our method.

FLOW CHART MAX



Enter the value of a.

As a first guess, let $c = a$.

Enter b, then compare a and b.

If $b > a$ is FALSE, then we were correct in setting $c = a$. Otherwise, we gracefully acknowledge our mistake and let $c = b$.

In either case, we end with $c = MAX\ (a, b)$.

PROGRAM MAX

STEP	KEY	X	Y	Z	REMARKS
00	↑	a	a		Let $c = a$ (c = content of Y)
01	STOP	b	a		Enter b
02	IF $x>y$	b	a		See note below
03	$x \rightleftarrows y$	a	b		Let $c = b$
04	CONTINUE	a	b		
F 05	CLEAR x	0	b		
06	↑	0	0	c	
07	END	0	0	c	Display c

Note: If the condition is TRUE, the computer executes the next two steps (Steps 03 and 04). If the condition is FALSE, the computer skips the next two steps.

TO RUN THE PROGRAM

(1) Press END
(2) Enter a, press CONTINUE
(3) Enter b, press CONTINUE

FINAL DISPLAY

c	Z
0.	Y
0.	X

Here are some numerical examples.

$a = 5,\ b = 3$

5.	Z
0.	Y
0.	X

$a = -2,\ b = -1$

−1.	Z
0.	Y
0.	X

$a = 6,\ b = 6$

6.	Z
0.	Y
0.	X

You have probably noticed that the previous program does not use a CLEAR or STOP instruction as the first step We are assuming you will not be bothered by any "junk" remaining in the X, Y, and Z registers from a previous program. Putting a CLEAR instruction at the beginning of a program will eliminate this problem—if it is a problem.

We have also eliminated the first STOP instruction in this particular program. Since most programs start with the input of some data, it becomes routine to input data at the beginning and not add this step in the program.

Let's see what happened. Step 02 is the point in the program where the computer must make the decision about the relationship between a and b. Since b is in the X register and a is in the Y register, the following question controls the decision.

> Is the content of the X register greater than the content of the Y register? That is, is $b > a$?

If the answer is yes, the calculator executes the next two steps (steps 03 and 04) then executes steps 05, 06, and 07 and stops with $c = b$.

If the answer is no, the calculator skips steps 03 and 04, then executes steps 05, 06, and 07 and stops with $c = a$.

In the following TRUE path, we note that step 03 does the actual work, but step 04 is a CONTINUE which acts as a "do nothing" instruction. It is included because every IF key must be followed by two steps to be executed when the condition is TRUE, but skipped when the condition is FALSE.

The following numeric traces show the exact path followed in PROGRAM MAX for each of two sets of values of *a* and *b*.

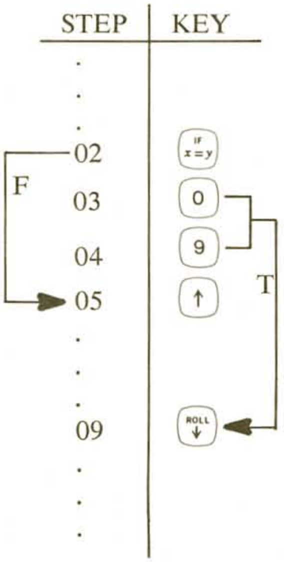
CASE 1: $a = 2,\ b = 5$

STEP	KEY	X	Y	Z
00	↑	2	2	
01	STOP	5	2	
02	IF $x>y$	5	2	
03	$x \rightleftarrows y$	2	5	
04	CONTINUE	2	5	
05	CLEAR x	0	5	
06	↑	0	0	5
07	END	0	0	5

CASE 2: $a = 33,\ b = 7$

STEP	KEY	X	Y	Z
00	↑	33	33	
01	STOP	7	33	
02	IF $x>y$	7	33	
05	CLEAR x	0	33	
06	↑	0	0	33
07	END	0	0	33

In PROGRAM MAX, the step following the IF $x=y$ key is an operational instruction—something to do. We can, however, write a different type of IF instruction, illustrated by the following program segment.



In this example, the step following the IF key is numeric. Therefore, the computer interprets it as the first character of a step number. If the condition is TRUE, the computer goes to this step next.

The third type of IF instruction is illustrated by the following program to compute *c*, where

$$c = \text{MIN}\ (a,\ b) = \text{the smaller of } a \text{ and } b.$$

For this program the final display shows *a*, *b*, and *c* in the following manner.

c	Z
b	Y
a	X

PROGRAM MIN

STEP	KEY	X	Y	Z	REMARKS
00	↑	a	a		Let $c = a$ (In Z register)
01	↑	a	a	a	
02	STOP	b	a	a	Enter b
03	$x \rightleftarrows y$	a	b	a	
04	IF $x<y$	a	b	a	If the condition is TRUE, then $c = a$.
05	0				In this case, go to step 09.
06	9				
07	↑	a	a	b	If condition is FALSE, let $c = b$.
08	↓	a	b	b	
09	END	a	b	c	

If you are not convinced that PROGRAM MIN works, examine the following numeric traces.

CASE 1: $a = 2$, $b = 5$

STEP	KEY	X	Y	Z
00	↑	2	2	
01	↑	2	2	2
02	STOP	5	2	2
03	$x \rightleftarrows y$	2	5	2
04	IF $x<y$	2	5	2
05	0	2	5	2
06	9	2	5	2
09	END	2	5	2

CASE 2: $a = 33,\ b = 7$

STEP	KEY	X	Y	Z
00	↑	33	33	
01	↑	33	33	33
02	STOP	7	33	33
03	$x \rightleftarrows y$	33	7	33
04	IF $x < y$	33	7	33
07	↑	33	33	7
08	↓	33	7	7
09	END	33	7	7

Here are some examples of both types of IF instructions.

F
IF X < Y
2
c
etc.

If X < Y is TRUE, go to step 2 c.

F
IF X = Y
+
x
etc.

If X = Y is TRUE, execute the + and x operation and continue with the following step.

F
IF X > Y
b
3
etc.

