REFERENCE MANUAL

FRIDEN MODEL 1155 ADVANCED PROGRAMMABLE CALCULATOR

SINGER
BUSINESS MACHINES

WILLIAM R. PALLESCHI

REFERENCE MANUAL

FRIDEN MODEL 1155 ADVANCED PROGRAMMABLE CALCULATOR

Publication No. 55-538
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 1-4	9/72
2-1 thru 2-36	9/72
3-1 thru 3-30	9/72
4-1 thru 4-8	9/72
A-1 thru A-2	9/72
B-1 thru B-3	9/72
I-1 thru I-3	9/72

CONTENTS

1 INTRODUCTION

The Friden* 1155 is a programmable calculator with the following features:

- It may be used as an ordinary desktop calculator; many operations may be performed directly from the keyboard with no programming required.
- 13 digit accuracy in the range 10^{-99} to 10^{99}.
- Printing in either fixed point or scientific notation.
- Programming either from the keyboard or with the Model 511 magnetic card reader, of up to 511 program steps.
- Easy insertion and deletion of program steps.
- Program listing.
- Two working registers and twenty memory registers.
- Up to 100 program codes (labels).
- Indirect addressing of registers and program codes.
- Full complement of arithmetic and mathematical functions.

Friden 1155 Programmable Calculator

*A trademark of The Singer Company

MODES

The 1155 may be in any of seven states or modes. The central mode, called the keyboard mode, may be reached at any time from any of the other modes by pressing RESET.

The following is a brief description of each mode. Greater detail is available in the associated sections of this manual.

KEYBOARD MODE

Allows the user to directly operate all arithmetic and mathematical functions and to input numbers and print answers. All other modes are reached via the keyboard mode.

PROGRAMMING MODES

Write Mode

Allows the user to enter programs into the 1155's program memory from the keyboard.

Run Mode

Runs programs entered either from the keyboard or from the Model 511 magnetic card reader.

List Mode

Allows the user to list the program in the program memory, entirely or one step at a time, to print a checksum, or to print the number of unused program steps.

CARD READER CONTROL MODE

Allows the user to enter programs into the program memory from the Model 511 magnetic card reader, and to store programs onto magnetic cards from the program memory.

EDITING MODES

Insert Mode

Allows the user to insert new steps in the stored program.

Delete Mode

Allows the user to delete steps from the stored program.

REGISTERS

Numbers are stored in registers. There are

2 WORKING REGISTERS

20 MEMORY REGISTERS

Each register can store one number at a time.

Working Registers

The two working registers in the 1155 are thought of as being "stacked" one on top of the other. For this reason they are named Lower, abbreviated L, and Upper, abbreviated U.

All numeric operations are done with the working registers, with the single exception of ACCUMULATE STORE which operates on L and a memory register.

Numeric operations are divided into two groups called unary and binary operations. Unary operations affect only the L register. An example of a unary operation is CHANGE SIGN. Binary operations affect both L and U. The result is put into L, and U is cleared (set to zero). Examples of binary operations are addition and multiplication.

Memory Registers

The twenty memory registers in the 1155 are numbered 00 through 19. These registers may be used to store numbers.

Numbers are transferred from L to the memory registers using STORE, and from the memory registers to L using RECALL. All memory registers may be cleared (set to zero) using CLR REG.

In addition, memory register 00 may be used to address other registers in conjunction with the INDIRECT key. Otherwise, register 00 behaves exactly like memory registers 01 through 19.

PROGRAM MEMORY

The program memory stores program instructions in the order they are entered. The maximum number of single key instructions is 511; since some instructions require two or more keystrokes (e.g., GO TO INDIRECT), the maximum number of instructions will generally be less than 511.

PROGRAM CODES

A program may contain up to 100 program codes, which are two-digit numbers identifying places in a program to be used as entry points (for example, in program loops, subroutines, etc.). Program codes are numbered 00 through 99.

ERROR MESSAGES

When an error (overflow, argument outside the range of a function, division by zero, undefined term, and so on) is encountered in keyboard, list or running modes, the 1155 prints an ERROR message. The table below lists all error messages with brief explanations.

ERROR	EXPLANATION
1	Exponent in e^x or $\|A\|^x$ positive and large enough to create an answer greater than 10^{100}.
2	The absolute value of the tangent of the angle specified is greater than 10^{100}.
3	Attempt to compute logarithm of zero.
4	Attempt to compute ARC SIN or ARC COS of x for which $\|x\|>1$.
5	Division by zero.
6	Attempt to compute $\|x\|!$ for $\|x\|\geq 100$.
7	Exponent of normalized answer greater than 99 or $\|x\|!$ where $69 < \|x\| < 100$.
8	Attempt to compute reciprocal of zero.
9	Illegal key entry during manual operation.
10	Attempt to compute polar coordinates when rectangular coordinates are both zero (x = 0, y = 0).
11	Incomplete address during list.
12	Incorrect indirect address in REG 00 or incorrect indirect address key sequence.

The ON-OFF switch is located underneath the right side of the 1155. When the 1155 is turned ON:

- It is in keyboard mode.
- It is set to two-decimal place printout.
- The print wheel spins.
- Working registers L and U are set to zero.
- Angles are set to read in degrees and the DEG key light comes on.
- It prints '.00'.

