OPERATOR'S PRIMER

FRIDEN MODEL 1155
ADVANCED PROGRAMMABLE CALCULATOR

Service manual
Avail

SINGER
BUSINESS MACHINES

OPERATOR'S PRIMER

FRIDEN MODEL 1155 ADVANCED PROGRAMMABLE CALCULATOR

Publication No. 55-537
September, 1972

SINGER
BUSINESS MACHINES
2350 WASHINGTON AVE.
SAN LEANDRO, CALIF. 94577

PRINTED IN U.S.A.

LIST OF EFFECTIVE PAGES

Page	**Effective Date**
i thru vi	9/72
1-1 thru 1-6	9/72
2-1 thru 2-2	9/72
3-1 thru 3-2	9/72
4-1 thru 4-4	9/72
5-1 thru 5-3	9/72
6-1 thru 6-2	9/72
7-1	9/72
8-1 thru 8-2	9/72
9-1 thru 9-3	9/72
10-1 thru 10-5	9/72
11-1 thru 11-6	9/72
12-1 thru 12-5	9/72
13-1	9/72
14-1 thru 14-3	9/72
15-1 thru 15-4	9/72
16-1	9/72
17-1 thru 17-4	9/72
18-1 thru 18-2	9/72
19-1 thru 19-4	9/72
20-1 thru 20-3	9/72
21-1 thru 21-4	9/72
22-1 thru 22-2	9/72
23-1 thru 23-5	9/72
A-1 thru A-2	9/72
B-1 thru B-2	9/72
C-1 thru C-8	9/72
I-1 thru I-2	9/72

CONTENTS

1 GETTING STARTED

The Friden*1155 is a programmable calculator, a machine to help solve problems that require numerical calculation.

It's a calculator . . . press the keys, the 1155 does the work.

It's programmable . . . the 1155 can work on a problem automatically, under control of a stored program.

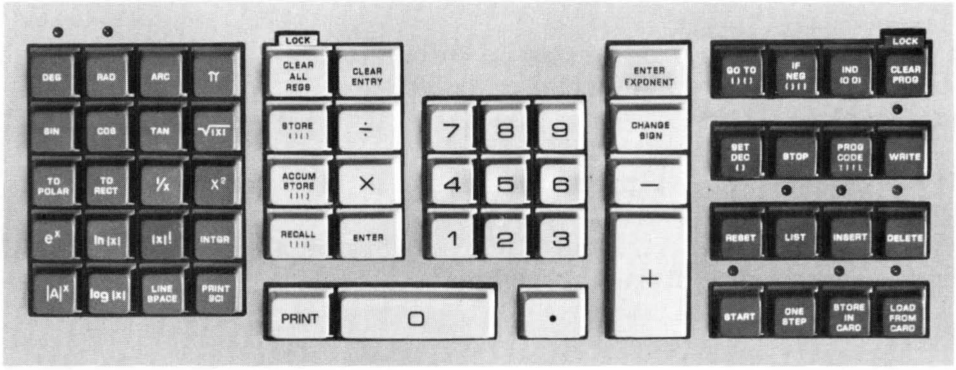
This is the 1155.

The ON-OFF switch is located on the right hand side of the machine, under the KEYBOARD. Turn the 1155 on. The PRINT wheel will spin, the decimal setting will be set at two, and .00 will print . . . the 1155 is ready to work.

---

*A trademark of The Singer Company

The KEYBOARD is the control center.



Press RESET . . . the print wheel spins and the machine line spaces. The 1155 is ready to accept instructions.

Now press LINE SPACE several times. Each time LINE SPACE is pressed the 1155 spaces the paper tape up one line.

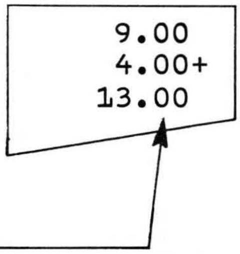
Press SET DEC ( ) then press 2

The 1155 sets the number of decimal places for printout to 2 places. Until the decimal setting is changed, numbers printed by the printer will be rounded to two decimal places.

Solve the addition problem: 9 + 4 = ?

Press the following keys in left to right order.

RESET 9 ENTER 4 + PRINT

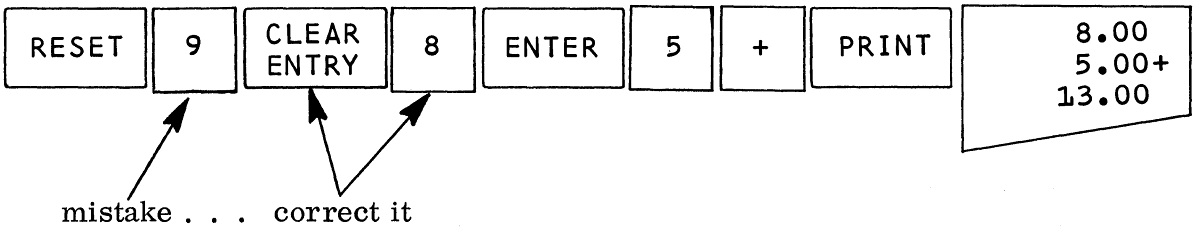


```
  9.00
  4.00+
 13.00
```

Two decimal places

If a mistake is made while keying in a number, it can be erased by pressing the CLEAR ENTRY key.

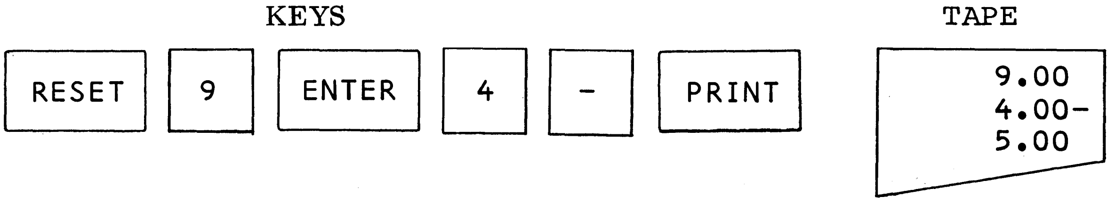
8 + 5 = ?

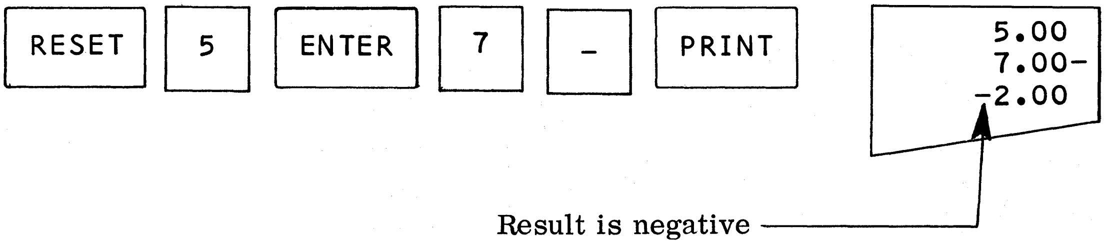


When the CLEAR ENTRY key is pressed, the print wheel spins. This is simply a signal that the 1155 has carried out the CLEAR ENTRY operation.

NOTE: If the entry of a number has been terminated by pressing the ENTER key or the + key, then pressing CLEAR ENTRY will not have the desired effect.

Solve the subtraction problems: 9 - 4 = ? and 5 - 7 = ?





Balance your checkbook?

COMMENTARY	KEY(S)	PRINTOUT
Get ready.	RESET	
Old balance.	1 2 3 . 4 5	
	ENTER	123.45
A check.	6 6 . 5 9	
	-	66.59-
New balance.	PRINT	56.86
Another check.	7 . 9 5	
	-	7.95-
New balance.	PRINT	48.91
Still another check.	2 0	
	-	20.00-
New balance.	PRINT	28.91
At last! A deposit.	1 6 7 . 0 3	
	+	167.03+
New balance.	PRINT	195.94

and so on! May your new balance always be positive!

Multiplication and division.

9 x 4 = ?

`RESET` `9` `ENTER` `4` `X`

TAPE

```
  9.00
  4.00
 36.00
```

In multiplication and division, printout of the result is automatic.

9 ÷ 4 = ?

`RESET` `9` `ENTER` `4` `÷`

TAPE

```
 9.00
 4.00
 2.25
```

3 x 4 ÷ 5 = ?

`RESET`

`3` `ENTER` `4` `X` `5` `÷`

TAPE

```
  3.00

  4.00
 12.00

  5.00
  2.40
```

1 ÷ 0 = ?

`RESET` `1` `ENTER` `0` `÷`

TAPE

```
   1.00
ERROR 5
```

The 1155 won't divide by zero. Instead, it prints ERROR 5. For a complete list of error notices, see Appendix A.

So far, all printouts have been rounded to two decimal places. Follow the next example carefully.

KEY(S)	PRINTOUT	REMARKS
RESET		
SET DEC ( ) 3		Printouts will be rounded to three decimal places.
5		
ENTER	5.000	Three decimal places.
3		
÷	3.000 1.667	Rounded to three places.
SET DEC ( ) 5		Change the decimal setting to 5 places.
PRINT	1.66667	Rounded to 5 places.
SET DEC ( ) 9		Change the decimal setting to 9 places.
PRINT	1.666666667	Rounded to 9 places.

The number of decimal places for printout can be set to any number from 0 to 9, inclusive. Simply press SET DEC and then press the numeric key for the number of places (0-9) desired.

Note that the decimal place setting determines only the number of decimal places printed. It has nothing to do with the number of places carried inside the 1155.

One more thing. When the 1155 is first turned on, the number of decimal places will be set to 2. If you want to set it to a different number of places, do so by using the SET DEC key.

2 SCIENTIFIC NOTATION

The population of the earth is about 3.6 billion people.

$$3.6 \text{ billion} = \underset{\uparrow \atop \text{"ordinary" notation}}{3{,}600{,}000{,}000} = \underset{\uparrow \atop \text{"scientific" notation}}{3.6 \times 10^{9}}.$$

The rest mass of the hydrogen atom is about

$$1.67 \times 10^{-21} \text{ kilogram.}$$

Scientific notation is simply a shorthand way of expressing very large or very small numbers. In scientific notation a number is represented by a mantissa and an exponent.

$$\underset{\text{mantissa}}{\underline{3.6}} \times 10^{9} \longleftarrow \text{exponent}$$

For more information on scientific notation, see Appendix B.

The above scientific notation is shown as it appears in math and science literature. A slightly different form is used in the 1155.

The $\boxed{\text{PRINT SCI}}$ key instructs the 1155 to print in scientific notation.

For example,

$\boxed{\text{RESET}}$ $\boxed{1}$ $\boxed{0}$ $\boxed{0}$ $\boxed{0}$ $\boxed{\text{PRINT SCI}}$

```
1.00000000000E+03
```

$$\underset{\uparrow \atop \text{mantissa}}{\underline{\texttt{1.00000000000}}}\texttt{E}\underset{\uparrow \atop \text{exponent}}{\underline{\texttt{+03}}}$$

To obtain more examples of 1155 scientific notation:

- Press RESET
- Key in a number, using ordinary notation
- Press PRINT SCI

Numbers can also be entered in scientific notation.

The ENTER EXPONENT key is used prior to keying in an exponent.

RESET	3	.	6	ENTER EXPONENT	9	PRINT SCI	3.60000000000E+09

A negative exponent is entered as follows.

RESET								
1	.	6	7	ENTER EXPONENT	-	2	1	
PRINT SCI								1.67000000000E-21

Internally, the mantissa of a non-zero number is a 13-digit number between 1 and 9.999999999999 or, in case of a negative number, between -1 and -9.999999999999. During printout the mantissa is rounded to 12 digits.

One more example. The population of the U.S. is about 205 million people.

RESET	2	0	5	ENTER EXPONENT	6	PRINT SCI	2.05000000000E+08

The number entered was 205 x 10^6.