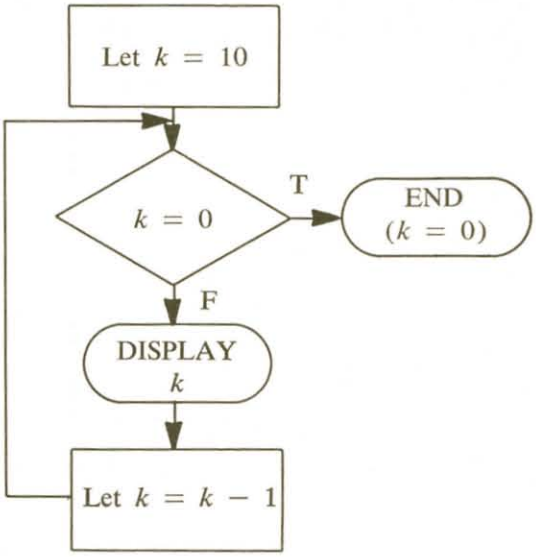
If X > Y is TRUE, go to step b 3.

The preceding examples illustrate the rule: If the step following the IF key is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c or d then the computer will interpret it as the first character of a step number. In this case, the second step must be the second character of a step number.

Section 2-7 Counting

In Section 2 - 4, we wrote programs that caused the computer to count. Under control of these programs, the computer counted, and counted, and counted . . . until someone pressed STOP. With the help of the IF instructions, we can write programs in which the computer stops automatically at a predetermined value.

Our first such program is called COUNTDOWN. Look at the flow chart first.



PROGRAM COUNTDOWN

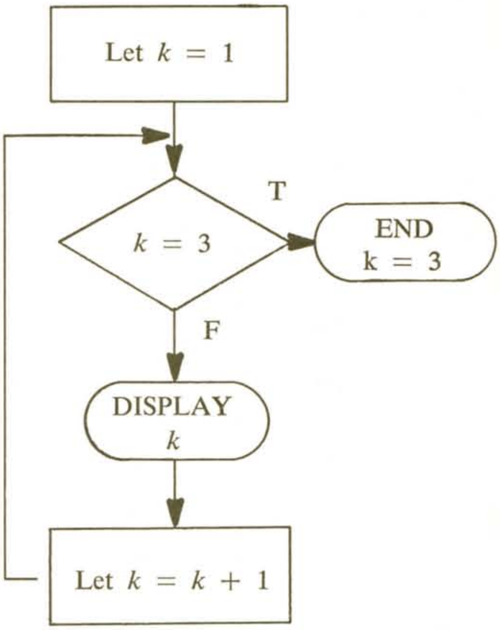
STEP	KEY	REMARKS
00	CLEAR	
01	1	
02	0	Let k = 10 (k = content of **Y**)
03	↑	
04	CLEAR x	
05	IF $x = y$	
06	1	If k = 0, go to step 10.
07	0	
F → 08	PAUSE	Display k.
09	1	Let $k = k - 1$. (Reduce the count by one)
0a	−	
0b	GO TO ( )( )	
0c	0	Go around again.
0d	4	
10	END	End of program. ($k = 0$)

Next, a program to count from 1 to 3.

PROGRAM COUNT TO 3

STEP	KEY
00	CLEAR
01	1
02	↑
03	3
04	IF $x = y$
05	0
06	d
07	PAUSE
08	1
09	+
0a	GO TO ( ) ( )
0b	0
0c	3
0d	END

FLOW CHART COUNT TO 3



Trace PROGRAM COUNT to 3.

STEP	KEY	X	Y	Z	REMARKS
00	CLEAR				
01	1				
02	↑				
03	3				First time through the loop.
04	IF $x=y$				
07	PAUSE				DISPLAY: 0. 1. 3.
08	1				
09	+				
0a	GO TO ( ) ( )				
0b	0				
0c	3				
03	3				Second time through the loop.
04	IF $x=y$				
07	PAUSE				DISPLAY: 0. 2. 3.
08	1				
09	+				
0a	GO TO ( ) ( )				
0b	0				
0c	3				
03	3				Third and last time through the loop.
04	IF $x=y$				
05	0				
06	d				
0d	END				End of Program

The final display is

0.	Z
3.	Y
3.	X

Complete the following program, making a flow chart first, directing the calculator to count to n where n is entered by the operator. We want the counting numbers to appear in the Y register.

TO RUN THE PROGRAM

1. Press END.
2. Enter n, press CONTINUE.
   The computer now counts from 1 to n.

FLOW CHART

PROGRAM COUNT TO n.

STEP	KEY	REMARKS
00	$x\rightarrow(\ )$	Store n into register a.
01	a	
02	CLEAR	
03		
04		
05		
06		
07		
08		
09		
0a		
0b		
0c		
0d		

Section 2-8 Raising a Number to a Power

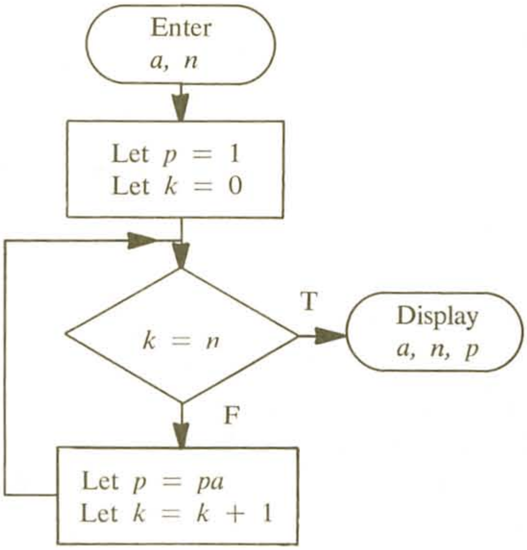
Let n be a positive integer, and let a be any real number. Then

$$a^n = \overset{n \text{ factors}}{a \cdot a \cdot a \cdot \cdot \cdot a}$$

If $a \neq 0$, then

$$a^0 = 1.$$

Below is a flow chart of a program to compute the value of a^n for values of a and n.



Let's unwrap the loop and trace the flow chart. In the trace, we show the values assigned to each variable (a, n, p, k) after the flow chart statement on the same line has been executed. We will trace the flow chart for $a = 5$, $n = 3$.