KEYBOARD MODE KEYS

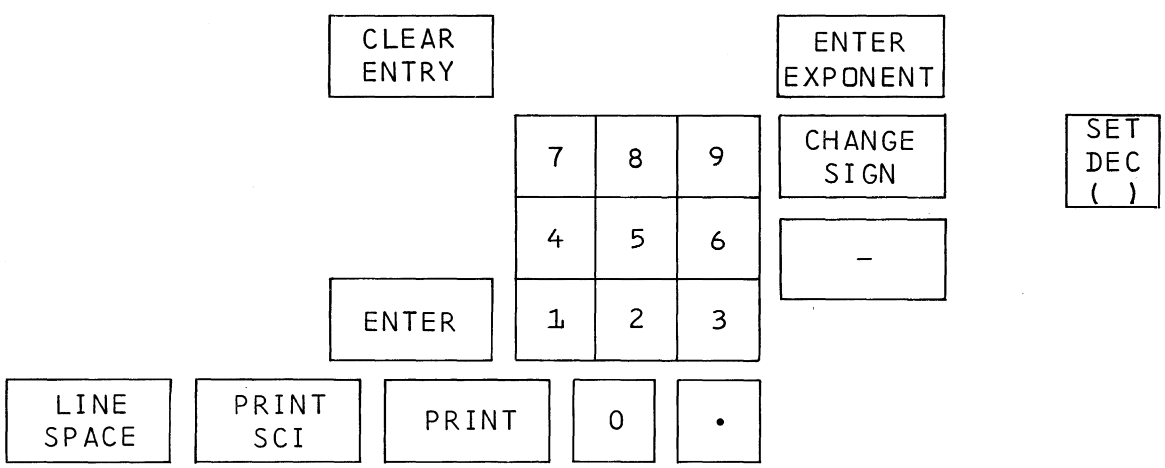
Keyboard mode is the central, "normal" mode of operation. All other modes are reached only through the keyboard mode. In this mode, direct operation of the number entry, printer, arithmetic, memory register and function keys is available to the user.

RESET

Clears both working registers,L and U. Does not clear memory registers or program memory. Returns the 1155 to KEYBOARD MODE from all other modes; for this reason the RESET key may be used to terminate an unknown or undesirable state (for example, an infinite loop in a program). Pressing RESET causes the print wheel to spin and a line space to be printed.

INPUT/OUTPUT

The following keys are used when entering numbers into the 1155 or printing numbers on the printer.



ENTER	PRINT	Terminates entry of a number and causes it to be printed.

In the following examples it is assumed that the number of decimal places for printout has been set to 2. Press RESET.

- Enter 123

PRINTOUT

1	2	3	ENTER

123.00

- Enter 123.45

1	2	3	.	4	5	PRINT

123.45

- Enter .37 (two ways are shown)

.	3	7	ENTER
.	3	7	PRINT

.37
.37

When a number is keyed in:

- The previous content of L (Lower register) is transferred to U (Upper register).
- The number being entered is placed in L.
- The previous content of U is lost.

The following example records the contents of L and U as successive numbers are entered. The number of decimal places for printout is set to 2.

KEYS	L	U	PRINTOUT
RESET	0	0	
2 5 ENTER	25	0	25.00
7 ENTER	7	25	7.00
3 0 ENTER	30	7	30.00
9 9 ENTER	99	30	99.00

A number may contain up to 13 significant digits. All digits (up to 13) are stored in L. However, the printout is rounded to the number of decimal places set for printout. For example, assume that the number of places for printout is set to 2.

KEYS	L	PRINTOUT
1 • 2 3 4 ENTER	1.234	1.23
1 • 2 3 5 ENTER	1.235	1.24
• 0 3 7 9 ENTER	.0379	.04
• 0 0 4 9 ENTER	.0049	.00

CHANGE SIGN	Changes the sign of the number being entered.

In the following examples the printout is set for two decimal places.

PRINTOUT

- Enter -123

					PRINTOUT
1	2	3	CHANGE SIGN	ENTER	-123.00

- Enter -5.9

5	.	9	CHANGE SIGN	ENTER	-5.90

- Enter -.375

.	3	7	5	CHANGE SIGN	ENTER	-.38

- Enter -.0023

.	0	0	2	3	CHANGE SIGN	ENTER	-.00

In the last example, the 1155 printed "negative zero" because the number of decimal places for printout was set to 2. The number is correctly stored in L as -.0023, however.

In the above examples, the CHANGE SIGN key is pressed after entering the last digit of the number. Actually, it may be pressed anytime during entry. For example:

- Enter -123

1	2	CHANGE SIGN	3	ENTER	-123.00

Internally, numbers are stored in scientific notation. In scientific notation, a number is represented by a mantissa and an exponent. For example, the number 123.45 is represented as follows.

$$\underbrace{\underline{1.2345}}_{\text{mantissa}} \times 10^{2} \xleftarrow{} \text{exponent}$$

In the 1155, a number is represented by a 13 digit mantissa and a two digit exponent. The following table indicates how certain numbers are represented in the 1155.

NUMBER	MANTISSA	EXPONENT
123.45	1.234500000000	+02
.0023	2.300000000000	-03
-1234567	-1.234567000000	+23
6.02×10^{23}	6.020000000000	+06
9×10^{99}	9.000000000000	+99
1×10^{-99}	1.000000000000	-99

The 1155 can operate on numbers with up to 13 significant digits in the following ranges:

ZERO:

POSITIVE NUMBERS: $1.000000000000\text{X}10^{-99}$ to $9.999999999999\text{X}10^{99}$

NEGATIVE NUMBERS: $-1.000000000000\text{X}10^{-99}$ to $-9.999999999999\text{X}10^{99}$

PRINT SCI

Terminates entry of a number and causes the number to be printed in scientific notation. The mantissa is rounded to 12 significant digits and the letter "E" is printed between the mantissa and the exponent.