The 1155 converted the mantissa to 2.05000000000 and adjusted the exponent to +08.

3 REGISTERS

The 1155 has two working registers in which numbers are stored and arithmetic is performed.

UPPER REGISTER (U) ☐

LOWER REGISTER (L) ☐

The following example illustrates the interaction of the registers.

PROBLEM: 9 + 4 = ?

KEY(S)	L	U	COMMENTARY
RESET	0	0	Both registers are set to zero.
9	9	0	When a number is keyed in, it is placed in L.
ENTER	9	0	ENTER terminates the entry of a number and causes it to be printed.
4	4	9	When a new number is entered in L, the previous number is bumped up into U.
+	13	0	The number that was in L is added to the number that was in U and the result is put into L. Then U is cleared to zero.
PRINT	13	0	Print the number that is in L. End of program.*

---

*A program is a sequence of keystrokes to solve a problem. The above program is executed manually.

The following example shows what occurs in the working registers and also what is printed on the tape as a problem is solved.

PROBLEM: $\frac{75 + 68 + 83}{3} = ?$

First: SET DEC ( ) 5

KEY(S)	L	U	PRINTOUT
RESET	0	0	
7 5	75	0	
ENTER	75	0	75.00000
6 8	68	75	
+	143	0	68.00000+
8 3	83	143	
+	226	0	83.00000+
PRINT	226	0	226.00000
3	3	226	
÷	75.333...3	0	3.00000 75.33333

Inside: 13 significant digits. Printout: rounded to 5 decimal places.

4 SQUARE, SQUARE ROOT, RECIPROCAL

The $\boxed{X^2}$ key is used in the following manner.

$7^2 = ?$	RESET	7	X^2		7.00000 49.00000
$(-5)^2 = ?$	RESET	5	CHANGE SIGN	X^2	-5.00000 25.00000

NOTE: To enter a negative number, first enter the digits of the number, then press the CHANGE SIGN key. The CHANGE SIGN key instructs the 1155 to change the sign of the number in L from - to + or from + to -.

The $\boxed{\sqrt{\lvert X \rvert}}$ key instructs the 1155 to compute the square root of the absolute value of the number and is used in the following manner.

$\sqrt{16} = ?$	RESET	1	6	$\sqrt{\lvert X \rvert}$	16.00000 4.00000
$\sqrt{2} = ?$	RESET	2	$\sqrt{\lvert X \rvert}$		2.00000 1.41421
$\sqrt{-4} = ?$	RESET	4	CHANGE SIGN	$\sqrt{\lvert X \rvert}$	-4.00000 2.00000

The $\boxed{\text{1/X}}$ key (reciprocal key) is used in the following manner.

$\frac{1}{2} = ?$	RESET	2	1/X	2.00000 .50000
$\frac{1}{3} = ?$	RESET	3	1/X	3.00000 .33333
$\frac{1}{6} = ?$	RESET	6	1/X	6.00000 .16667
$\frac{1}{0} = ?$	RESET	1/X		ERROR 8

The number 0 (zero) does not have a reciprocal . . . the 1155 prints an error notice. For a complete list of error notices, see Appendix A.

The following examples combine the use of the SQUARE, SQUARE ROOT, and RECIPROCAL keys.

$\sqrt{\frac{1}{2}} = ?$	RESET	2	1/X	$\sqrt{\lvert X \rvert}$		2.00000 .50000 .70711
$\frac{1}{\sqrt{2}} = ?$	RESET	2	$\sqrt{\lvert X \rvert}$	1/X		2.00000 1.41421 .70711
$\left(\frac{1}{4}\right)^2 = ?$	RESET	4	1/X	X^2		4.00000 .25000 .06250
$\frac{1}{4^2} = ?$	RESET	4	X^2	1/X		4.00000 16.00000 .06250
$\left(\frac{1}{\sqrt{5}}\right)^2 = ?$	RESET	5	$\sqrt{\lvert X \rvert}$	1/X	X^2	5.00000 2.23607 .44721 .20000

The following examples illustrate what happens internally when the SQUARE, SQUARE ROOT and RECIPROCAL keys are pressed.

KEY	L	U	COMMENTARY
RESET	0	0	
7	7	0	
X^2	49	0	The X^2 key instructs the 1155 to compute the square of the number in L, and put the results into L.
RESET	0	0	
1 6	16	0	
$\sqrt{\lvert X \rvert}$	4	0	The $\sqrt{\lvert X \rvert}$ key instructs the 1155 to compute the square root of the number in L and put the result into L.
RESET	0	0	
2	2	0	
1/X	.5	0	The 1/X key instructs the 1155 to compute the reciprocal of the number in L and put the result into L.

The X^2, $\sqrt{|X|}$ and 1/X keys do not affect the upper register. It remains unchanged.

$$\sqrt{3^2 + 4^2} = ?$$

KEY(S)	L	U	PRINTOUT
RESET	0	0	
3	3	0	
X^2	9	0	3.00000 9.00000
4	4	9	
X^2	16	9	4.00000 16.00000
+	25	0	+
PRINT	25	0	25.00000
$\sqrt{\lvert X \rvert}$	5	0	5.00000

For further practice, use the 1155 to verify each of the following calculations.

$$\sqrt{7^2 + 8^2} = 10.63015 \qquad \frac{1}{\sqrt{2^2 + 3^2}} = .27735$$

$$\sqrt{23^2 + 45^2 + 67^2} = 83.92258$$

$$1 + \frac{1}{2^2} + \frac{1}{3^2} = 1.36111$$

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \frac{1}{7} = .75952$$

5 SHORTHAND

A program is a sequence of keys to solve a problem. So far, programs have been displayed by means of pictures of the keys. From now on, however, keys will be referred to by either the actual label on the key or by a suitable mnemonic abbreviation.

For the following keys, we will use the actual label on the key, exactly as it appears on the key.

RESET   ENTER   PRINT   PRINT SCI

+   −   X   ÷

0   1   2   3   4   5   6   7   8   9   •

For the following keys, abbreviations will be used as shown.

KEY	ABBREVIATION	
LINE SPACE	SPACE	
CHANGE SIGN	OPP SIGN	(OPPOSITE SIGN)
X^2	X SQ	(X SQUARED)
$\sqrt{\|X\|}$	SQ RT	(SQUARE ROOT)
1/X	RECIP	(RECIPROCAL)
ENTER EXPONENT	ENTER EXP	

Other abbreviations will be introduced as the functions of other keys are discussed.

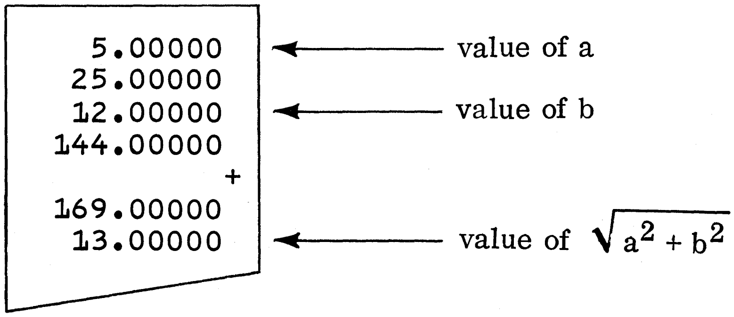
Use this new notation in a general program to compute

$$\sqrt{a^2 + b^2}$$

The program is a sequence of instructions. Each instruction tells you to press a key or to key in a number. Across from each instruction, we keep track of what is in L (LOWER REGISTER) and U (UPPER REGISTER) after the instruction on that line has been carried out.

INSTRUCTION	L	U
Press RESET	0	0
Key in the value of a	a	0
Press X SQ	a^2	0
Key in the value of b	b	a^2
Press X SQ	b^2	a^2
Press +	$a^2 + b^2$	0
Press PRINT	$a^2 + b^2$	0
Press SQ RT	$\sqrt{a^2 + b^2}$	0

The program was verified, using a = 5, b = 12. The correct answer is 13. The tape of the check run is shown below.



```
  5.00000
 25.00000
 12.00000
144.00000
        +
169.00000
 13.00000
```

For further practice, use the program to verify the following.

If a = 1.2, b = 3.45, then $\sqrt{a^2 + b^2} = 3.65274$

If a = .357, b = .246, then $\sqrt{a^2 + b^2} = .43355$

Programs can be made still less wordy by dropping the word "Press" and by using a phrase such as "Key in a" instead of "Key in the value of a." Some examples are shown below.

$$\sqrt{a^2 + b^2}$$

INSTRUCTION	L	U
RESET	0	0
Key in a	a	0
X SQ	a^2	0
Key in b	b	a^2
X SQ	b^2	a^2
+	$a^2 + b^2$	0
PRINT	$a^2 + b^2$	0
SQ RT	$\sqrt{a^2 + b^2}$	0

$$\frac{1}{p} - \frac{1}{q}$$

INSTRUCTION	L	U
RESET	0	0
Key in p	p	0
RECIP	1/p	0
Key in q	q	1/p
RECIP	1/q	1/p
-	1/p - 1/q	0
PRINT	1/p - 1/q	0

$$\sqrt{\frac{1}{xy}}$$

INSTRUCTION	L	U
RESET	0	0
Key in x	x	0
ENTER	x	0
Key in y	y	x
X	xy	0
RECIP	1/xy	0
SQ RT	$\sqrt{1/xy}$	0

VERIFY: If p = 3 and q = 7, then $\frac{1}{p} - \frac{1}{q} = .19048$

If x = 2.3 and y = .67 then $\sqrt{\frac{1}{xy}} = .80556$

6 PI

PI is the abbreviation for the $\boxed{\pi}$ key.

The number of decimal places to which π is printed is determined by the SET DEC key as shown below.

```
RESET, SET DEC, 9, PI      3.141592654

       SET DEC, 5, PI          3.14159
```

The circumference C of a circle of diameter d is: $C = \pi d$

A program to compute C is shown below.

INSTRUCTION	L	U
RESET	0	0
Key in d	d	0
PI	π	d
X	$C = \pi d$	0

The equatorial diameter of the earth is 7908 miles. What is the circumference?

```
    7908.00000  ←——— d
       3.14159
24   843.71470  ←——— C
```

C = 24844 miles, approximately.

Note that it isn't necessary to use the ENTER key to terminate the number 7908. Pressing PI terminates the preceding number string (if any) moves the content of L into U and inserts π into L. Below is another program to compute C. Use it to compute the circumference of the earth.

INSTRUCTION	L	U
RESET	0	0
PI	π	0
Key in d	d	π
X	$C = \pi d$	0

The formula for determining the area of a circle is $A = \pi r^2$. The program for this formula is:

INSTRUCTION	L	U
RESET	0	0
Key in r	r	0
X SQ	r^2	0
PI	π	r^2
X	A	0

For practice, compute $A = \pi r^2$ where r = 2, 3 and 15. The tapes should look like the following:

r = 2

```
 2.00000
 4.00000
 3.14159
12.56637
```

r = 3

```
 3.00000
 9.00000
 3.14159
28.27433
```

r = 15

```
 15.00000
225.00000
  3.14159
706.85835
```

The volume V of a sphere of radius r is computed using the formula:

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

Below is one way to program the formula to compute V.

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
Key in r	r	0	r must be entered twice
X SQ	r^2	0	
Key in r	r	r^2	
X	r^3	0	$r(r^2) = r^3$
PI	π	r^3	
X	πr^3	0	
4	4	πr^3	
X	$4\pi r^3$	0	
3	3	$4\pi r^3$	
÷	V	0	

7 A TO X

$\boxed{|A|^X}$ Computes the Xth power of the absolute value of A. The value of X is the number in L. The value of A is the number in U. The result is placed in L and U is cleared to zero.