FLOW CHART STATEMENT	a	n	p	k
ENTER a, n	5	3		
LET $p = 1$	5	3	1	
LET $k = 0$	5	3	1	0
$k = n$ (FALSE)	5	3	1	0
LET $p = pa$	5	3	5	0
LET $k = k + 1$	5	3	5	1
$k = n$ (FALSE)	5	3	5	1
LET $p = pa$	5	3	25	1
LET $k = k + 1$	5	3	25	2
$k = n$ (FALSE)	5	3	25	2
LET $p = pa$	5	3	125	2
LET $k = k + 1$	5	3	125	3
$k = n$ (TRUE)	5	3	125	3
DISPLAY a, n, p	5	3	125	3

Will the process defined by the flow chart produce the correct answer under the following conditions?

(a) $a \neq 0,\ n = 0$
(b) $a < 0,\ n = 0$
(c) $a < 0,\ n > 0$
(d) $a = 0,\ n = 0$

Here is a program to go along with the flow chart.

PROGRAM POWER

	STEP	KEY	X	Y	Z	a	b	REMARKS
	00	x→( )	a					Store a
	01	a	a			a		
	02	STOP	n			a		Enter n
	03	x→( )	n			a		Store n
	04	b	n			a	n	
	05	1	1			a	n	Let $p = 1$
	06	↑	1	1		a	n	
	07	CLEAR x	0	1		a	n	Let $k = 0$
	08	↑	0	k	p	a	n	
	09	b	n	k	p	a	n	
	0a	IF x=y	n	k	p	a	n	
	0b	1	n	k	p	a	n	If $k = n$, go to 18.
	0c	8	n	k	p	a	n	
	0d	a	a	k	p	a	n	Let $p = pa$
	10	↑ ROLL	p	a	k	a	n	
	11	×	p	p(new)	k	a	n	
T	12	1	1	p(new)	k	a	n	Let $k = k + 1$
	13	↑ ROLL	k	1	p(new)	a	n	
	14	+	k	k(new)	p(new)	a	n	
	15	GO TO ( )( )	k	k(new)	p(new)	a	n	Go around again.
	16	0	k	k(new)	p(new)	a	n	
	17	9	k	k(new)	p(new)	a	n	
	18	a	a	$n = k$	p	a	n	DISPLAY a, n, p
	19	END	a	n	p	a	n	

TO RUN THE PROGRAM

(1) Press END.
(2) Enter a, press CONTINUE.
(3) Enter n, press CONTINUE.

FINAL DISPLAY

```
p = a^n
   n
   a
```

EXAMPLES

$a = 5,\ n = 3$

```
125.
  3.
  5.
```

$a = 2,\ n = 12$

```
4096.
  12.
   2.
```

EXERCISES

1. PROGRAM POWER will not work if: (a) $a = 0$ and $n = 0$, (b) $n < 0$. Modify the program so that if the operator enters this type of data, the computer stops with the following display.

```
9.999 999 999     99
         n
         a
```

2. Review PROGRAM COUNTDOWN. Then write a program to compute a^n by counting down instead of counting up. Begin by drawing a flow chart.

Section 2-9 The Flag Decisions

You have probably been wondering about the keys:

The "FLAG" is simply an on-off circuit in the computer. Pressing SET FLAG turns this circuit on. We use the key when we wish to test the setting of the flag. Here is what happens when the computer comes to an IF FLAG.

FLAG OFF: Skip the next two steps.

FLAG ON: Turn the flag OFF, then:

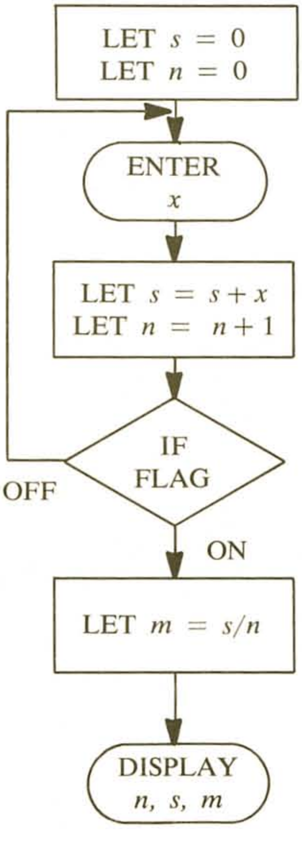
(1) If the next two steps are alphanumeric, then go to the step number that they specify.
(2) If the step following the IF FLAG is *not* alphanumeric, execute it and continue.

For example,

If the FLAG is OFF, skip two steps.	IF FLAG 2 5	If the FLAG is ON, turn it OFF and go to Step 25.
If the FLAG is OFF, skip two steps.	IF FLAG + X	If the FLAG is ON, turn it OFF and execute the two steps.

We will illustrate the use of SET FLAG and IF FLAG by means of a program to compute the arithmetic mean of a set of n numbers. We will call the numbers $x_1, x_2, \ldots, x_n$. For example, suppose we wish to compute the mean of 7, 5, 6, 8, and 6. For this set of numbers, $n = 5$, $x_1 = 7$, $x_2 = 5$, $x_3 = 6$, $x_4 = 8$ and $x_5 = 6$.

First, we draw a flow chart.



s = sum of numbers.
n = number of numbers entered so far.

x = one of the numbers. When entering last number, press SET FLAG before pressing CONTINUE.

Add x to the previous sum, obtaining a new sum. Increase the count by one.

If the FLAG is OFF, go around for another number; if the FLAG is ON, compute the mean, m.

Display the results.

UNIT 3

FUNCTIONS

Section 3-1 Logarithmic and Exponential Functions

The computer has three keys for direct evaluation of logarithmic and exponential functions.

KEY	FUNCTION
e^x	Exponential Function, Base e (e = 2.71828 . . .)
ln x	Logarithmic Function, Base e (natural logarithm).
log x	Logarithmic Function, Base 10 (common logarithm).

LOGARITHM, BASE 10

log x Computes the logarithm, base 10 (common logarithm) of the content of the X register and put the result into X.