- Enter 123.45 and print in scientific notation.

Keys	Printout
1 2 3 . 4 5 PRINT SCI	1.23450000000E+02

- Enter .0023 and print in scientific notation.

Keys	Printout
. 0 0 2 3 PRINT SCI	2.30000000000E-03

- Enter 7 and print in scientific notation.

Keys	Printout
7 PRINT SCI	7.00000000000E+00

- Enter 0 and print in scientific notation.

Keys	Printout
RESET PRINT SCI	0

The following illustrates the entry of negative numbers followed by a printout in scientific notation.

- Enter -123

Keys	Printout
1 2 3 CHANGE SIGN PRINT SCI	-1.23000000000E+02

- Enter -5.9

Keys	Printout
5 . 9 CHANGE SIGN PRINT SCI	-5.90000000000E+00

- Enter -.375

Keys	Printout
. 3 7 5 CHANGE SIGN PRINT SCI	-3.75000000000E-01

- Enter -.0023

Keys	Printout
. 0 0 2 3 CHANGE SIGN PRINT SCI	-2.30000000000E-03

ENTER EXPONENT

Allows user to enter a number in scientific notation.

- Enter 6.02×10^{23}

$\underbrace{\boxed{6}\ \boxed{\bullet}\ \boxed{0}\ \boxed{2}}_{\text{MANTISSA}}\ \boxed{\text{ENTER EXPONENT}}\ \underbrace{\boxed{2}\ \boxed{3}}_{\text{EXPONENT}}\ \boxed{\text{PRINT SCI}}$

```
6.02000000000E+23
```

- Enter 205×10^{6}

$\boxed{2}\ \boxed{0}\ \boxed{5}\ \boxed{\text{ENTER EXPONENT}}\ \boxed{6}\ \boxed{\text{PRINT SCI}}$

```
2.05000000000E+08
```

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

The following example illustrates the method for entering a negative exponent.

- Enter 3.7×10^{-9}

$\underbrace{\boxed{3}\ \boxed{\bullet}\ \boxed{7}}_{\text{MANTISSA}}\ \boxed{\text{ENTER EXPONENT}}\ \underbrace{\boxed{-}\ \boxed{9}}_{\text{EXPONENT}}\ \boxed{\text{PRINT SCI}}$

```
3.70000000000E-09
```

The following shows how to enter a negative number, expressed in scientific notation.

- Enter -2×10^{6}

$\underbrace{\underbrace{\boxed{2}\ \boxed{\text{ENTER EXPONENT}}\ \boxed{6}}_{2 \times 10^{6}}\ \boxed{\text{CHANGE SIGN}}}_{-2 \times 10^{6}}\ \boxed{\text{PRINT SCI}}$

```
-2.00000000000E+06
```

SET DEC ( )	Sets the number of decimal places for printout. The number of decimal places for printout can be set to any number from 0 to 9 inclusive. Press SET DEC, then press the numeric key for the desired number of places (0-9). The decimal place setting governs only the number of places printed. It has nothing to do with the number of places carried internally.

The following example shows the printout of the same number for various decimal place settings.

KEYS								PRINTOUT
RESET	SET DEC ( )	0						
1	•	2	3	4	5	6	ENTER	1
SET DEC ( )	0	PRINT						1
SET DEC ( )	2	PRINT						1.23
SET DEC ( )	3	PRINT						1.235
SET DEC ( )	5	PRINT						1.23456
SET DEC ( )	9	PRINT						1.234560000

In each case, the printout is rounded to the number of decimal places specified.

When entry of a number is terminated by pressing the PRINT or ENTER keys, the number entered is printed in fixed point if possible. The maximum number of positions for fixed point printout is fifteen. If a number requires more than 15 digits (including decimal places) it will automatically be printed in scientific notation.

KEYS	PRINTOUT
[RESET] [SET DEC ( )] [5]	
[1] [2] [3] [4] [5] [6] [7] [ENTER]	1234567.00000
[SET DEC ( )] [8] [PRINT]	1234567.00000000
[SET DEC ( )] [9] [PRINT]	1.23456700000E+06
[RESET] [SET DEC ( )] [0]	
[2] [ENTER EXPONENT] [1] [0] [ENTER]	20 000 000 000
[SET DEC ( )] [5] [PRINT]	2.0000000000E+10
[RESET] [SET DEC ( )] [0]	
[2] [ENTER EXPONENT] [1] [0] [CHANGE SIGN] [ENTER]	-20 000 000 000
[SET DEC ( )] [5] [PRINT]	-2.0000000000E+10

π

The PI key ( ) causes the contents of L to be transferred into U. The previous contents of U are lost. The mathematical constant pi is placed in the L register correct to 12 decimal places, as follows:

$$\pi = 3.141592653590$$

During printout the value of pi will be rounded to the number of decimal places set for printout (0 through 9) or if printed in scientific notation, pi will be printed as follows (printout set for two decimal places):

RESET	π	PRINT SCI

```
          3.14
3.14159265359E+00
```

LINE SPACE

The LINE SPACE key causes one vertical line space on the printer.

CLEAR ENTRY

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

- Enter 25

2	4	CLEAR ENTRY	2	5	ENTER

mistake

```
25.00
```

When the CLEAR ENTRY key is pressed, the print wheel s pins as a signal that the 1155 has erased the previously entered number. The correct number may then be entered.