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
Key in A	A	0	
ENTER	A	0	
Key in X	X	A	
A TO X	$\|A\|^X$	0	A TO X means $\boxed{\|A\|^X}$

Verify each of the following.

$2^{10} = 1024$

$.5^3 = .125$

$10^3 = 1000$

$10^{1.23} = 16.98244$

$1.2^{3.45} = 1.87575$

$|-2|^3 = 8$

$10^{-3} = .001$

$10^{100} = 9.99999999994E+99$

Another way to compute $V = 4\pi r^3/3$, using A TO X, is shown below.

INSTRUCTION	L	U
RESET	0	0
Key in r	r	0
ENTER	r	0
3	3	r
A TO X	r^3	0
PI	π	r^3
X	πr^3	0
4	4	πr^3
X	$4\pi r^3$	0
3	3	$4\pi r^3$
÷	V	0

Congratulations! You are the big winner on a TV show. Your prize is selected as follows.

A number between 100 and 1000 is chosen at random.
Call it N.

You then select one and only one of the following prizes.

PRIZE #1: You receive N dollars.

PRIZE #2: You receive D dollars where D is computed as follows:

$$D = 1.01^N$$

Perhaps you recognize the formula for D. It is the amount that you would receive if you invested $1 at 1% interest per day, compounded daily for N days.

The question, of course, is: For a given value of N, which prize do you take, PRIZE #1 or PRIZE #2? For each of the following values of N, indicate which prize you take by putting a check mark (√) in the appropriate column. If you wish, use the 1155 to help you decide.

N	PRIZE #1	PRIZE #2
100		
200		
300		
500		
700		
1000		

Make your choices, then read on.

Use the 1155 to help decide. First, set the number of decimal places to 2 so that the results will be rounded to the nearest penny.

Then, for each value of N, carry out the following program.

INSTRUCTION	L	U
RESET	0	0
Key in 1.01	1.01	0
ENTER	1.01	0
Key in N	N	1.01
A TO X	1.01^N	0

Do it for N = 100, 200, 300, 500, 700 and 1000. The tape should look like the following.

```
    1.01
  100.00
    2.70 - - - - - - - - - - take PRIZE #1

    1.01
  200.00
    7.32 - - - - - - - - - - take PRIZE #1

    1.01
  300.00
   19.79 - - - - - - - - - - take PRIZE #1

    1.01
  500.00
  144.77 - - - - - - - - - - take PRIZE #1

    1.01
  700.00
 1059.16 - - - - - - - - - - take PRIZE #2

    1.01
 1000.00
20959.16 - - - - - - - - - - take PRIZE #2!!!
```

We leave you with this thought . . . at what value of N does PRIZE #2 first become greater than PRIZE #1?

9 METAMORPHOSIS

The 1155 is a calculator. However, it is a calculator capable of "remembering" a program and running it automatically.

- A program is a sequence of instructions.
- Each instruction is equivalent to one or more key strokes.
- The 1155 can "remember" a program equivalent to up to 511 keystrokes.

Start with a problem.

PROBLEM: Given r, compute $A = \pi r^2$

Proceed as follows.

ONE: Write a program.

TWO: Store the program in the 1155.

THREE: List the program and proofread it.

FOUR: Run the program. Given a value of r, the 1155 computes and prints the corresponding value of $A = \pi r^2$. Repeat for each value of r.

Now, locate the following keys on the right side of the keyboard.

STOP | CLEAR PROG | WRITE | LIST | START

ONE STEP | INSERT | DELETE

ONE: Write a program.

```
SPACE
 STOP
PRINT
 X SQ
   PI
    X
PRINT
```

This is the program.
It has seven instructions.

TWO: Store the program

- Press RESET (to attract the 1155's attention)
- Press CLEAR PROG (to erase all programs stored in the 1155)
- Press WRITE (to tell the 1155 to get ready to "remember" a new program)

Write the program . . . press, in order, the keys corresponding to the instructions in the program.

LINE SPACE	STOP	PRINT	X^2	π	X	PRINT

The program is now stored in the 1155's memory.

THREE: List the program and proof-read it.

- Press RESET
- Press LIST then press +
  The 1155 prints a <u>check</u> <u>sum</u> — — — — — — — — — — ++152
- Press START
  The 1155 starts listing — — — — — — — — — — — — —

```
SPACE
 STOP
PRINT
 X SQ
   PI
    X
PRINT
```

If the tape looks like this, the program is ready to RUN.

FOUR: Run the program for r = 2, 3 and 12.

- Set the number of decimal places to 5.

  Press RESET, SET DEC, 5

- Find the top (beginning) of the program.

  Press RESET, LIST, +

The 1155 prints the checksum (++152), then stops at the top of the program, ready to list it. But don't list the program.

- Press RESET (to get out of LIST mode)
- Press START

  The 1155 starts running the program. It does a SPACE, then STOPS.

```
Key in 2, press START————————————→   2.00000
                                     12.56637

Key in 3, press START———————————→    3.00000
                                     28.17433

Key in 12, press START——————————→   12.00000
                                    452.38934
```

Key in a value of r, then press START . The 1155 will compute and print the value of A = πr^2 then space the paper tape up one line, then stop for the next value of r.

When you are finished, press RESET. This returns the 1155 to keyboard mode, but does not erase the program. Keyboard operations can be carried out or the program can be run again or a new program can be entered.

10 EDITING A PROGRAM

PROBLEM: Compute the volume V of a cylinder of radius r, height h.

$$V = \pi r^2 h$$

Use the following program.

```
SPACE
 STOP    (for a value of r)
PRINT
 X SQ
   PI
    X
 STOP    (for a value of h)
PRINT
    X
PRINT    (the answer, V)
```

Next, store the program, exactly as shown below.

RESET | CLEAR PROG | WRITE

LINE SPACE | STOP | π | X | STOP | PRINT | X | PRINT

List the program.

```
RESET, LIST, +          ++204
START                  SPACE
                        STOP
                          PI
                           X
                        STOP
                       PRINT
                           X
                       PRINT
```

Two steps were purposely omitted.

The program stored in the 1155's memory is incorrect. It must be edited in order to correct the mistakes.

To correct the mistake, the following keys will be used.

ONE STEP and INSERT

Insert the missing steps, as follows.

```
     —KEY(S)——————————————————————————————————— TAPE—

     RESET, LIST, +                              ++204

     ONE STEP                                   SPACE
     ONE STEP                                    STOP

     RESET, INSERT, PRINT, X SQ
```

The two instructions (PRINT and X^2) have been inserted following the STOP instruction. List the complete program.

```
     RESET, LIST +      new checksum----++239

     START                                      SPACE
                                                 STOP
                                                PRINT
                                                 X SQ]
                                                   PI]——— inserted steps.
                                                    X
                                                 STOP
                                                PRINT
                                                    X
                                                PRINT
```

The program is now stored correctly. Run it.

```
     RESET, LIST, +                              ++239
     RESET, START

       Key in r, press  START                  2.00000
       Key in h, press  START                  5.00000
                                              62.83185

       Key in r, press  START                  3.00000
       Key in h, press  START                 10.00000
                                             282.74334
```

and so on.

> Remember: The correct checksum for the above program is 239.

The next example uses the same program. During entry of the program, instructions are omitted in two different places.

The correct program and checksum are shown below.

```
  ++239
SPACE
 STOP
PRINT
 X SQ
   PI
    X
 STOP
PRINT
    X
PRINT
```

Enter the program, exactly as shown below.

RESET	CLEAR PROG	WRITE					
LINE SPACE	STOP	X^2	π	X	PRINT	X	PRINT

Instruct the 1155 to find the top of the program.

```
RESET, LIST, +            ++190  The checksum is incorrect.
```

List the program.

```
START                     SPACE
                           STOP
                           X SQ←PRINT left out here.
                             PI
                              X
                          PRINT←STOP left out here
                              X
                          PRINT
```

> REMEMBER: An incorrect checksum indicates that mistakes have been made in storing the program.

Correct the program by inserting

- PRINT between STOP and X SQ and
- STOP between PRINT and X.

The method for doing the inserts is shown below.

```
——— KEY(S) ——————— TAPE ———

RESET, LIST, +             ++190
ONE STEP                  SPACE
ONE STEP                   STOP

RESET, INSERT, PRINT

RESET, LIST
ONE STEP                   X SQ
ONE STEP                     PI
ONE STEP                      X

RESET, INSERT, STOP
```

List the program.

```
RESET, LIST, +             ++239 ◄────── correct
START                     SPACE
                           STOP
                          PRINT ◄─────── correct
                           X SQ
                             PI
                              X
                           STOP ◄─────── correct
                          PRINT
                              X
                          PRINT
```

The program is now stored correctly, ready to run.

SUMMARY

Press RESET	to put the 1155 in keyboard mode. This causes the 1155 to stop whatever it was doing and wait for your next move.
Press CLEAR PROG	to erase all instructions in the 1155's memory. It does not clear the register.
Press WRITE	to put the 1155 in write mode. It is now ready to "remember" program steps as you key them in. To get back to keyboard mode, press RESET.
Press LIST	to put the 1155 in list mode. It is ready to list instructions beginning with the one at which it is now stopped. If you press ONE STEP, the 1155 will list one step. If you press START the 1155 will list all remaining instructions, (i.e., to the bottom of the program). To get out of list mode, press RESET.
Press LIST, +	to cause the 1155 to compute a checksum of the program in its memory, print the checksum, then stop at the top of the program, ready to LIST. To get out of list mode, press RESET.
Press INSERT	to put the 1155 into insert mode. You can now key in as many instructions as you wish . . . they will be inserted into the program. To get out of insert mode, press RESET.
Press START	to instruct the 1155 to start listing a program (if in mode) or to start running a program (if in keyboard mode). If the 1155 has stopped because of a STOP instruction, pressing START causes it to resume automatic operation with the instruction following the STOP instruction.
Press ONE STEP	to instruct the 1155 to list one instruction (if in LIST mode) or carry out one instruction (if in program mode).

11 WRONG KEY DEPRESSED

PROBLEM: The area A of a triangle with base b and altitude h is

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

PROGRAM:

```
  ++268
SPACE
 STOP      (for value of b)
PRINT
 STOP      (for value of h)
PRINT
    X
    2
    ÷
PRINT
```

Store the program.

```
RESET, CLEAR PROG, WRITE

SPACE, STOP, PRINT, STOP, PRINT, ÷
```

mistake (should be X)

Delete the mistake.

```
RESET, DELETE, START
```

(deletes last key entered)

When DELETE was pressed, the delete light came on. When START was pressed, the delete light went off. The 1155 is now in keyboard mode. Enter the correct instruction and the rest of the program.

```
WRITE, X, 2, ÷, PRINT
```

List the program.

```
RESET, LIST, +         ++268
START                 SPACE
                       STOP
                      PRINT
                       STOP
                      PRINT
                          X
                          2
                          ÷
                      PRINT
```

The program and checksum are correct.

If a "wrong key" mistake is not discovered until after the entire program is stored, it can still be corrected. Store the program again, including the mistake, as shown below.

```
RESET, CLEAR PROG, WRITE

SPACE, STOP, PRINT, STOP, PRINT, ÷, 2, ÷, PRINT
                                 ↑
                                /
                        mistake/
```

Find the top and list the program.

```
RESET, LIST, +         ++270 ←——— wrong checksum
START                 SPACE
                       STOP
                      PRINT
                       STOP
                      PRINT
Here is the mistake - - - - ÷
                          2
                          ÷
                      PRINT
```

Correct the mistake, as follows.

LIST the program, one step at a time, until the mistake appears.

KEY (S)	TAPE	COMMENTS
RESET, LIST, +	++270	WRONG CHECKSUM
ONE STEP	SPACE	O.K.
ONE STEP	STOP	O.K.
ONE STEP	PRINT	O.K.
ONE STEP	STOP	O.K.
ONE STEP	PRINT	O.K.
ONE STEP	÷	MISTAKE

DELETE the mistake.

```
RESET, DELETE, START
```

INSERT the correct instruction.