Before Pressing log x			After Pressing log x	
1.	Z		1.	Z
2.	Y		2.	Y
3.	X	log 3 = 0.47712	.47712	X
1.	Z		1.	Z
2.	Y		2.	Y
−3.	X	log −3 = ?	−9.999 999 999 99	X
1.	Z		1.	Z
2.	Y		2.	Y
0.	X	log 0 = ?	−9.999 999 999 99	X

To prevent disasters such as the last two examples, expect the HP 9100A to compute Log x for *positive numbers only.*

LOGARITHM, BASE e

ln x Computes the logarithm, base e (natural logarithm) of the content of the X register and put the result into X.

Before Pressing ln x			After Pressing ln x	
−3.	Z		−3.	Z
5.	Y		5.	Y
2.	X	ln 2 = 0.69315	.69315	X

Now it's your turn. Compute the natural logarithm of each of the following numbers. Complete the After displays.

Before Pressing ln x			After Pressing ln x	
0.	Z			Z
0.	Y			Y
1.	X	ln 1 = ________		X
0.	Z			Z
0.	Y			Y
0.	X	ln 0 = ________		X
0.	Z			Z
0.	Y			Y
−1.	X	ln −1 = ________		X
0.	Z			Z
0.	Y			Y
2.71828	X	ln 2.71828 = ________		X

EXPONENTIAL, BASE e

e^x Raise e (2.71828 . . .) to the power (exponent) indicated in the X register.

Before Pressing e^x			After Pressing e^x		
3.	Z		3.	Z	
2.	Y		2.	Y	
1.	X	$e^1 = 2.71828$	2.71828	X	*
3.	Z		3.	Z	
2.	Y		2.	Y	
−1.	X	$e^{-1} = .36788$	.36788	X	*
3.	Z		3.	Z	
2.	Y		2.	Y	
.5	X	$e^{.5} = 1.64872$	1.64872	X	*
3.	Z		3.	Z	
2.	Y		2.	Y	
0.	X	$e^0 = 1$	1.00000	X	*

*DECIMAL DIGITS wheel set to 5.

You may wish to try the following program for manual operation for several values of *x*.

STEP	INSTRUCTION
0	CLEAR
1	ENTER *x*
2	e^x
3	ln *x*

Predict the final displays.

$x = 2$		$x = -3$	
	Z		Z
	Y		Y
	X		X

Now, try this one.

STEP	INSTRUCTION
0	CLEAR
1	ENTER x
2	ln x
3	e^x

Predict the final display

$x = 2$

	Z
	Y
	X

$x = -3$

	Z
	Y
	X

$x = 1$

	Z
	Y
	X

$x = \pi$

	Z
	Y
	X

$x = 0$

	Z
	Y
	X

Use the computer to check your answers. If there are any surprises, study the following identities.

(1) $\ln e^y = y$ for all real numbers y.

(2) $e^{\ln y} = y$ for all positive real numbers y.

Since there is no key to permit direct computation of 10^x, let's write a short program to do this calculation. For any real number x, 10^x is a positive real number. Therefore, using identity #2 above, we have

$e^{\ln 10^x} = 10^x$ (Replace y by 10^x)

$e^{x \ln 10} = 10^x$ (Since $\ln 10^x = x \ln 10$)

PROGRAM 10^x

STEP	KEY	X	Y	Z	REMARKS
00	↑	x	x		
01	↑	x	x	x	
02	1	1	x	x	Compute ln 10
03	0	10	x	x	
04	ln x	ln 10	x	x	
05	×	ln 10	x ln 10	x	
06	CLEAR x	0	x ln 10	x	Let $y = 10^x$
07	$x \rightleftarrows y$	x ln 10	0	x	
08	e^x	y	0	x	
09	↑ ROLL	x	y	0	
0a	END	x	y	0	Results

TO RUN THE PROGRAM

(1) Press END.

(2) Enter x, Press CONTINUE.

FINAL DISPLAY

0.	Z
$y = 10^x$	Y
x	X

EXAMPLES (Decimal Wheel at 5)

$x = 2$

0.	Z
100.00000	Y
2.	X

$x = 0.5$

0.	Z
3.16228	Y
.5	X

$(10^{0.5} = \sqrt{10})$

$x = 1,000,000$

0.	Z
9.999 999 999 99	Y
1.000 000 000 06	X

Overflow! x too big.

Try these:

$x = 57$

	Z
	Y
	X

$x = -2$

	Z
	Y
	X

$x = 0.333333333$

	Z
	Y
	X

EXERCISES

1. The EXPONENTIAL FUNCTION, BASE 2 is defined by

   $$\{(x, y) : x \text{ is a real number and } y = 2^x\}.$$

   Write a program to compute the value of $y = 2^x$ for an input value of x.

2. The EXPONENTIAL FUNCTION, BASE b is defined by

   $$\{(x, y) : b > 0,\ b \neq 1 \text{ and } y = b^x\}.$$

   Write a program to compute the value of $y = b^x$ for input value of b and x. The final display should look like this:

$y = b^x$	Z
x	Y
b	X

3. If we invest P dollars at r percent annual interest where the interest is compounded n times per year, then the amount A accumulated in t years is

   $$A = P\left[1 + \frac{r}{n}\right]^{nt}$$

   Write a program to compute the value of A, given values of P, r, n and t.

Section 3-2 Trigonometric Functions

We can also use the computer to evaluate the trigonometric functions SINE, COSINE and TANGENT. Here are the keys.

sin x Computes the sine of the content of X and put the result into X.

cos x Computes the cosine of the content of X and put the result into X.

tan x Computes the tangent of the content of X and put the result into X.

The content of X can be given either in degrees or radians. If we wish to use degrees, we set the DEGREES-RADIANS switch to DEGREES; if we want radians, we set the switch to RADIANS. For the following examples, the switch is set to DEGREES and the DECIMAL DIGITS wheel is on 5.

Before Pressing sin x

0.	Z
1.	Y
30.	X

sin 30° = 0.50000

After Pressing sin x

0.	Z
1.	Y
.50000	X

Before Pressing cos x

0.	Z
1.	Y
30.	X

cos 30° = 0.86603

After Pressing cos x

0.	Z
1.	Y
.86603	X

Before Pressing tan x

0.	Z
1.	Y
30.	X

tan 30° = 0.57735

After Pressing tan x

0.	Z
1.	Y
.57735	X

Set the DEGREES-RADIANS switch to DEGREES and the D. W. at 9 and complete the following table. Compare your results with those in a book of tables.