- Enter 6 x 10^{22}

6	ENTER EXPONENT	3	CLEAR ENTRY	6	ENTER EXPONENT	2	2	ENTER

mistake

```
6.0000000000E+22
```

NOTE: If entry of a number has already been terminated by pressing the ENTER, PRINT, PRINT SCI keys, then pressing CLEAR ENTRY has no effect. This is also true if an arithmetic key or function key has been pressed.

BINARY OPERATION KEYS

The operations +, -, x, ÷, and $|A|^x$ are called binary operations because they are performed on two numbers and produce a single result. The binary option keys are:

[+] [−] [X] [÷] [$|A|^x$]

These keys perform operations on numbers in L and U. The result is put in L and U is cleared. These keys terminate number entry and cause the number that was entered to be printed.

[X] [÷] and [$|A|^x$] also cause the result to be printed.

[+] Adds the number in L to the number in U and stores the result in L. Causes the previous entry to be printed along with a plus sign (in keyboard mode). U is cleared. To obtain the result, the PRINT key must be pressed following the addition instruction. The first number in each arithmetic operation may be terminated using the ENTER key.

KEYS	L	U	PRINTOUT
[7] [5]	75		
[ENTER]	75		75.00
[3] [7]	37	75	
[+]	112	0	37.00+
[PRINT]	112	0	112.00

[-] Subtracts the number in L from the number in U, stores the result in L, and clears U to zero. In keyboard mode, the previous entry is printed. To obtain the result, the PRINT key must be pressed following the subtraction instruction.

KEYS	L	U	PRINTOUT
[2] [5]	25		
[ENTER]	25		25.00
[9]	9	25	
[-]	16	0	9.00
[PRINT]	16	0	16.00

[X] Multiplies the number in U by the number in L and stores the result in L, then clears U to zero. Terminates a previous number entry (if any) and prints the number. Prints the result automatically.

KEYS	L	U	PRINTOUT
[2] [•] [3]	2.3		
[ENTER]	2.3		2.30
[4] [•] [5]	4.5	2.3	
[X]	10.35	0	4.50 10.35

÷

Divides the number in U by the number in L. and puts the result into L, then clears U to zero. Terminates a previous number entry and causes the number to be printed. Prints the result (quotient) automatically.

KEYS	L	U	PRINTOUT
3 7	37		
ENTER	37		37.00
4	4	37	
÷	9.25	0	4.00 9.25

$|A|^x$

Raises the absolute value of the number in U to the power of the number in L. The result is put in L and U is cleared to zero. Terminates a number entry and causes the number to be printed. Prints the result automatically.

KEYS	L	U	PRINTOUT
2	2		
ENTER	2		2.00
5	5	2	
$|A|^x$	32	0	5.00 32.00

Mixed Operations

The following examples illustrate sequential use of the binary keys.

EXAMPLE: 7 + 5 + 14 + 3 = 29

KEYS	L	U	PRINTOUT
7	7		
ENTER	7		7.00
5	5	7	
+	12	0	5.00+
14	14	12	
+	26	0	14.00+
3	3	26	
+	29	0	3.00+
PRINT	29	0	29.00

TAPE

```
  7.00
  5.00+
 14.00+
  3.00+
 29.00
```

EXAMPLE: 12 - 18 + 33 - 4 = 23

KEYS	L	U	PRINTOUT
12	12		
ENTER	12		12.00
18	18	12	
−	-6	0	18.00−
33	33	-6	
+	27	0	33.00+
4	4	27	
−	23	0	4.00−
PRINT	23	0	23.00

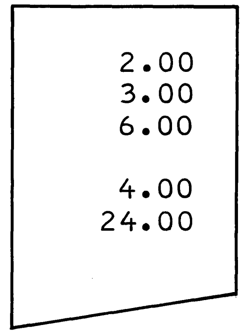
TAPE

```
 12.00
 18.00-
 33.00+
  4.00-
 23.00
```

EXAMPLE: 2 x 3 x 4 = 24

KEYS	L	U	PRINTOUT
2	2		
ENTER	2		2.00
3	3	2	
X	6	0	3.00 6.00
4	4	6	
X	24	0	4.00 24.00

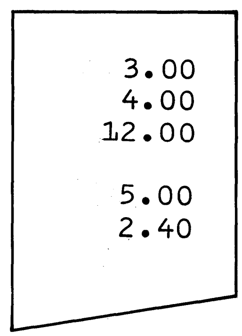
TAPE



EXAMPLE: $\frac{3 \times 4}{5} = 2.4$

KEYS	L	U	PRINTOUT
3	3		
ENTER	3		3.00
4	4	3	
X	12	0	12.00
5	5	12	
÷	2.4	0	5.00 2.40

TAPE



EXAMPLE: $2 \times 3 + 4 = 10$

KEYS	L	U	PRINTOUT
2	2		
ENTER	2		2.00
3	3	2	
			3.00
X	6	0	6.00
4	4	6	
+	10	0	4.00+
PRINT	10	0	10.00

TAPE

```
  2.00
  3.00
  6.00

  4.00+
 10.00
```

EXAMPLE: $7 \times 2^3 - 4 = 52$

KEYS	L	U	PRINTOUT
2	2		
ENTER	2		2.00
3	3	2	
			3.00
$\lvert A \rvert^x$	8	0	8.00
7	7	8	
			7.00
X	56	0	56.00
4	4	56	
–	52	0	4.00–
PRINT	52	0	52.00

TAPE

```
  2.00
  3.00
  8.00

  7.00
 56.00

  4.00-
 52.00
```

EXAMPLE: $\frac{66 + 83 + 75}{3} = 74.67$

KEYS	L	U	PRINTOUT	TAPE
66	66			
ENTER	66		66.00	66.00 83.00+ 75.00+ 224.00 3.00 74.67
83	83	66		
+	149	0	83.00+	
75	75	149		
+	224	0	75.00+	
PRINT	224	0	224.00	
3	3	224		
÷	74.666..*	0	3.00 74.67	

FUNCTION KEYS

The function keys shown below are used to compute the indicated function of the number in L and store the result in L. The number in U is not involved and is not changed in any way.