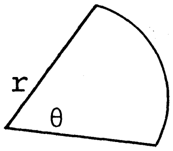
```
INSERT, X
```

List the program.

```
RESET, LIST, +          ++268 ←— new checksum
START                   SPACE
                         STOP
                        PRINT
                         STOP
                        PRINT
                            X ←——— O.K.
                            2
                            ÷
                        PRINT
```

The program is now correct and ready to go.

PROBLEM: Compute the length of a circular arc.



$$L = \frac{\pi r \theta}{180} \quad (\theta \text{ in degrees})$$

PROGRAM:

```
  ++393
SPACE
 STOP         (for value of r)
PRINT
 STOP         (for value of θ)
PRINT
    X
   PI
    X
    1
    8   >180
    0
    ÷
PRINT         (the answer, L)
```

The program was entered as follows.

```
RESET, CLEAR PROG, WRITE

SPACE, STOP, PRINT, STOP, PRINT, X, PI, X, 8, 1, 0, ÷, PRINT
```

Two digits transposed.

The mistake was not discovered until the program was listed.

```
  ++393     Checksum is correct. A checksum is
SPACE       not changed by transposed steps.
 STOP
PRINT
 STOP
PRINT
    X
   PI
    X
    8]
    1]------ Mistake. Two digits transposed.
    0
    ÷
PRINT
```

Correct the mistake, as follows:

KEY(S)	TAPE	COMMENT
RESET, LIST, +	++393	
ONE STEP	SPACE	O.K.
ONE STEP	STOP	O.K.
ONE STEP	PRINT	O.K.
ONE STEP	STOP	O.K.
ONE STEP	PRINT	O.K.
ONE STEP	X	O.K.
ONE STEP	PI	O.K.
ONE STEP	X	O.K.
ONE STEP	8	mistake
ONE STEP	1	mistake

```
RESET, DELETE, START   (deletes the '1')
DELETE, START          (deletes the '8')

RESET, INSERT, 1, 8    (inserts 1 and 8)
```

List the corrected program.

```
RESET, LIST, +           ++393 --------checksum is same as
START                  SPACE           before.
                        STOP
                       PRINT
                        STOP
                       PRINT
                           X
                          PI
                           X
                           1
                           8         Correct.
                           0
                           ÷
                       PRINT
```

The program is now correct.

Note that, in deleting instructions, the 1155 deletes one step each time DELETE and START are pressed.

SUMMARY

- TO STORE A PROGRAM

  (1) Press RESET, CLEAR PROG, WRITE
  (2) Key in the program.

- TO LIST A PROGRAM

  (1) Press RESET, LIST, +
  (2) Press START

- TO LIST A PROGRAM ONE STEP AT A TIME

  (1) Press RESET, LIST, +
  (2) Press ONE STEP once for each step listed.

- TO RUN A PROGRAM

  (1) Press RESET, LIST, +
  (2) Press RESET, START

- TO INSERT INSTRUCTIONS IN A STORED PROGRAM

  (1) List the program, one step at a time, until you reach the place where new instructions are to be stored.
  (2) Press RESET, INSERT.
  (3) Insert as many instructions as you wish.
  (4) Press RESET to terminate INSERT MODE.

- TO DELETE INSTRUCTIONS FROM A STORED PROGRAM

  (1) List the program, one step at a time, until the last step to be deleted appears.
  (2) Press RESET.
  (3) For each step to be deleted, press DELETE, START.

- TO CHANGE INCORRECT INSTRUCTIONS IN A STORED PROGRAM

  (1) List the program, one step at a time, until the last incorrect instruction appears.
  (2) Delete the incorrect instructions.
  (3) Insert the correct instructions.
  (4) When finished, press RESET.

PROBLEM: Convert a temperature measurement from degrees Centigrade to degrees Fahrenheit.

$$F = \frac{9}{5}C + 32$$

The following program illustrates the use of two new instructions, PROGRAM CODE and GO TO.

```
                                   ++576
                      10
                                   SPACE
program code ↗                      STOP      Key in C
                                   PRINT
                                       9      Compute F.
                                       X
                                       5
                                       ÷
                                       3
                                       2
                                       +
                                   PRINT      Print F.
                                ↗GO TO 10     Go to beginning
GO TO instruction
```

A program code is a two digit number that marks a particular place in a program. There are 100 possible program codes, 00 through 99. In the above program, the program code '10' was a purely arbitrary choice.

The GO TO 10 instruction instructs the 1155 to find program code 10 and then execute instructions, beginning with the instruction following the program code.

- To enter a program code, use the [PROG CODE ( ) ( )] key

  For example, the program code '10' is entered in the following manner.

  [PROG CODE ( ) ( )] [1] [0]

  Or, in abbreviated form: PROG CODE 10

- To enter a `GO TO` use the [GOTO ( ) ( )] key.

  For example, enter `GO TO 10` in the following manner.

  [GOTO ( ) ( )] [1] [0]

  In abbreviated form: `GO TO 10`

- Store the program on the preceding page.

  ```
  RESET, CLEAR PROG, WRITE
  ```

  Store the program code: `PROG CODE 10`

  Then store the instructions.

  ```
  SPACE, STOP, PRINT, 9, X, 5, ÷, 32, +, PRINT,GO TO 10
  ```

- List the program. It should appear exactly as shown on page 12-1.

- Run the program for C = 100° and C = 37°.

  ```
  RESET, GO TO 10, START

  Key in C, press START    100.00000
                           212.00000 <---------- F

  Key in C, press START     37.00000
                            98.60000 <---------- F
  ```

The program described on page 12-1 and above converts from Centigrade to Fahrenheit. The following program does the inverse operation, converting from Fahrenheit to Centigrade.

$$C = \frac{5}{9}(F - 32)$$

```
                                ++578
                      23
                             SPACE
program code                  STOP        Key in F.
                             PRINT
                                 3        Compute C.
                                 2
                                 -
                                 5
                                 X
                                 9
                                 ÷
                             PRINT        Print C.
                          GO TO 23        Go to beginning of this
                                          program.

GO TO instruction
```

It is assumed that the previous program (pages 12-1 and 12-2) is still stored in the 1155. If not, store it.

- Append the above program to the one already stored.

DO NOT press CLEAR PROG

Instead, press `RESET`, `WRITE`.

The 1155 will find the last step of the program that is already stored. Key in the new program.

Both programs are now stored.

- List all programs in the 1155's memory.

```
RESET, LIST, +        ++153
START          10
                    SPACE
                     STOP
                    PRINT
                        9
                        X
                        5
                        ÷
                        3
                        2
                        +
                    PRINT
                 GO TO 10
               23
                    SPACE
                     STOP
                    PRINT
                        3
                        2
                        -
                        5
                        X
                        9
                        ÷
                    PRINT
                 GO TO 23
```

$F = \frac{9}{5}C + 32$ program begins at PROG CODE 10.

$C = \frac{5}{9}(F - 32)$ program begins at PROG CODE 23.

- Press ONE STEP

```
        N477
   PROG CODE
```

unused program space.

first keystroke

If ONE STEP is pressed immediately following a list, the 1155 prints the number of unused program storage positions. Following the unused program space, the first keystroke in the program is printed. The 1155 has 511 program storage positions, each equivalent to one keystroke. The above two programs require 34 positions. Therefore, there are 511 - 34 = 477 positions still available. Instructions such as SPACE, PRINT and + require one position. PROG CODE and GO TO instructions require three positions.

- To run the $F = \frac{9}{5}C + 32$ program,

```
RESET, GO TO 10, START

    Key in C, press START      20.00000
                               68.00000←—F

    Key in C, press START     -10.00000
                               14.00000←—F
```

- To run the $C = \frac{5}{9}(F - 32)$ program,

```
RESET, GO TO 23, START

    Key in F, press START      32.00000
                                 .00000←—C

    Key in F, press START     212.00000
                              100.00000←—C
```

The programs can be run on any desired schedule. For example:

```
RESET

GO TO 10, key in C, START      50.00000
                              122.00000←—F

GO TO 23, key in F, START      17.00000
                               -8.33333←—C

          key in F, START      32.00000
                                 .00000←—C

GO TO 10, key in C, START    -273.00000
                             -459.00000←—F

          key in C, START    1000.00000
                             1832.00000←—F
```

13 MULTIPLE PROGRAMS

The number of programs that can be stored in the 1155 at one time is limited by two things.

- The number of possible program codes. There are 100 possible program codes, 00 through 99.
- The total number of key steps for all programs must not exceed 511. This limitation is likely to occur long before the program code limitation.

To store several different programs simultaneously, simply begin each program at a different program code. The program codes do not have to be in numerical sequence since they are merely arbitrary labels that mark a place in the program.

Keying in a long program or several short programs is slow and tedious work. However, if a Model 511 magnetic card reader is available, programs can be recorded on magnetic cards and re-entered in a few seconds.

For information on the Model 511 magnetic card reader, see Appendix D.

14 MEMORY REGISTERS

Inside the 1155 there are 20 memory registers. Each memory register is identified by a two digit number, 00 through 19.

00		05		10		15	
01	9	06		11		16	
02	4	07		12		17	
03		08	-5	13	3.7	18	
04		09		14		19	

20 MEMORY REGISTERS NUMBERED 00 THROUGH 19

Each memory register can store one number. In the registers above:

9 is in register 01
4 is in register 02
-5 is in register 08
3.7 is in register 13

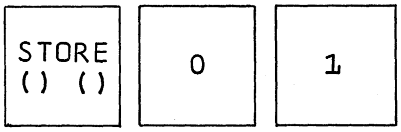
The following keys are used to manipulate numbers in memory registers.

KEY(S)	ABBREVIATION
STORE ( ) ( )	STORE
RECALL ( ) ( )	RECALL
ACCUM STORE ( ) ( )	ACCSTORE
CLEAR ALL REGS	CLR REGS

The following examples illustrate how numbers are keyed in and then stored in memory registers.

- Press RESET
- Put 9 into register 01

  9, STORE 01 STORE 01 means



- Put 4 into register 02

  4, STORE 02

- Put -5 into register 08

  5, OPP SIGN, STORE 08 OPP SIGN means

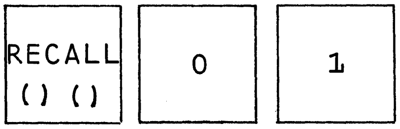
- Put 3.7 into register 13

  3, ·, 7, STORE 13

Recall each number from the register in which it is stored. During manual operation, the RECALL operation also causes the number to be printed.

KEY(S)	TAPE
RECALL 01	9.00000
RECALL 02	4.00000
RECALL 08	-5.00000
RECALL 13	3.70000

RECALL 01 means



The numbers are still stored in registers 01, 02, 08, 13. The RECALL operation does not erase them.

The following table illustrates the activity of the 1155 as numbers are stored and recalled.

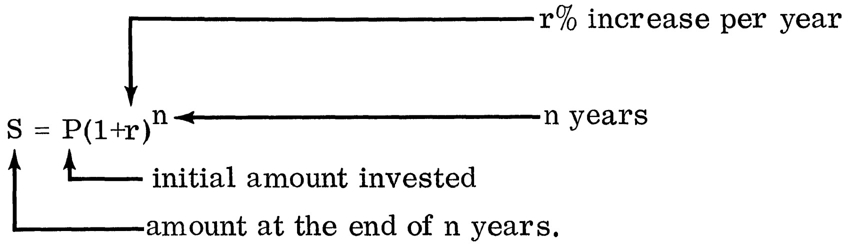
	REGISTERS					
	WORKING		MEMORY			
KEY(S)	L	U	01	02	08	13
RESET	0	0				
9	9	0				
STORE 01	9	0	9			
4	4	9	9			
STORE 02	4	9	9	4		
5	5	4	9	4		
OPP SIGN	-5	4	9	4		
STORE 08	-5	4	9	4	-5	
3	3	-5	9	4	-5	
.	3.	-5	9	4	-5	
7	3.7	-5	9	4	-5	
STORE 13	3.7	-5	9	4	-5	3.7
RECALL 01	9	3.7	9	4	-5	3.7
RECALL 02	4	9	9	4	-5	3.7
RECALL 08	-5	4	9	4	-5	3.7
RECALL 13	3.7	-5	9	4	-5	3.7

INSTRUCTION	DEFINITION
STORE m	Stores the content of the lower register into memory register m. (m is a two digit number between 00 and 19.) This operation does not change the content of the lower register.
RECALL m	Moves the content of the lower register into the upper register, then recalls the content of memory register m into the lower register.