	70°	165°	5000°
SINE			
COSINE			
TANGENT			

What happens on the HP 9100A if you try to compute the following on the computer?

(1) tan 90° ____________________

(2) tan (−90°) ____________________

(3) tan 270° ____________________

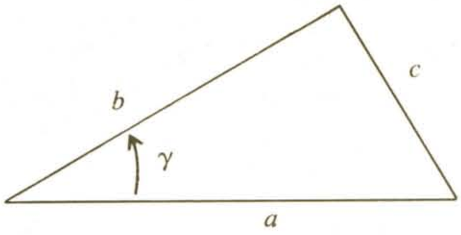
Now set the DEGREES-RADIANS switch to RADIANS and complete the table below.

	1 radian	2 radians	π radians
SINE			
COSINE			
TANGENT			

Again, use the computer to complete the table.

	27°	3.6 radians	3333°
SECANT			
COSECANT			
COTANGENT			

Let's compute the area of the following triangle.



We will measure angle γ and the lengths of sides a and b.

$$A = \tfrac{1}{2} ab \sin \gamma$$

STEP	KEY	X	Y	Z	REMARKS
00	↑	a	a		
01	STOP	b	a		Enter b
02	×	b	ab		
03	STOP	γ	ab		Enter γ
04	sin x	$\sin \gamma$	ab		
05	×	$\sin \gamma$	$ab \sin \gamma$		
06	2	2	$ab \sin \gamma$		
07	÷	2	A		$A = \tfrac{1}{2} ab \sin \gamma$
08	CLEAR x	0	A		
09	↑	0	0	A	Optional, simply used to provide a neat display.
0a	END	0	0	A	

TO RUN THE PROGRAM

(1) Press END.
(2) Enter a, press CONTINUE.
(3) Enter b, press CONTINUE.
(4) Enter γ, press CONTINUE.

FINAL DISPLAY

A	Z
0.	Y
0.	X

Before entering γ, you must set the DEGREES-RADIANS switch. If your value of γ is expressed in degrees, set the switch to DEGREES. If your value of γ is given in radians, set the switch to RADIANS.

Set the switch to DEGREES and do the following exercises. Complete the final display for each set of input data.

$a = 1, b = 2, \gamma = 30°$ $a = 2, b = 3, \gamma = 135°$

	Z		Z
	Y		Y
	X		X

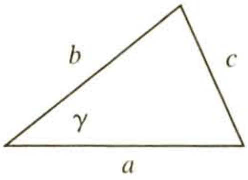
Set the switch to RADIANS and complete the following.

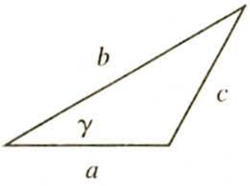
$a = 1,\ b = 2, \gamma = 1^{\text{radians}}$ $a = 2,\ b = 3, \gamma = 2.5^{\text{radians}}$

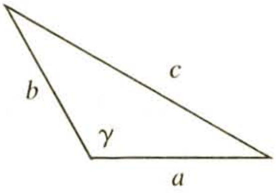
	Z		Z
	Y		Y
	X		X

EXERCISES

1. Write a program to compute the length of side c, given the measure of the angle γ and the lengths of side a and b. (see diagram)







2. Write a program to compute the area of a regular polygon with n sides given:
   (a) the value of n and the radius r of the inscribed circle.
   (b) the value of n and the radius R of the circumscribed circle.
   (c) the value of n and the length of one side.

If you have studied the arc trigonometric functions, you may be pleased to learn how to use the computer to evaluate these functions. Each of the functions can be evaluated with two keystrokes.

arc ▼ **sin x** Computes the arc sine of the content of the X register and place the result into X.

Before Pressing arc ▼ sin x   After Pressing arc ▼ sin x

Arc sine 0.5 = 30°

3.	Z	3.	Z
2.	Y	2.	Y
.5	X	30.0000	X

Arc sine 0.5 = $\pi/6$

3.	Z	3.	Z
2.	Y	2.	Y
.5	X	.52360	X

In the first example, the DEGREE-RADIANS switch was set to DEGREES, and in the second example it was set at RADIANS.

arc ▼ **cos x** Computes the arc cosine of the content of the X register and put the result into X.

arc ▼ **tan x** Computes the arc tangent of the content of the X register and place the result into X.

Set the DEGREES-RADIANS switch to RADIANS and use the computer to complete the table.

x	ARC SIN x	ARC COS x	ARC TAN x
0.			
0.1			
−0.1			
0.99			
−0.99			
1.			
−1.			
1.01			
−1.01			
5000.			

Again we wonder about that red light!

If you continue experimenting, you will find that the ARC SIN and ARC COS operations will work properly only if

$$-1 \leqq x \leqq 1.$$

If $x > 1$ or $x < -1$, pressing these keys will cause the red error light to come on. If you have studied the inverse trigonometric functions, you will know why this happens. You may also have noticed that all of the ARC functions give principal values. For example, the arc sine and arc tangent result is always in the range,

$$-90^\circ \leqq \text{result} \leqq 90^\circ \text{ (switch set to DEGREES)}$$

or

$$-\frac{\pi}{2} \leqq \text{result} \leqq \frac{\pi}{2} \text{ (switch set to RADIANS)}$$

EXERCISES

1. Run the following programs for manual operation for each value of θ.

STEP	INSTRUCTION
0	CLEAR
1	ENTER θ
2	SIN x
3	ARC
4	SIN x

STEP	INSTRUCTION
0	CLEAR
1	ENTER θ
2	COS x
3	ARC
4	COS x

STEP	INSTRUCTION
0	CLEAR
1	ENTER θ
2	TAN x
3	ARC
4	TAN x

Values of θ: 30°, 45°, 90°, 150°, 210°, 300°, 360°, 5000°, −60°, −90°.