The operations defined by these keys are called unary operations because they are carried out on a single operant (number).

CHANGE SIGN	$\sqrt{\lvert X \rvert}$	e^X	ARC
1/X	INTGR	$\ln \lvert X \rvert$	SIN
X^2	$\lvert X \rvert !$	$\log \lvert X \rvert$	COS
			TAN

*Carried to 13 significant digits internally.

The TO POLAR and TO RECT keys operate on the number in L and U and produce two results. The results are stored in L and U.

TO POLAR | TO RECT

In keyboard mode, all function keys except CHANGE SIGN and ARC terminate number entry and print the number. The result or results are printed automatically.

CHANGE SIGN — Changes the sign of the number in L from plus to minus or from minus to plus. The sign of zero is not changed, however. It remains plus.

KEYS	L	U	PRINTOUT
5	5	*	
PRINT	5	*	5.00
CHANGE SIGN	-5	*	5.00
CHANGE SIGN	5	*	5.00

1/X — Causes the number in L to be replaced by its reciprocal. The number in the U register is unaffected. Overflow (Error 8) will occur if L is equal to zero.

KEYS	L	U	PRINTOUT
4	4	*	
1/X	.25	*	4.00 .25
.2	.2	.25	
1/X	5	.25	.20 5.00

PROBLEM: Given a and b compute $\frac{1}{a+b}$

KEYS	L	U	PRINTOUT
RESET	0	0	
Key in a	a	0	
ENTER	a	0	a
Key in b	b	a	
+	a+b	0	b+
1/X	1/(a+b)	0	1/(a+b)

*The CHANGE SIGN and 1/X keys do not change the number in U. Remember, however, that entry of a new number causes the number in L to be transferred to U.

X^2 Causes the number in L to be multiplied by itself (squared). The number in U is unaffected. Overflow (Error 7) will occur if the absolute value of the number in L is greater than or equal to 10^{50}.

KEYS	L	U	PRINTOUT
RESET	0	0	
5	5	0	
X^2	25	0	5.00 25.00
12	12	25	
X^2	144	25	12.00 144.00

$\sqrt{\lvert X \rvert}$ Causes the number in L to be replaced by the square root of its absolute value. The number in U is unaffected.

KEYS	L	U	PRINTOUT
RESET	0	0	
25	25	0	
$\sqrt{\lvert X \rvert}$	5	0	25.00 5.00
3	3	25	
CHANGE SIGN	-3	25	
$\sqrt{\lvert X \rvert}$	1.732...	25	-3.00 1.73

PROBLEM: Given a and b compute $\sqrt{a^2 + b^2}$

KEYS	L	U	PRINTOUT
RESET	0	0	
Key in a	a	0	
X^2	a^2	0	a a^2
Key in b	b	a	
X^2	b^2	a	b b^2
+	$a^2 + b^2$	0	+
$\sqrt{\lvert X \rvert}$	$\sqrt{a^2 + b^2}$	0	$\sqrt{a^2 + b^2}$

INTGR Causes the number in L to be replaced by its integer part. The number in U is unaffected.

KEYS	L	U	PRINTOUT
8	8	0	
INTGR	8	0	8.00 8.00
5.99	5.99	8	
INTGR	5	8	5.99 5.00
13.01	13.01	5	
INTGR	13	5	13.01
2.9	2.9	13	13.00
CHANGE SIGN	-2.9	13	
INTGR	-2	13	-2.90 -2.00

|X|! Causes the number in L to be replaced by its absolute value and rounded. L is then replaced by the factorial function* of the resultant number. This operation does not affect the number of U. Overflow (Error 1) will result if the absolute value of the original number in L is greater than or equal to 69.5. A different error (Error 6) will occur if the absolute value of the original number in L is greater than or equal to 100.

KEYS	L	U	PRINTOUT
RESET	0	0	
5	5	0	
|X|!	120	0	5.00 120.00
4.49	4.49	120	
|X|!	24	120	4.49 24.00
4.5	4.5	24	
CHANGE SIGN	-4.5	24	
|X|!	120	24	-4.50 120.00

*The factorial function is defined as follows.

$$x! = \begin{cases} 1 \text{ if } x = 0. \\ 1(2)(3). \ . \ .(x) \text{ if } x \text{ is a positive integer.} \end{cases}$$

Exponential and Logarithmic Functions

Two exponential function keys are available on the 1155.

The $|A|^x$ key is described in the Binary Operations section.

The e^x key is described below.

The number e is the base for natural or Napierian logarithms. To 12 decimal places:

$$e = 2.718281828459$$

$\boxed{e^x}$ Causes the number in L to be replaced by e^x where x is the original number in L. U is not affected. Overflow (Error 7) will occur if the number in L is greater than $\ln 10^{100}$ (approximately 230.25851).