When a number is stored in a memory register, the previous content of that register is erased.

When a number is recalled from a memory register, the content of the memory register is not erased or changed in any way. It is simply copied into the lower register.

P dollars are placed in a savings account which pays r% per year compound interest, compounded annually. In n years, the investment will be worth S dollars, where



Store the following program.

```
       ++125
10
     SPACE
      STOP          Key in P
     PRINT
     STORE 01       Store P in register 01
      STOP          Key in r
     PRINT
         1          Compute 1+r
         +
     STORE 02       Store 1+r in register 02
20
   RECALL 02        Recall 1+r
     SPACE
      STOP          Key in n
     PRINT
    A TO X          Compute (1+r)^n
   RECALL 01        Recall P
         X          Compute S
     PRINT          Print S
     GO TO 20       Go back for a new value of n.
```

Run the program for

P = \$100
r = 6%
n = 2, 5 and 10 years

- RESET, SET DEC 2
- GO TO 10, START

```
Key in P, START      100.00
Key in r, START         .06*

Key in n, START        2.00
                     112.36 - - - - - S

Key in n, START        5.00
                     133.82 - - - - - S

Key in n, START       10.00
                     179.08 - - - - - S
```

If interest in compounded monthly instead of yearly, the monthly interest rate is 6/12 = .5% and n will now refer to the number of months. For example, for 2 years, n = 24 months.

- RESET, SET DEC 3
- GO TO 10, START

```
Key in P, START      100.000
Key in r, START         .005**

Key in n, START       24.000
                     112.716 - - - - S

Key in n, START       60.000
                     134.885 - - - - S

Key in n, START      120.000
                     181.940 - - - - S
```

The previous program illustrates the use of the STORE and RECALL instructions. The following examples illustrate the use of the ACCSTORE (accumulate store) and CLR REGS (clear registers) instructions.

---

*6% must be entered as .06.

**.5% must be entered as .005.

INSTRUCTION	DEFINITION
CLR REGS	Clears all registers. In other words, stores zeros in the lower register, the upper register and memory registers 00-19.
ACCSTORE m	Adds the content of L to the content of memory register m and puts the result in memory register m. This operation does not change L.

For example:

	REGISTERS						
	WORKING		MEMORY				
KEY(S)	L	U	00	01	02	...	19
RESET	0	0			NO CHANGE		
CLR REGS	0	0	0	0	0	...	0
8	8	0	0	0	0	...	0
ACCSTORE 01	8	0	0	8	0	...	0
3	3	8	0	8	0	...	0
ACCSTORE 01	3	8	0	11	0	...	0
7	7	3	0	11	0	...	0
ACCSTORE 01	7	3	0	18	0	...	0
5	5	7	0	18	0	...	0
ACCSTORE 01	5	7	0	23	0	...	0
RECALL 01	23	7	0	23	0	...	0

Set the decimal point to 5 places. Run the above program manually. When finished the tape should look like this:

```
      .00000
     8.00000
 ACCSTORE 01
     3.00000
 ACCSTORE 01
     7.00000
 ACCSTORE 01
     5.00000
 ACCSTORE 01
    23.00000
```

The mean of a list of numbers is the sum of the numbers divided by the number of numbers.

$$\overline{x} = \frac{x_1 + x_2 + \cdots x_n}{n} = \frac{s}{n}$$

where

s = the sum of the numbers $x_1, x_2, \ldots, x_n$

n = the number of numbers.

The following program directs the 1155 to compute the mean of a list of numbers. As the numbers are entered, the 1155 accumulates the sum s in register 01 and also counts the numbers. The count is accumulated in register 02.

```
          ++200
 10
     CLR REGS            Clear all registers (sets s = 0 and n = 0)
        SPACE
 13
         STOP            Key in a number x
        PRINT
     ACCSTORE 01         Add x to the sum s
            1
     ACCSTORE 02         Increase the count by 1
        GO TO 13         Go around for another number
 17
        SPACE
       RECALL 01         Recall s
        PRINT            Print s
       RECALL 02         Recall n
        PRINT            Print n
            ÷            Compute x̄
        PRINT            Print x
         STOP            Stop.  To run program again, simply
        GO TO 10         press START
```

Store the program, list it, proofread it, then run it as described on the next page.

16 TO RUN THE PROGRAM

- RESET, GO TO 10, START
- For each number in the list (x_1, x_2, ..., x_n): Key in x, START
- After all x's have been entered, GO TO 17, START

The 1155 will print s, n and $\bar{x}$.

Below is a run using the list of numbers: 74, 87, 81, 93, 78.

- RESET, GO TO 10, START

```
Key in x, START          74.00000
Key in x, START          87.00000
Key in x, START          81.00000
Key in x, START          93.00000
Key in x  START          78.00000
```

```
• GO TO 17, START       413.00000←——s
                          5.00000←——n
                         82.60000←——x̄
```

Run the program for the following list

5, 8, 7, 9, 3, 7, 6

The tape should look like this

```
 5.00000
 8.00000
 7.00000
 9.00000
 3.00000
 7.00000
 6.00000

45.00000←——s
 7.00000←——n
 6.42857←——x̄
```

17 COUNTING

The following program instructs the 1155 to count. The 1155 will generate and print consecutive whole numbers beginning with 1. The printout will continue until the 1155 is stopped manually.

> To stop the 1155, press STOP

The program is shown below.

```
10      ++457
            1
20
        PRINT
            1
            +
        GO TO 20
```

Store the program, set the number of decimal places to 0 and run the program. The tape might look like the following.

```
RESET, GO TO 10, START    1
                          2
                          3
                          4
                          5
                          6
                          7
                          8
                          9
                         10
                         11
                         12
STOP  was pressed ──────→
```

Pressing START will cause the 1155 to continue from the point at which it was stopped.

Pressing GO TO 10, START will cause the 1155 to start again from the beginning.

Change the program so that it is possible to key in the number at which the count starts.

```
          ++457
    10
        SPACE
         STOP ←—————————— Key in starting number
    20
        PRINT
            1
            +
        GO TO 20
```

Run the program twice. The first time start the count at 12 and the second time at 37.

- RESET, SET DEC 0
- GO TO 10, START

```
        Key in 12, START      12
                              13
                              14
                              15 ←—STOP was pressed.
```

- RESET, GO TO 10, START

```
        Key in 37, START      37
                              38
                              39
                              40 ←—STOP was pressed.
```

Another counting program. This time the count is kept in memory register 07.

```
          ++698
  10
        SPACE
         STOP                     Key in starting number
        STORE 07                  Store it in register 07.
  20
      RECALL 07                   Recall the count
        PRINT                     Print it
            1
   ACCSTORE 07                    Increase the count (in 07) by one
        GO TO 20                  Go around again
```

Try it. The run procedure is the same as for the program at the top of this page.

In the previous counting programs, the counting increment is one. That is, the number printed each time is one more than the preceding number printed. Modify* the program so that the increment can be keyed in and stored in register 12.

```
              ++958
  10
           SPACE
            STOP                  Key in starting number
           PRINT                  Print it
           STORE 07               Store it in 07
            STOP                  Key in the increment
           PRINT                  Print it
           STORE 12               Store it in 12
           SPACE
  20
          RECALL 07               Recall the count
           PRINT                  Print it
          RECALL 12               Recall the increment
        ACCSTORE 07               Increase the count by the increment
           GO TO 20               Go around again
```

Start at 1 and increment by 2.

- RESET, GO TO 10, START

```
        Key in 1,START          1
        Key in 2,START          2

                                1
                                3
                                5
                                7<------ STOP was pressed.
```

---

*Remember: An already stored program can be modified quickly by using the editing methods described on pages 10-1/10-5.

Start at 10 and decrement by 5.

- RESET, GO TO 10, START

```
Key in 10, START      10
Key in -5, START      -5

                      10
                       5
                       0
                      -5
                     -10
                     -15 <------ STOP  was pressed.
```

SET DEC 2 then start at 1 and increment by .25.

- RESET, GO TO 10, START

```
Key in 1, START        1.00
Key in .25, START       .25

                       1.00
                       1.25
                       1.50
                       1.75
                       2.00
                       2.25 <------ STOP was pressed.
```

18 TABLE MAKER

Here are two short square root tables.

Table 1

x	$\sqrt{x}$
1	1.00000
2	1.14121
3	1.73205
4	2.00000
5	2.23607
6	2.44949
7	2.64575
8	2.82843

Table 2

x	$\sqrt{x}$
2.00	1.41421
2.01	1.41774
2.02	1.42127
2.03	1.42478
2.04	1.42829
2.05	1.43178
2.06	1.43527
2.07	1.43875

In Table 1, x begins at 1 and increases in increments of 1. In Table 2, x begins at 2 and increases in increments of .01. The following program can be used to compute tables such as the above. This program is a simple modification of the program on page 17-3.

```
          ++044
10
        SPACE
         STOP                Key in starting value of x
        PRINT
        STORE 07             Store x in register 07
         STOP                Key in the increment
        PRINT                Store it in register 12
        STORE 12
20
        SPACE
       RECALL 07             x
        PRINT
         SQRT                √x
        PRINT
       RECALL 12
     ACCSTORE 07             Increase x by increment
        GO TO 20             Go around again.
```

Compute Table 1.

```
•RESET, SET DEC 5
•GO TO 10, START

        Key in 1, START          1.00000
        Key in 1, START          1.00000

                                 1.00000
                                 1.00000

                                 2.00000
                                 1.41421

                                 3.00000
                                 1.73205← STOP was pressed.
```

When you want the 1155 to stop, press STOP.

Compute Table 2

```
•RESET, GO TO 10, START

        Key in 2, START          2.00000
        Key in .01, START         .01000

                                 2.00000
                                 1.41421

                                 2.01000
                                 1.41774

                                 2.02000
                                 1.42127← STOP was pressed.
```

Write a program to compute a table of

(1) x and x^2

(2) x and 1/x

(3) x and x^3

The programs for (1) and (2) above can be obtained by changing one instruction in the program on page 18-1.

19 COUNTDOWN

The following program directs the 1155 to count down from 10 to 0 then stop automatically.

In the program below, `IF-GO TO 99` means IF NEG ()() 9 9

```
       ++921
10                             Start the count at 10 in register 05
           1
           0
       STORE 05
20
      RECALL 05                Recall the count
    IF-GO TO 99                If it is negative, go to 99
       PRINT                   Otherwise, print it and
                               Decrease it by 1.*
           1
    OPP SIGN
    ACCSTORE 05                Then go around again
       GO TO 20
99                             End of program.  Count has become
        STOP                   negative.
```

Run it.