Is the computer operation arc sin x the inverse of the computer operation sin x? Is arc cos x the inverse of cos x? Is arc tan x the inverse of tan x?

Can we restrict the range of θ so that arc sin x is the inverse of sin x for all values of θ in this range? If so, what is the restriction?

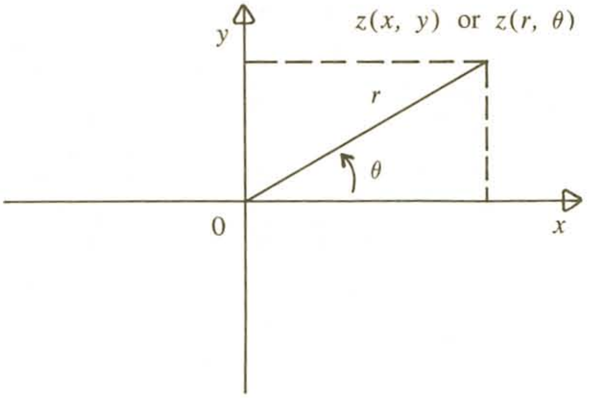
2. Under what conditions will sin x be the inverse of arc sin x? Cos x the inverse of arc cos x? Tan x the inverse of arc tan x?

Section 3-3 Polar Coordinates

TO POLAR The POLAR key changes coordinates from rectangular to polar form.

TO RECT This key changes coordinates from polar to rectangular form.

Are you familiar with these two ways of representing vectors and complex numbers? For example, let z be a complex number. We can represent z by the ordered pair (x, y) in rectangular Cartesian coordinates or by (r, θ) in polar coordinates, as shown below.



For example, the complex number $1 + i$ can be represented by the rectangular coordinates $(1, 1)$ or by the polar coordinates $(\sqrt{2}, 45^\circ)$ or, using radian measure,

$$(\sqrt{2}, \frac{\pi}{4}).$$

Using rectangular coordinates, we write

$$z = x + yi$$

and using polar coordinates we write

$$z = r \cos \theta + (r \sin \theta)\, i$$

$$= r(\cos \theta + i \sin \theta)$$

where

$$r = |z| = \sqrt{x^2 + y^2}$$

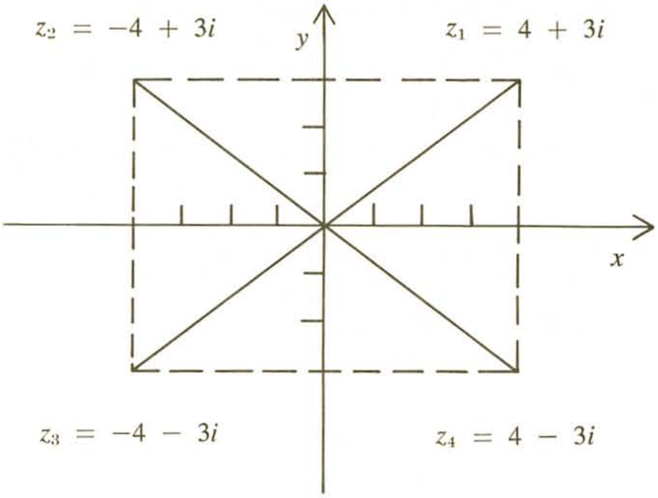
In measuring θ, mathematicians long ago established the following convention.

$$-180^\circ < \theta \leqq 180^\circ$$

We find, therefore, the following correspondences.

x	y	r	θ
1	1	$\sqrt{2}$	45°
−1	1	$\sqrt{2}$	135°
−1	−1	$\sqrt{2}$	−135°
1	−1	$\sqrt{2}$	−45°

Give r and θ for each of the following complex numbers.



For z_1, r = __________ and θ = ______________ .
For z_2, r = __________ and θ = ______________ .
For z_3, r = __________ and θ = ______________ .
For z_4, r = __________ and θ = ______________ .

We are not sure just how you did the preceding exercise, but here is an easy way,

STEP	INSTRUCTION	X	Y	Z
0	ENTER y	y		
1	↑	y	y	
2	ENTER x	x	y	
3	POLAR	r	θ	

This leads us up to a more complete definition of the TO POLAR key.

TO POLAR Let x denote the original content of the register X and y the original content of the register Y. This operation replaces x and y by r and θ, respectively, where r and θ are the polar coordinates of the point represented by x and y. That is,

$$r = \sqrt{x^2 + y^2} \quad \text{and} \quad -180° < \theta \leqq 180°$$

The converse operation is represented by the key

TO RECT Let r denote the content of the X register and θ the content of the Y register. This operation replaces X with x and Y with y where,

$$x = r \cos \theta \quad \text{and} \quad y = r \sin \theta.$$

What is the polar form of each point whose x and y coordinates are given in the table below?

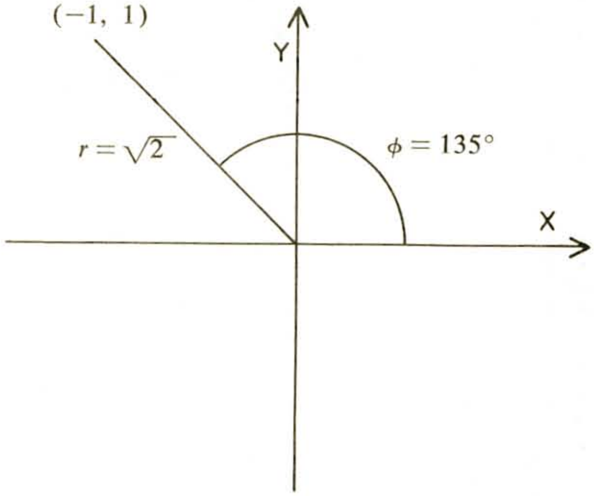
x	y	r	θ
12	5		
12	−5		
−12	5		
−12	−5		

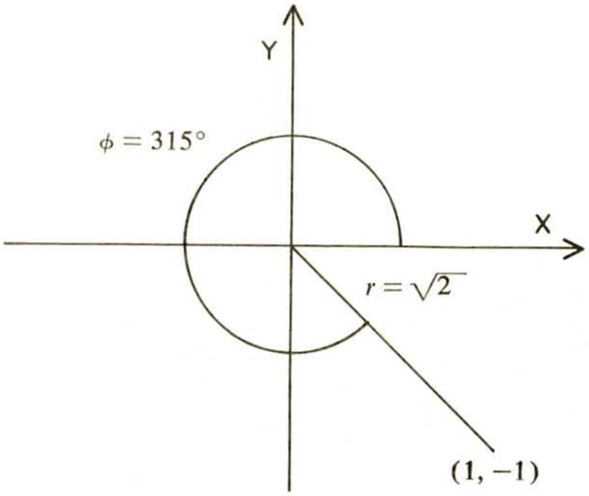
What are the rectangular coordinates of each point whose polar coordinates are given?