KEYS	L	U	PRINTOUT
RESET	0	0	
2	2	0	
e^x	7.389...	0	2.00 7.39
1	1	7.389...	
CHANGE SIGN	-1	7.389...	
e^x	.367...	7.389...	-1.00 .37

PROBLEM: Given x, compute e^{-x^2}

KEYS	L	U	PRINTOUT
RESET	0	0	
Key in x	x	0	
X^2	x^2	0	x x^2
CHANGE SIGN	$-x^2$	0	
e^x	e^{-x^2}	0	$-x^2$ e^{-x^2}

Two types of logarithms are available on the 1155:

- Base e logarithms using the ln|x| key.
- Base 10 logarithms using the log|x| key.

ln|x| Computes the logarithm base e (natural log) of the absolute value of the number in L and puts the result in L. U is not affected. An attempt to compute the logarithm of zero will cause an error condition (Error 3).

KEYS	L	U	PRINTOUT
RESET	0	0	
2	2	0	
ln|x|	.693...	0	2.00 .69
10	10	.693...	
CHANGE SIGN	-10	.693...	
ln|x|	2.302...	.693...	-10.00 2.30

log|x| Computes the logarithm base 10 (common log) of the absolute value of the number in L and puts the result in L. U is not affected. An attempt to compute the logarithm of zero will cause an error condition (Error 3).

KEYS	L	U	PRINTOUT
RESET	0	0	
2	2	0	
log|x|	.301...	0	2.00 .30
1000	1000	.301...	
CHANGE SIGN	-1000	.301...	
log|x|	3	.301...	-1000.00 3.00
RESET	0	0	
log|x|	0	0	ERROR 3

Trigonometric Functions: SIN, COS, TAN

The following keys are used to compute the sine, cosine or tangent of a number.

SIN COS TAN

The sine, cosine and tangent functions may be computed for arguments expressed in either degrees or radians.

• DEG — Turns on the degrees light (above the DEG key) and specifies that the trigonometric functions operate on arguments in degree measure. The DEG key does not affect the numbers in the registers. When the 1155 is turned on it is set for degrees. The degree status remains in effect until the RAD key is pressed.

• RAD — Turns on the radians light (above the RAD key) and specifies that the trigonometric functions operate on arguments in radian measure. The RAD key does not affect the numbers in the registers. The radian status remains in effect until the DEG key is pressed.

The relationships between degree measure and radian measure are shown below.

$$\text{DEGREES} = \text{RADIANS} \times \frac{180}{\pi}$$

$$\text{RADIANS} = \text{DEGREES} \times \frac{\pi}{180}$$

SIN — Replaces the number in L by the sine of the number in L.

COS — Replaces the number in L by the cosine of the number in L.

TAN — Replaces the number in L by the tangent of the number in L. If the number in L is an odd multiple of 90 (DEG light on) or a multiple of /2 (RAD light on) an error message will be printed (Error 2).

To compute sin x, cos x or tan x:

- If not in keyboard mode, press RESET.
- Set number of decimal places for printout.
- Set 1155 to degrees or radians.
- For each value of x,

  ... key in x
  ... press SIN or COS or TAN.

For example, compute sin 60°, cos 45°, and tan 1000°.

KEYS	L	U	PRINTOUT
RESET	0	0	
SET DEC 5	0	0	
DEG	0	0	
60	60°	0	
SIN	SIN 60°	0	60.00000
			.86603
45	45°	SIN 60°	
COS	COS 45°	SIN 60°	45.00000
			.70711
1000	1000	COS 45°	
TAN	TAN 1000°	COS 45°	1000.00000
			-5.67128

The secant (sec), cosecant (csc) and cotangent (cot) of a number are defined as follows.

$$\sec x = \frac{1}{\cos x} \qquad (\cos x \neq 0)$$

$$\csc x = \frac{1}{\sin x} \qquad (\sin x \neq 0)$$

$$\cot x = \frac{1}{\tan x} \qquad (\tan x \neq 0)$$

Sec x, csc x and cot x are computed using the COS, SIN, TAN and 1/X keys, as follows.

COS 1/X Computes the secant of the number in L and puts the result in L. U is not affected.

SIN 1/X Computes the cosecant of the number in L and puts the result in L. U is not affected.

TAN 1/X Computes the cotangent of the number in L and puts the result in L. U is not affected.

NOTE: If the sine, cosine, or tangent is zero, then an error condition (Error 8) occurs when the 1/X key is pressed.

EXAMPLE: Compute sec 30°

KEYS	L	U	PRINTOUT
RESET	0	0	
SET DEC 5	0	0	
DEG	0	0	
30	30	0	
COS	.86603...	0	30.00000 .86603
1/X	1.15470...	0	1.15470 ← SEC 30°

Inverse Trigonometric Functions

The ARC key is used as a prefix to the SIN, COS and TAN keys to compute the arcsine, arccosine and arctangent functions.

ARC SIN — Computes the arcsine of the number in L and puts the result in L. U is not affected. The result will be in degrees if the DEG light is on or in radians if the RAD light is on. If the absolute value of the original number in L is greater than 1, an error condition (Error 4) occurs.

ARC COS — Computes the arccosine of the number in L and puts the result in L. U is not affected. The result will be in degrees if the DEG light is on or radians if the RAD light is on. If the absolute value of the original number in L is greater than 1, an error condition (Error 4) occurs.

ARC TAN — Computes the arctangent of the number in L and puts the result in L. U is not affected. The result will be in degrees if the DEG light is on or in radians if the RAD light is on.

The 1155 computes the principal value of the arcsine, arccosine or arctangent, as shown below.