- RESET, SET DEC 0
- GO TO 10, START

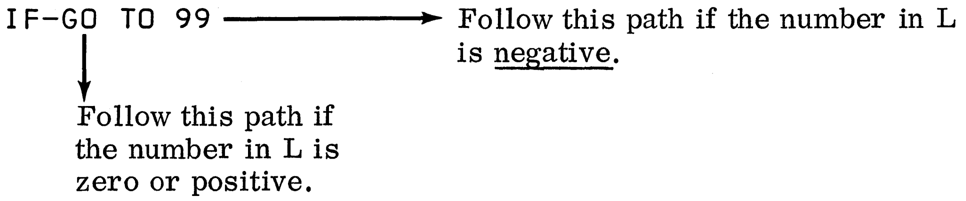
The 1155 counts down from 10 to 0 then stops automatically.

```
10
 9
 8
 7
 6
 5
 4
 3
 2
 1
 0
```

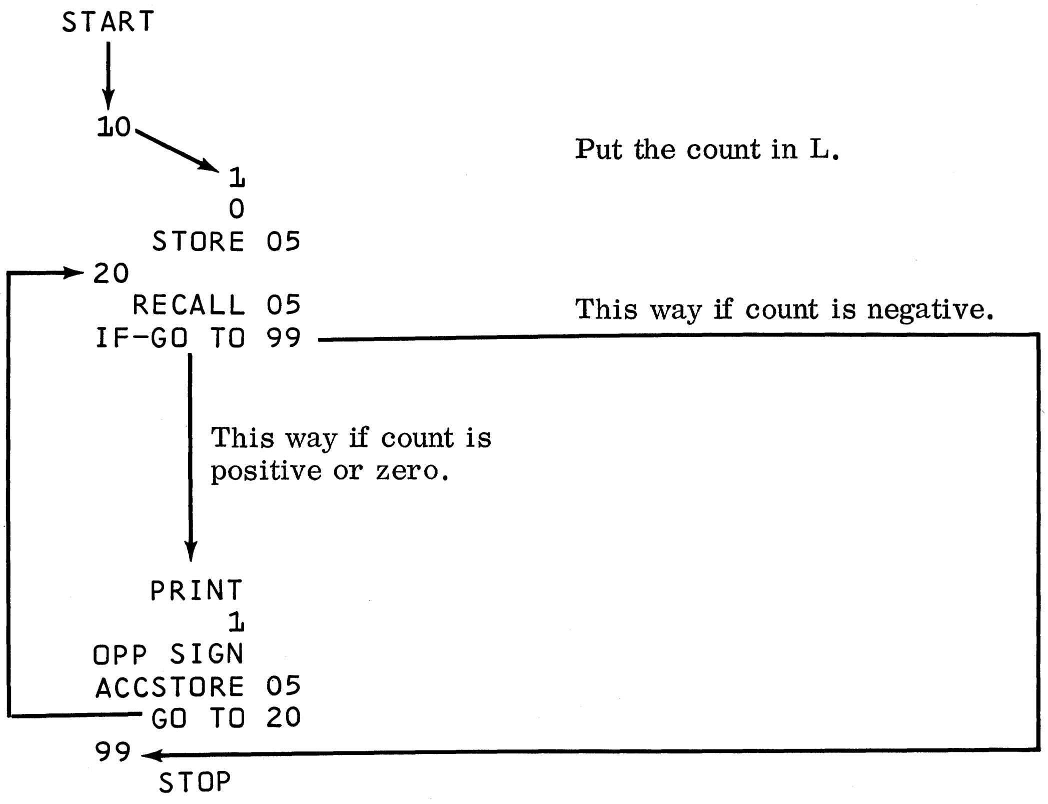
---

*The count is decreased by 1 by adding -1 to the previous count.

The program works. The count starts at 10, is decreased by one each time and stops at zero. Why? The key is the IF-GO TO 99 instruction



In the program, it works like this. Follow the arrows.



```
START
 10
      1
      0
   STORE 05
 20
   RECALL 05
 IF-GO TO 99
   PRINT
       1
 OPP SIGN
 ACCSTORE 05
   GO TO 20
 99
   STOP
```

The `IF-GO TO` instruction can be used to build a table maker program that stops automatically when x reaches a predetermined value. The following program instructs the 1155 to print a table x and $\sqrt{x}$ for x ranging from a to b in increments of h.

- x is in register 01 (start with x = a)
- b is in register 02
- h is in register 03

```
          ++6711
 10
        SPACE
         STOP                  Key in a.
        PRINT
        STORE 01               x = a  register 01
         STOP                  Key in b.
        PRINT
        STORE 02
         STOP                  Key in h.
        PRINT
        STORE 03
 20
       RECALL 02               Compute b - x.
       RECALL 01
            -
     IF-GO TO 99               If b - x < 0, table is complete.
                               Go to 99.

       RECALL 01               Otherwise, recall x and print it.
        SPACE
        PRINT
         SQRT                  Compute √x and print it.
        PRINT                  Increase x by h and go around.
       RECALL 03
     ACCSTORE 01
        GO TO 20
 99
         STOP                  End of program.
```

Does it work? Lets find out.

Data: a = 1, b = 8, h = 1

```
●RESET, SET DEC 5
●GO TO 10, START

        Key in a, START          1.00000
        Key in b, START          8.00000
        Key in h, START          1.00000

                                 1.00000
                                 1.00000

                                 2.00000
                                 1.41421

                                 3.00000
                                 1.73205

                                 4.00000
                                 2.00000

                                 5.00000
                                 2.23607

                                 6.00000
                                 2.44949

                                 7.00000
                                 2.64575

                                 8.00000
                                 2.82843
```

For practice, run the program for a = 2, b = 2.07, h = .01

Change the function: Write a program to compute a table of:

(1) x and 2^x.

(2) r and πr^2.

This section describes the IND (00) key and three new instructions.

INSTRUCTION	KEY(S)	INSTRUCTION ABBREVIATION
STORE INDIRECT	STORE ( ) ( ) IND (00)	STORE N
RECALL INDIRECT	RECALL ( ) ( ) IND (00)	RECALL N
ACCSTORE INDIRECT	ACCUM STORE ( ) ( ) IND (00)	ACCSTORE N

Memory register 00 is involved in all INDIRECT operations. The content of register 00 must be a whole number between 1 and 19, inclusive. Otherwise, an ERROR message will occur.

STORE N	Store the content of the lower register in memory register k where k is the content of memory register 00.
RECALL N	Recall the content of memory register k where k is the content of memory register 00.
ACCSTORE N	Add the content of the lower register to the memory register k where k is the content of memory register 00.

Remember: k must be a two digit whole number between 01 and 19, inclusive.

The following program first clears all registers, then permits the operator to key in up to 19 numbers. The first number entered is stored in memory register 01, the second number is stored in 02, and so on.

```
10
    CLR REGS               Clear all registers
           1
       STORE 00            Set k = 1 (k is content of register 00)
       SPACE
11
        STOP               Key in a number.
       PRINT
       STORE N             Store number  in register k.
           1
    ACCSTORE 00            Increase k by 1.
       GO TO 11            Go around for next number.
```

The following program recalls and prints the contents of registers 01 to 19 inclusive. That is, this program "dumps" the memory of the 1155.

```
20
       SPACE
           1
       STORE 00            Set k = 1
21
      RECALL N             Recall and print the content of
       PRINT               register k
           1
    ACCSTORE 00            Increase k by 1
      RECALL 00
           2
           0
           -               Compute k - 20
    IF-GO TO 21            If k - 20 < 0 go around again
        STOP               Otherwise, stop
```

Store both programs, then list and proofread them.

Here is a sample run using both programs.

```
•RESET, SET DEC 0
•GO TO 10, START
                                    3
                                    7
                                    5
                                    4
                                    6
                                    9
                                    2
```

Key in seven numbers. The 1155 stores them in registers 01 through 07.

Next, dump the memory.

```
•RESET, GO TO 20, START             3
                                    7
                                    5
                                    4
                                    6
                                    9
                                    2
                                    0
                                    0
                                    0
                                    0
                                    0
                                    0
                                    0
                                    0
                                    0
                                    0
                                    0
                                    0
```

The seven numbers entered previously are in registers 01-07. Registers 08-19 contain zeros.

21 SUMMARIZING SURVEY DATA

The following questionnaire was sent to 20 people.

DOES YOUR COMPUTER UNDERSTAND YOU?	
Circle your answer.	
YES	NO

The responses to the twenty questionnaires are shown below. Each response is YES or NO.

YES	YES	NO	YES	NO	NO	NO	YES	NO	NO
NO	NO	YES	YES	NO	YES	YES	NO	NO	NO

The following table is a summary of the results.

RESPONSE	HOW MANY
YES	8
NO	12
TOTAL	20

Code the responses. Use 1 to represent YES and 2 to represent NO.

RESPONSE	CODE
YES	1
NO	2

The coded responses are shown below.

1	1	2	1	2	2	2	1	2	2
2	2	1	1	2	1	1	2	2	2

Write a program to summarize the data. Count YES responses (Code 1) in register 01 and NO responses (Code 2) in register 02.

```
          ++036
10
   CLR REGS              Clear all registers. (Sets counts to zero)
      SPACE
11
       STOP              Key in a response code (1 for YES,
      PRINT                2 for NO)
      STORE 00           Store response code in register 00
          1
   ACCSTORE N            Increase YES count (REG  01) or NO
      GO TO 11             count (REG  02) by 1.
20
      SPACE
     RECALL 01           Recall YES count (REG  01) and print it.
      PRINT
     RECALL 02           Recall NO count (REG  02) and print it.
      PRINT
          +              Compute total count (YES + NO)
      PRINT
       STOP              End of program
```

```
● RESET, SET DEC 0
● GO TO 10, START
```

Enter data. ⟶

```
1
1
2
1
2
2
2
1
2
2
2
2
1
1
2
1
1
2
2
2
```

```
● GO TO 20, START
```

1155 prints results

```
 8      Yes
12      No
20      Total
```

Write a program to summarize data from a questionnaire with n possible responses.

Code the n possible responses: 1, 2, 3, ..., n.

The program, shown below, has three parts. Part 1 clears all registers, stops for the value of n, and stores this value in register 19. Part 2 does the actual work of counting responses. It also counts the total number of all responses and keeps this count in register 18. Part 3 prints the n response tallies (from registers 1 to n) and the total of all response tallies (from register 18).

```
        ++001
10                       PART 1
   CLR REGS
      SPACE
       STOP              Key in n
      PRINT
      STORE 19           Store n in register 19
11                       PART 2
       STOP              Key in response code
      PRINT
      STORE 00           Store response code in register 00.
          1
   ACCSTORE N            Increase appropriate tally.
   ACCSTORE 18           Increase total tally.
      GO TO 11
20                       PART 3
      SPACE
          1
      STORE 00           Set k = 1.
21
     RECALL N            Recall tally from register k and print it.
      PRINT
          1              Increase k by 1.
   ACCSTORE 00
     RECALL 19
     RECALL 00           Compute n - k.
          -              If n - k < 0, go to 99.
   IF-GO TO 99           Otherwise, go around again.
      GO TO 21
99
     RECALL 18
      PRINT
       STOP              End of program
```

Here is a sample run for n = 4.

```
● RESET, SET DEC 0
● GO TO 10, START                  4←—This is n
                                   3←—data starts here
        Key in n, START            1
        Sample data. Twenty        4
        response code. Each        2
        response code is 1, 2,     3
        3, or 4.                   4
                                   4
                                   1
                                   3
                                   3
                                   3
                                   4
                                   2
                                   3
                                   4
                                   1
                                   3
                                   2
                                   4
  After entering all data,         2

●GO TO 20, START                   3
                                   4
                                   7
                                   6
                                  20
```

The results show that 3 people gave response number 1, 4 people gave response number 2, 7 people gave response number 3 and 6 people gave response number 6. There were 20 responses in all.

In this program, register 19 is used to store n and register 18 is used to store the total number of responses. Therefore, registers 01 through 17 are available to store individual response tallies. In other words, remember, n = 17.

For practice, modify the program. In addition to the raw tallies, also compute and print the percent of the total represented by each tally. For example, in the above sample run, 15% of responses were response number 1.