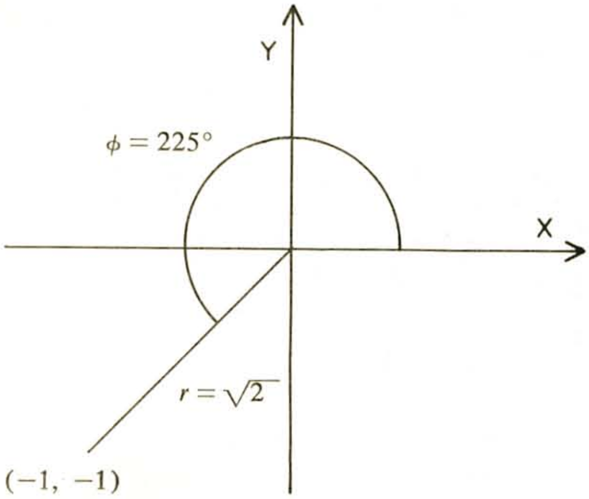
r	θ	x	y
1	30°		
1	−30°		
1	150°		
1	−150°		
1	210°		
1	315°		
2	30°		

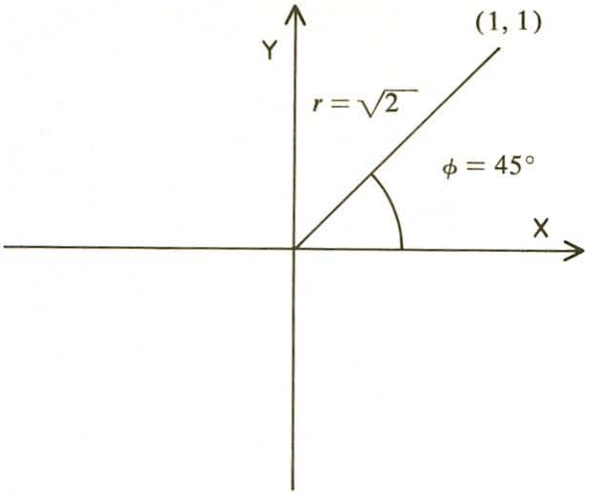
EXERCISES

1. Suppose we want to obtain polar coordinates (r, ϕ) such that $0° \leqq \phi < 360°$. That is, the angle ϕ is always measured in a counter-clockwise manner from the positive x axis. For example,









(a) What is the relationship between ϕ and θ?
(b) Write a program to compute the values of r and ϕ, given values of x and y as input data.

2. Use the computer to express each complex number in the form $a + bi$.

(a) $(1 + i)^2$ (b) $(2 + 3i)^5$ (c) $(2 - 3i)^{-2}$

3. We can express a complex number z in rectangular form as

$$z = a + bi$$

or in polar form as

$$z = r(\cos \theta + i \sin \theta).$$

Polar form is particularly useful in computing products, powers and roots of complex numbers. In computing positive integral powers of a complex number, we can apply DeMoivre's Theorem, stated below.