DEG light on:

$$-90^\circ \leq \text{ARC SIN } x \leq 90^\circ$$
$$0^\circ \leq \text{ARC COS } x \leq 180^\circ$$
$$-90^\circ \leq \text{ARC TAN } x \leq 90^\circ$$

RAD light on:

$$-\pi/2 \leq \text{ARC SIN } x \leq \pi/2$$
$$0 \leq \text{ARC COS } x \leq \pi$$
$$-\pi/2 \leq \text{ARC TAN } x \leq \pi/2$$

To compute arcsin x or arccos x or arctan x:

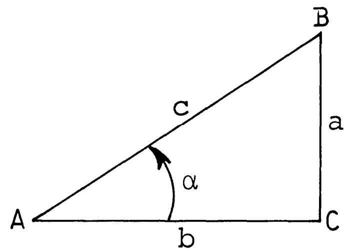
1. If not in keyboard mode, press RESET.
2. Set number of decimal places for printout.
3. Set 1155 to degrees or radians.
4. For each value of x:
   a. key in x
   b. press ARC, then SIN
      or ARC, then COS
      or ARC, then TAN.

For example, compute arcsin .5, arccos (-.5) and arctan 1000 with results in degrees.

KEYS	L	U	PRINTOUT
RESET	0	0	
SET DEC 5	0	0	
DEG*	0	0	
.5	.5	0	
ARC			
SIN	30	0	.50000
			30.00000
.5	.5	30	
CHANGE SIGN	-.5	30	
ARC			
COS	120	30	-.50000
			120.0000
1000			
ARC			
TAN	89.94270...	120	1000.00000
			89.94270

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

In the following right triangle, a is the measure of angle BAC.



$\sin\alpha = \frac{a}{c}$; $\cos\alpha = \frac{b}{c}$; $\tan\alpha = \frac{a}{b}$ $\alpha = \arcsin\frac{a}{c} = \arccos\frac{b}{c} = \arctan\frac{a}{b}$

Given a and c, compute α in radians.

KEYS	L	U	PRINTOUT
RESET	0	0	
SET DEC 5	0	0	
RAD	0	0	
Key in a	a	0	
ENTER	a	0	a
Key in c	c	a	
÷	a/c	0	c a/c
ARC			
SIN	arcsin (a/c)	0	arcsin (a/c)

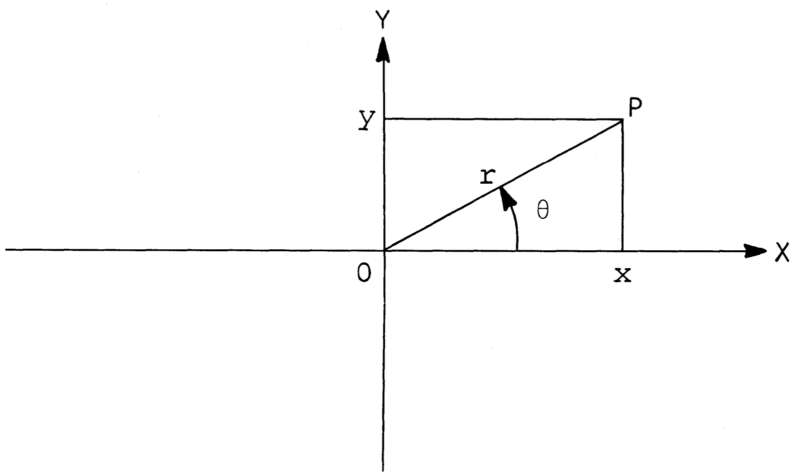
For example, if a = 3 and c = 5:

```
3.00000
5.00000
 .60000

 .64350   <----  α (radians)
```

Rectangular and Polar Coordinates

The following diagram and discussion show the relationship between rectangular coordinates (x, y) and polar coordinates (r, θ).



Conversion from rectangular to polar coordinates:

$$x = r \cos \theta$$

$$y = r \sin \theta$$

Conversion from polar to rectangular coordinates:

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

$$\theta = \tan^{-1} \frac{y}{x}$$

where: $-180 < \theta \leq 180$ (degrees)

$-\pi < \theta \leq \pi$ (radians)

TO RECT

Converts polar coordinates (r, θ) where r is in U and θ is in L to rectangular coordinates (x,y). Stores the results in U and L with x in U and y in L.

	Before pressing TO RECT	After pressing TO RECT
U	r	x
L	θ	y

If θ is given in degrees, the 1155 must be in DEG status before TO RECT is pressed.

If θ is given in radians, the 1155 must be in RAD status before TO RECT is pressed.

To convert from polar to rectangular coordinates:

1. If not in keyboard mode, press RESET.
2. Set number of decimal places for printout.
3. Select degrees (DEG key) or radians (RAD key).
4. For each set of values (r, θ):
   a. Key in r, press ENTER.
   b. Key in θ.
   c. Press TO RECT.

$r = 5,\ \theta = 150^{\circ}$

```
     5.00000
   150.00000
    -4.33013  <-- x
     2.50000  <-- y
```

$r = 2,\ \theta = 6$ rad

```
     2.00000
     6.00000
     1.92034  <-- x
     -.55883  <-- y
```

TO POLAR

Converts rectangular coordinates (x, y) where x is in U and y is in L to polar coordinates (r, θ). Stores the results in U and L with r in U and θ in L.

U	x
L	y

Before pressing TO POLAR

U	r
L	θ

After pressing TO POLAR

If the 1155 is in DEG status, θ will be in degrees in the range:

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

If the 1155 is in RAD status, θ will be in radians in the range:

$$-\pi < \theta \leq \pi$$

To convert from polar coordinates to rectangular coordinates:

1. If not in keyboard mode, press RESET.
2. Set number of decimal places for printout.
3. Select degrees (DEG key) or radians (RAD key).
4. For each set of values (x, y).
   a. Key in x, press ENTER.
   b. Key in y.
   c. Press TO POLAR.

x = 3, y = 4

```
  3.00000
  4.00000
  5.00000  <-- r
 53.13010  <-- θ°
```

x = 3, y = -4

```
  3.00000
 -4.00000
  5.00000  <-- r
  -.92730  <-- θ rad
```

MEMORY REGISTERS

The 1155 has 20 memory registers. Each memory register is identified by a two digit number (00 through 19) called the address of the register.