22 GO TO N

The GO TO N (GO TO, INDIRECT) instruction is programmed by pressing

GO TO ( ) ( ) then IND (00)

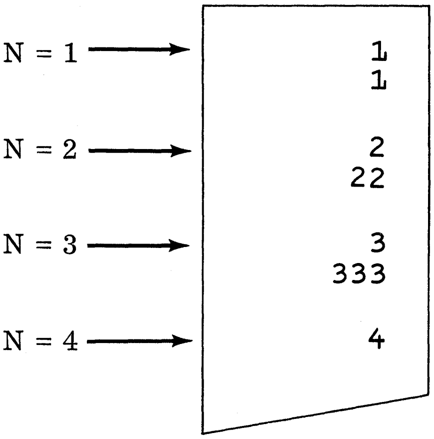
GO TO N Go to the program code specified by the content of register 00. The content of register 00 must be a whole number in the range 00 to 99, inclusive.

For example, if the content of register 00 is 37, then a GO TO N instruction will cause the 1155 to go to program code 37, and continue from there.

The following program illustrates the use of the GO TO N instruction.

```
        ++249
10
       SPACE                  Key in N.  (N = 1, 2, 3, or 10)
        STOP                  Store N in register 00.
       PRINT
       STORE 00
       GO TO N                If N = 1, this part is executed.
01
           1
       PRINT
       GO TO 10
02
           2                  If N = 2, this part is executed.
           2
       PRINT
       GO TO 10
03
           3                  If N = 3, this part is executed
           3
           3
       PRINT
       GO TO 10
```

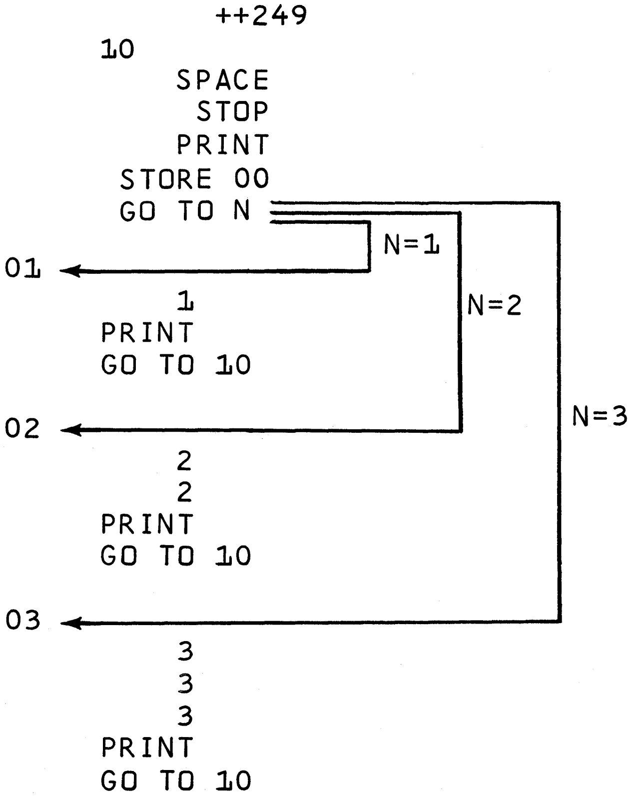
Set the number of decimal places to 0 and run the program for N = 1, 2, 3, and 4. The printout should look like the following.



```
  1
  1

  2
 22

  3
333

  4
```

For N = 4, nothing happened because there is no `PROG CODE 04` in the program. Press `RESET` to continue operating.

The following diagram may clarify what happens as the 1155 runs the program.



```
      ++249
  10
     SPACE
      STOP
     PRINT
    STORE 00
    GO TO N
01
      1
  PRINT
  GO TO 10
02
      2
      2
  PRINT
  GO TO 10
03
      3
      3
      3
  PRINT
  GO TO 10
```

Key in N (N = 1, 2, or 3)

Store N in register 00.

Remember: `GO TO N` instructs the 1155 to go to the <u>program code</u> specified by register 00.

23 SUBROUTINES

Let $f(x) = x^3 + x^2 + x + 1$

The following is a program to compute the value of f(x) for a given value of x.

```
       ++858
10
     SPACE
      STOP
     PRINT
     STORE 19
         3
     A TO X
    RECALL 19
      X SQ
         +
    RECALL 19
         +
         1
         +
     PRINT
     GO TO 10
```

This part of the program does the actual work of computing f(x). At the beginning of this section, x is in L. At the end, f(x) is in L.

PROBLEM 1: Given a and b, compute u, where

$$u = (a^3 + a^2 + a + 1)(b^3 + b^2 + b + 1)$$
$$= f(a)\ f(b)$$

PROBLEM 2: Given a, b and c compute v, where

$$v = (a^3 + a^2 + a + 1)(b^3 + b^2 + b + 1)(c^3 + c^2 + c + 1)$$

$$= f(a)\ f(b)\ f(c)$$

A program to solve PROBLEM 1 is shown on the following page. The program consists of a subroutine to compute f(x) and a main program which calls (uses) the subroutine twice.

```
INSTRUCTIONS                 COMMENTS

            ++126
10                     MAIN PROGRAM:  u = f(a)f(b).
         SPACE
          STOP         Key in a.
         PRINT
         STORE 19      Store a in register 19. ┐
             2         Put N in L.             │  Call the subroutine.
             0                                 │
         GO TO 50      Go to subroutine.       ┘

20                     Subroutine returns here with f(a) in L.
         STORE 01      Store f(a) in register 01.
          STOP         Key in b.
         PRINT
         STORE 19      Call the subroutine.
             3
             2
         GO TO 50

32                     Subroutine returns here with f(b) in L.
        RECALL 01      Compute and print u.
             X
         PRINT
         GO TO 10      Go to beginning of program.

50                     SUBROUTINE:  Compute f(x).
         STORE 00      Store N in register 00.
        RECALL 19      Compute f(x).
             3              │
        A TO X              │
        RECALL 19           │
          X SQ              │
             +              │
        RECALL 19           │
             +              │
             1              │
             +         f(x) is now in L.
         GO TO N       Return to program code N (in register 00).
```

Store the program and use the 1155 to verify the following calculations.

(1) If a = 1 and b = 2, then u = 60.

(2) If a = .3 and b = .5, then u = 2.65688.

To call the f(x) subroutine:

- Store x in register 19.
- Put return code N, in L.
- GO TO 50.

The subroutine returns to program code N with f(x) in L.

In the program on the preceding page, the f(x) subroutine is called twice.

FIRST TIME:	$x = a$, $N = 20$. Subroutine returns to program code 20 with f(a) in L.
SECOND TIME:	$x = b$, $N = 32$. Subroutine returns to program code 32 with f(b) in L.

For practice, write a main program to solve PROBLEM 2 on page 23-1. The main program should call the f(x) subroutine three times, once for $x = a$, once for $x = b$ and once for $x = c$. Store the main program and the f(x) subroutine and use the 1155 to verify the following.

(1) If $a = 1$, $b = 2$ and $c = 3$, then $v = 2400$.

(2) If $a = .7$, $b = .2$ and $c = .5$, then $v = 5.92722$.

For additional practice, write a subroutine to evaluate

$$g(x) = \sqrt{\frac{1}{x^2 + x + 1}}$$

and use it in a main program to compute the value of

$$\frac{g(r - 1)g(r + 1)}{g(r)}$$

for a given value of r.

PROBLEM. Write a subroutine to compute h(x) where

$$h(x) = 7.2x^{3/2}$$

The subroutine is to be called as follows:

- Put x in L.
- Put return code, N, in L. The value of x is pushed into U.
- GO TO 60.

The subroutine returns to program code N with h(x) in L.

The subroutine is shown below. The contents of L, U and register 00 are monitored.

INSTRUCTIONS	L	U	00	COMMENTS
60	N	x	N	Initially, N is in L, x is in
STORE 00	N	x	N	U.
EXCG	x	N	N	New instruction, described
3	3	x	N	below.
A TO X	x^3	0	N	
SQ RT	$x^{3/2}$	0	N	
7	7	$x^{3/2}$	N	
.	7.	$x^{3/2}$	N	
2	7.2	$x^{3/2}$	N	
X	h(x)	0	N	
GO TO N	h(x)	0	N	Return to program code N.

The EXCG (exchange) instruction is programmed by pressing the RECALL key twice.

EXCG means RECALL ( ) ( ) RECALL ( ) ( )

EXCG Exchange the contents of L and U.

The use of the subroutine is illustrated by the program on the following page.

Given a, compute w(a) = h(a) - h(a - 1).

```
 INSTRUCTIONS                      COMMENTS
-----------------------------------------------------------------------
          ++499
10                          MAIN PROGRAM:  w(a) = h(a) - h(a - 1).
        SPACE
         STOP               Key in a.
        PRINT
        STORE 01            Store a in register 01.
            2               Set return code (N) to 20.
            0
        GO TO 60            Go to subroutine.
20                          Subroutine returns here with h(a) in L.
        STORE 02            Store h(a) in register 02.
       RECALL 01            Compute a - 1.
            1
            -
            3               Set return code (N) to 30.
            0
        GO TO 60            Go to subroutine.
30                          Subroutine returns with h(a - 1) in L.
       RECALL 02            Recall h(a).
       EXCG                 Exchange h(a) and h(a - 1).
            -               w(a) = h(a) - h(a - 1) is in L.
        PRINT               Print w(a).
        GO TO 10            Go to beginning of program.
-----------------------------------------------------------------------
```

Store the above main program, then store the subroutine to compute h(x). Use the 1155 to verify the following.

(1) w(2) = 13.16468

(2) w(3) = 17.04762

APPENDIX A

ERROR 1 Overflow in e^x or $|A|^x$ operation. Result greater than 10^{100}.

`RESET` `2` `3` `1` `eˣ`

`RESET` `2` `ENTER` `1` `0` `0` `0` `|A|ˣ`

ERROR 2 Overflow in TAN operation. Absolute value of correct result greater than 10^{100}.

`RESET` `DEG` `9` `0` `TAN`

ERROR 3 Attempt to compute logarithm of zero.

`RESET` `ln|x|` or `RESET` `log|x|`

ERROR 4 Attempt to compute ARC SIN or ARC COS of x for which $|x| > 1$.

`RESET` `2` `ARC` `SIN`

ERROR 5 Division by zero.

`RESET` `1` `ENTER` `0` `÷`

ERROR 6 Attempt to compute $|x|!$ for $|x| \geqq 100$. Correct result greater than 10^{100}.

`RESET` `1` `0` `0` `|x|!`

ERROR 7 Overflow. Correct result greater than 10^{100}.

`RESET` `2` `ENTER EXPONENT` `5` `0` X^2

`RESET` `7` `ENTER EXPONENT` `9` `9` `ENTER` `2` `X`

ERROR 8 Attempt to compute reciprocal of zero.

`RESET` `1/X`

ERROR 9 Illegal key entry during manual operation.

`IF NEG ()()` or `PROG CODE ()()`

ERROR 10 Attempt to compute polar coordinates when rectangular coordinates are both zero (x = 0 and y = 0).

`RESET` `TO POLAR`

ERROR 11 Incomplete address during list. For example: STORE, RECALL, ACCSTORE or GO TO not followed by either a two digit number or N (INDIRECT key).

ERROR 12 Incorrect indirect address in register 00 or incomplete indirect instruction. For example, INDIRECT not preceded by STORE, RECALL, ACCSTORE or GO TO.

Scientific notation is a shorthand way of expressing very large or very small numbers. In scientific notation a number is represented by a mantissa and an exponent.

$$\text{mantissa} \nearrow \underline{3.6} \text{ x } 10^{9} \longleftarrow \text{exponent}$$

Examples of numbers expressed in both ordinary notation and scientific notation are shown below.