If $z = r(\cos \theta + i \sin \theta)$ and n is a positive integer,

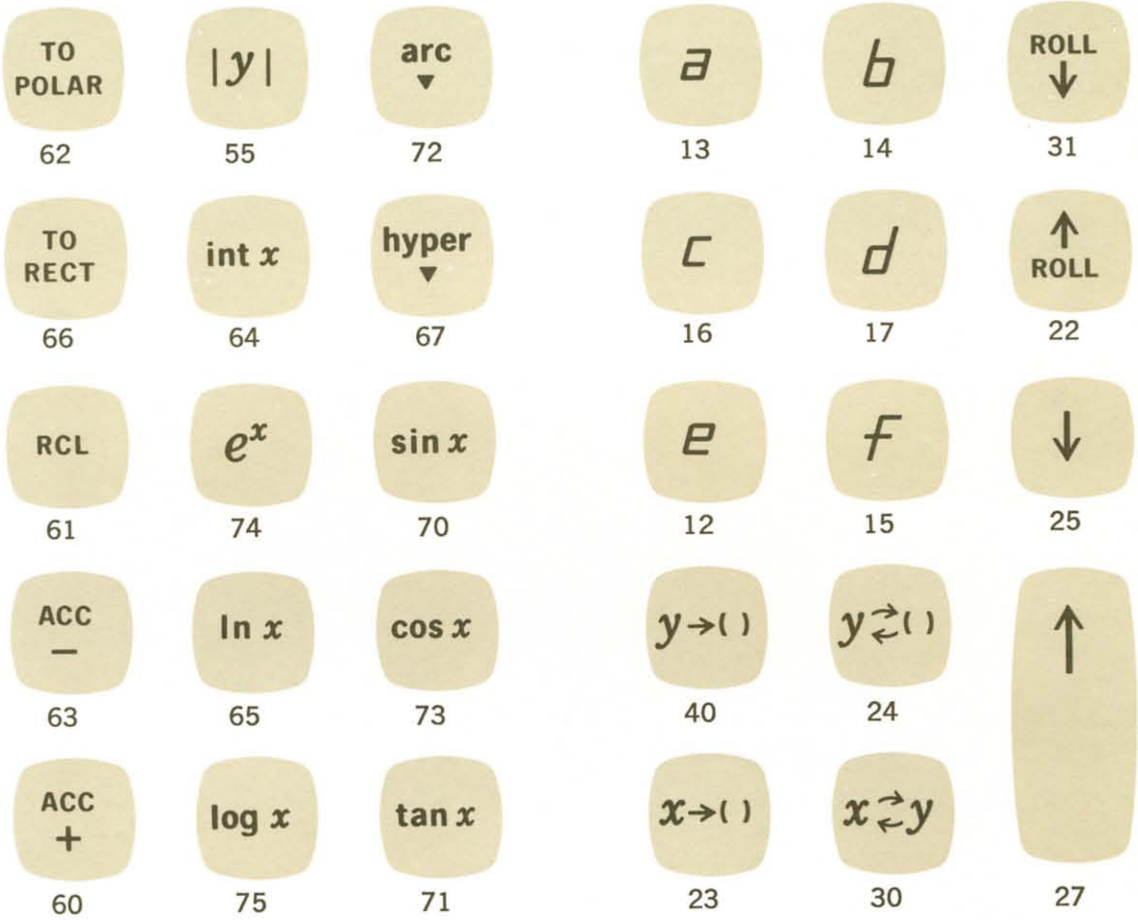
then

$$z^n = r^n (\cos n\,\theta + i \sin n\,\theta).$$

Write a program to evaluate z^n given z in rectangular form $a + bi$. We want the results in rectangular form $u + vi$ where

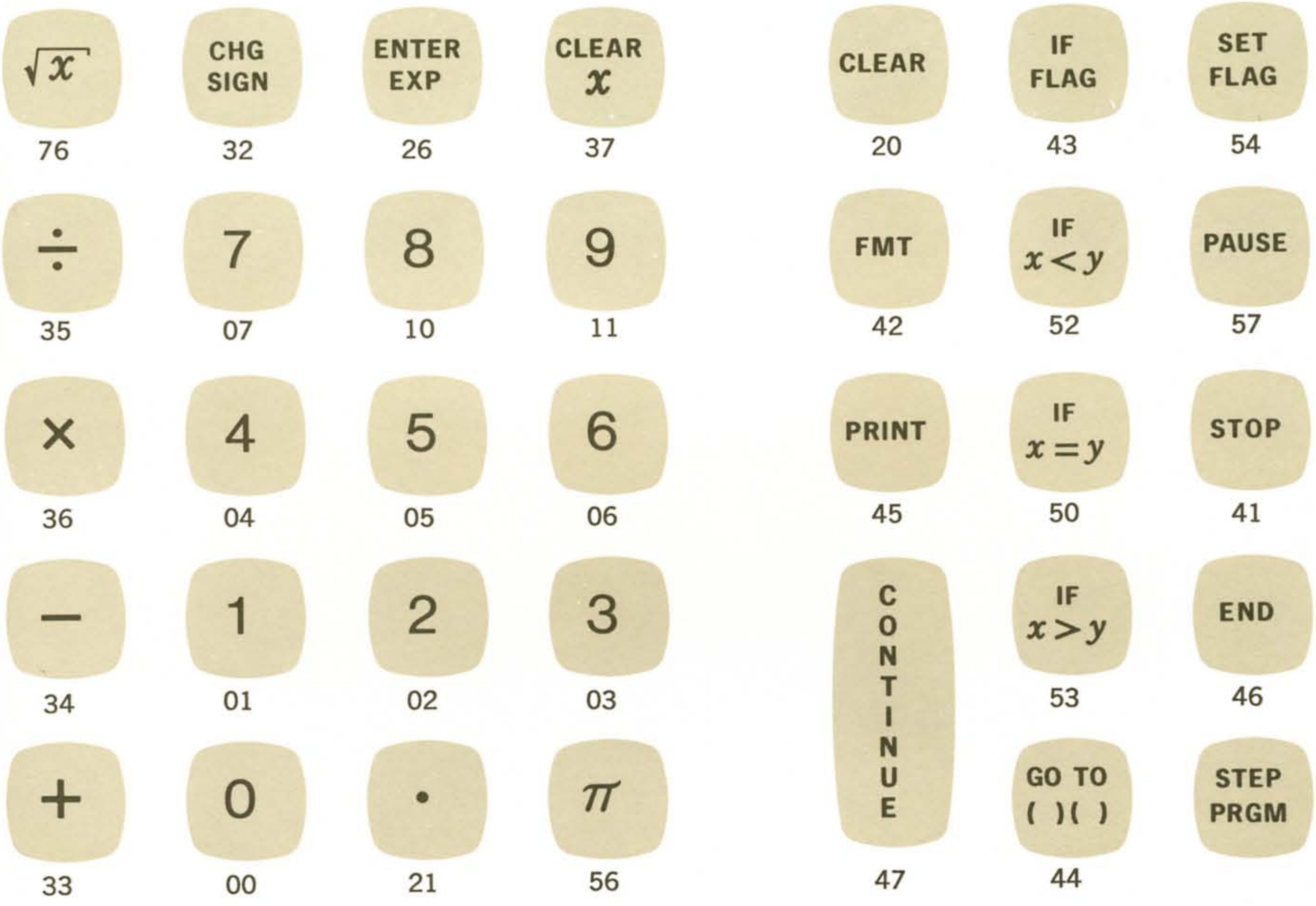
u = real part of z^n, and
v = imaginary part of z^n.

In other words, your program directs the computer to compute u and v, given a, b, and n as input data.



TO POLAR	$\lvert y \rvert$	arc ▼	a	b	ROLL ↓
62	55	72	13	14	31
TO RECT	int x	hyper ▼	c	d	↑ ROLL
66	64	67	16	17	22
RCL	e^x	sin x	e	f	↓
61	74	70	12	15	25
ACC −	ln x	cos x	$y \rightarrow (\ )$	$y \rightleftarrows (\ )$	↑
63	65	73	40	24	
ACC +	log x	tan x	$x \rightarrow (\ )$	$x \rightleftarrows y$	
60	75	71	23	30	27

KEY CODES



The number shown below each key is the (octal) instruction code. These are also shown, in numerical order, on the pull-out card at the front of the calculator.

Refer to Page 23 for an explanation of the code.

HEWLETT hp PACKARD

PART NO. 09100-90027

SEPT 1969

PRINTED IN U.S.A.