- One trillion

  Ordinary notation: 1,000,000,000,000

  Scientific notation: 1 x 10^{12}

  12 zeros

- Mass of the Earth in tons.

  Ordinary notation: 6,588,000,000,000,000,000,000

  Scientific notation: 6.588 x 10^{21} (exponent is positive)

- Mass of a hydrogen atom in kilograms.

  Ordinary notation: .00000000000000000000167

  Scientific notation: 1.67 x 10^{-21} (exponent is negative)

- π, rounded to 5 decimal places

  Ordinary notation: 3.14159

  Scientific notation: 3.14159 x 10^{0} (exponent is zero)

Numbers expressed in scientific notation can be converted to everyday notation, as follows:

Case 1. Exponent is positive or zero.

(1) Write down the mantissa separately.

(2) Move the decimal point of the mantissa to the right the number of places specified by the exponent. If necessary, add zeros.

Example A: 2.05 x 10^8

(1) 2.05

(2) 205000000. (8 places)

Example B: 205 x 10^6

(1) 205

(2) 205000000. (6 places)

Example C: 3.14159 x 10^0

(1) 3.14159

(2) 3.14159 (0 places)

Case 2. Exponent is negative.

(1) Write down the mantissa separately.

(2) Move the decimal point of the mantissa to the left the number of places specified by the exponent. If necessary, add zeros.

Example: 1.23 x 10^{-16}

(1) 1.23

(2) .0000000000000000123 (16 places)

APPENDIX C

The following functions are described on pages 4-1 through 4-4 and page 7-1.

X SQ, SQ RT, RECIP, A TO X

The X SQ, SQ RT and RECIP keys affect only L. The A TO X key affects both L and U.

The rest of the function keys are shown below.

KEY(S)	ABBREVIATION	
e^x	E TO X	These keys affect only L.
ln|x|	LN E	
log|x|	LOG 10	
SIN	SIN	
COS	COS	
TAN	TAN	
|X|!	FACTRL	
INTGR	INTEGER	
TO POLAR	TO POLAR	These keys affect both L and U .
TO RECT	TO RECT	

The DEG, RAD and ARC keys are used in connection with the SIN, COS, TAN, TO POLAR and TO RECT keys.

Examples of the use of the function keys are shown below.

- Compute e^x

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
Key in x	x	0	Repeat these two instructions for each value of x.
E TO X	e^x	0	

Verify: (1) e^1 = 2.71828

(2) $e^{2.3}$ = 9.97418

(3) e^{231} = ERROR 1

- Compute ln|x|

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
Key in x	x	0	Repeat for each value of x.
LN E	ln|x|	0	

Verify: (1) ln 2 = .69315

(2) ln 10 = 2.30259

(3) ln|-2| = .69315

(4) ln 0 = ERROR 3

- Compute log |X|

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
Key in x	x	0	Repeat for each value of x.
LOG 10	log|x|	0	

Verify: (1) log 2 = .30103

(2) log 10 = 1.00000

(3) log|-2| = .30103

(4) log 0 = ERROR 3

The SIN, COS and TAN keys may be used to compute the sine, cosine or tangent of an angle expressed in either degrees or radians.

- To compute sin x where x is expressed in degrees:

INSTRUCTION	L	U	COMMENTS
RESET	0	0	Do these once.
DEG	0	0	
Key in x	x	0	Repeat for each value of x.
SIN	sin x	0	

Verify:
(1) sin 30° = .50000
(2) sin 60° = .86603
(3) sin 225° = -.70711
(4) sin 5000° = -.64279

- To compute cos x where x is expressed in radians:

INSTRUCTION	L	U	COMMENTS
RESET	0	0	Do these once.
RAD	0	0	
Key in x	x	0	Repeat for each value of x.
COS	cos x	0	

Verify:
(1) cos 1 = .54030
(2) cos π = -1.00000
(3) cos 10 = -.83907
(4) cos (π/4) = .70711*

* $\boxed{\pi}$ $\boxed{4}$ $\boxed{\div}$ $\boxed{\text{COS}}$

- Verify the following using the TAN key.

ANGLE IN DEGREES

(1) tan 30° = .57735
(2) tan 293° = -2.35585
(3) tan 5000° = -.83910
(4) tan 90° = ERROR 2

ANGLE IN RADIANS

(1) tan 1 = 1.55741

(2) tan π = .00000

The ARC key is used as a prefix to the SIN, COS and TAN keys to compute the arc sine, arc cosine and arc tangent for a number.

- To compute arc sin x, result in radians:

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
RAD	0	0	
Key in x	x	0	Repeat for each value of x.
ARC	x	0	
SIN	arcsin x	0	

Use the 1155 to verify the following results.

(1) arcsin .5 = .52360 radians

(2) arcsin 2 = ERROR 4

- To compute arc tan x, result in degrees:

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
DEG	0	0	
Key in x	x	0	Repeat for each value of x.
ARC	x	0	
TAN	arctan x	0	

Verify: (1) arctan .5 = 26.56505°

(2) arctan 1 = 45.00000°

(3) arctan 2 = 63.43495°

(4) arctan 1000 = 89.94270°

(5) arctan (-1) = -45.00000°

(6) arctan (-1000) = -89.94270°

The $\boxed{|X|\,!}$ key instructs the 1155 to first compute the absolute value of the number in L, then round it to the nearest integer then compute the factorial function.

INSTRUCTION	L	U
RESET	0	0
Key in x	x	0
FACTRL	$\lvert x\rvert !$	0

Verify: (1) 5! = 120

(2) $\lvert -4.9\rvert$! = 120

(3) 4.3! = 24

(4) 52! = 8.06581751709 E +67

(5) 69! = 1.71122452428 E +98

(6) 70! = ERROR 7

The INTEGER key instructs the 1155 to compute the <u>integer part</u> of the number in L.

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
Key in x	x	0	
INTEGER	int(x)	0	int(x) = integer part of x.

Verify: (1) int(4.9) = 4

(2) int(π) = 3

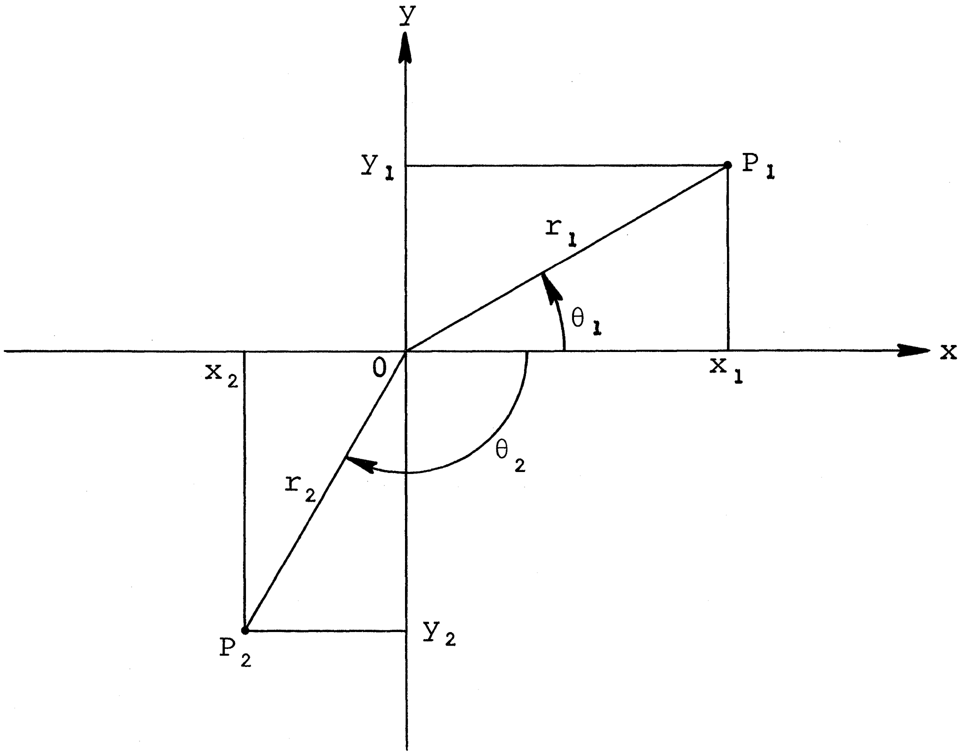
(3) int(.999) = 0

(4) int(-2.5) = -2

(5) int(10) = 10

(6) int(-7) = -7

The `TO POLAR` key is used to convert from rectangular to polar coordinates and the `TO RECT` key is used to convert from polar to rectangular coordinates. The following diagram shows the relationship between rectangular coordinates (x, y) and polar coordinates (r, θ).



Conversion from rectangular to polar coordinates:

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

$$\theta = \tan^{-1} \frac{y}{x} \qquad -180 < \theta \leq 180 \quad \text{(degrees)}$$

$$-\pi < \theta \leq \pi \quad \text{(radians)}$$

Conversion from polar to rectangular coordinates:

$$x = r \cos\theta$$

$$y = r \sin\theta$$

Convert from rectangular to polar coordinates with θ computed in degrees.*

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
DEG*	0	0	
Key in x	x	0	Repeat for each set of (x, y) coordinates.
ENTER	x	0	
Key in y	y	x	
TO POLAR	θ	r	

Use the 1155 to verify the following:

x	y	r	θ
1	1	1.41421	45.00000
1	-1	1.41421	-45.00000
-1	1	1.41421	135.00000
-1	-1	1.41421	-135.00000
-1	0	1.00000	180.00000
0	1	1.00000	90.00000
2	3	3.60555	56.30993

If x and y are both zero, the 1155 prints: ERROR 10

---

*To obtain θ in radians, press RAD instead of DEG.

Convert from polar to rectangular coordinates. θ is expressed in degrees.*

INSTRUCTION	L	U	COMMENTS
RESET	0	0	
DEG*	0	0	
Key in r	r	0	Repeat for each set of coordinates (r, θ).
ENTER	r	0	
Key in θ	θ	r	
TO RECT	y	x	

Use the 1155 to verify the following: (θ in degrees)

r	θ	x	y
1	0	1.00000	.00000
1	30	.86603	.50000
1	90	.00000	1.00000
1	150	-.86603	.50000
1	180	-1.00000	.00000
1	210	-.86603	-.50000
1	330	.86603	-.50000
1	-30	.86603	-.50000
2	53	1.20363	1.59727
3	-123	-1.63392	-2.51601

---

*If θ is given in radians, press RAD instead of DEG.

FRIDEN MODEL 1155 ADVANCED PROGRAMMABLE CALCULATOR
OPERATOR'S PRIMER
Publication No. 55-537

We produce manuals for you, and we want you to find them useful and informative. That's our job.

So we're asking you to help us furnish you with the best possible publications. Please take a few minutes to answer the following questions. Add any comments you wish. If you desire a reply to any question, be sure to include your name and address.

Thank you.

DETACH HERE

- Does this manual meet your needs? Yes ☐ No ☐
  If not, what additional information would be of help to you?

- Can you find what you're looking for quickly and easily? Yes ☐ No ☐
  How can the organization be improved?

- Is the material easy to read and to understand? Yes ☐ No ☐
  Are there enough illustrations to support the text? Yes ☐ No ☐
  Comments

- Did you find any errors or ambiguities in the manual? Yes ☐ No ☐
  If yes, please cite page, line, and/or figure number with your comments.

- Other comments.

- What is your relationship to the product described?
  - ☐ Operator.
  - ☐ Programmer.
  - ☐ Other (please specify)

SINGER
BUSINESS MACHINES

FOLD BACK

FIRST CLASS
PERMIT No. 320
San Leandro, Calif.

BUSINESS REPLY MAIL
No postage stamp necessary if mailed in the United States

POSTAGE WILL BE PAID BY

SINGER BUSINESS MACHINES
2350 Washington Ave.
San Leandro, California 94577

Attn: Customer Technical Publications,
Department 753

FOLD BACK