00	0	05	0	10	0	15	0
01	9	06	0	11	0	16	0
02	4	07	0	12	0	17	0
03	0	08	-5	13	3.7	18	0
04	0	09	0	14	0	19	6.02E23

A memory register can store one number. The above register diagram indicates that:

9 is stored in register 01
4 is stored in register 02
-5 is stored in register 08
3.7 is stored in register 13
6.02E23 is stored in register 19

All other registers contain 0.

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

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

[CLEAR ALL REGS] Clears all registers. Stores zeros in L, U and memory registers 00 through 19.

The CLR ALL REGS key has a lock to prevent accidental clearing of the registers.

The STORE, RECALL and ACCSTORE keys must always be followed by a two digit address designating the memory register involved in the operation. Examples of legal operations are shown below

KEYS			ABBREVIATION	COMMENTS
[STORE ( ) ( )]	[0]	[0]	STORE 00	Store in register 00.
[STORE ( ) ( )]	[1]	[3]	STORE 13	Store in register 13.
[RECALL ( ) ( )]	[0]	[1]	RECALL 01	Recall from register 01.
[ACCUM STORE ( ) ( )]	[1]	[9]	ACCSTORE 19	Accumulate store in register 19.

The following examples are not correct memory operations.

KEYS			REASON WHY NOT CORRECT
[STORE ( ) ( )]	[8]		Must use two digit address.
[RECALL ( ) ( )]	[2]	[0]	There is no register 20. Addresses must be whole number, 00 through 19.

STORE
( ) ( )

Stores the number in L in the memory register designated by the two digit address following STORE. A STORE operation terminates a number entry and causes the number to be printed. U is not affected.

EXAMPLE: Key in 2.3 and store it in register 17. Then key in 9 and store in register 05 and 06.

KEYS	L	U	05	06	17	PRINTOUT
RESET	0	0				
2.3	2.3	0				
STORE 17	2.3	0			2.3	2.30 STORE 17
9	9	2.3				
STORE 05	9	2.3	9		2.3	9.00
STORE 06	9	2.3	9	9	2.3	STORE 05 STORE 06

The STORE operation put a copy of the number in L into the designated memory register, replacing the previous content of the memory register. Note that the operation does not change L.

RECALL
( ) ( )

Recalls the number in the indicated memory register to L. The previous content of L is transferred to U. A RECALL operation does not change the content of the memory registers. In keyboard mode, the number that is recalled is also printed.

The following example shows the contents of L, U and memory register 06 before and after a RECALL 06 operation.

	L	U	REG 06
BEFORE	-5	7	2.3
AFTER	2.3	-5	2.3

The following example illustrates the use of STORE and RECALL.

EXAMPLE: $2 \times 3 + 4 \times 5 = 26$

KEYS	L	U	01	PRINTOUT
RESET	0	0		
2	2	0		
ENTER	2	0		2.00
3	3	2		
X	6	0		3.00 6.00
STORE 01	6	0	6	STORE 01
4	4	6	6	
ENTER	4	6	6	4.00
5	5	4	6	
X	20	0	6	5.00 20.00
RECALL 01	6	20	6	6.00
+	26	0	6	+
PRINT	26	0	6	26.00

ACCUM STORE () ()

Adds the number in L to the number in the designated memory register and stores the result in the memory register. Does not change the number in L. U is unaffected.

KEYS	L	U	00	01	02 ...
RESET	0	0	0	0	0
CLR REGS	0	0	0	0	0
8	8	0	0	0	0
ACCSTORE 01	8	0	0	8	0
X^2	64	0	0	8	0
ACCSTORE 02	64	0	0	8	64
5	5	64	0	8	64
ACCSTORE 01	5	64	0	13	64
X^2	25	64	0	13	64
ACCSTORE 02	25	64	0	13	89
7	7	25	0	13	89
ACCSTORE 01	7	25	0	20	89
X^2	49	25	0	20	89
ACCSTORE 02	49	25	0	20	138
RECALL 01	20	49	0	20	138
RECALL 02	138	20	0	20	138

If the operations are actually carried out, the printout should look like the following.

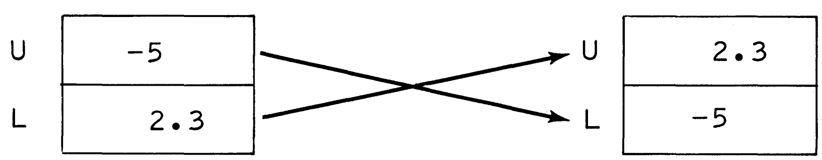
```
     .000000
    8.000000
ACCSTORE 01
   64.000000
ACCSTORE 02
    5.000000
ACCSTORE 01
   25.000000
ACCSTORE 02
    7.000000
ACCSTORE 01
   49.000000
ACCSTORE 02
   20.000000 ← Σx
  138.000000 ← Σx²
```

Exchange

If the RECALL key is pressed twice, without keying a register address, the 1155 exchanges the contents of L and U.

RECALL ( ) ( )	RECALL ( ) ( )	Exchanges the contents of L and U.

The following example shows the contents of L and U before and after pressing RECALL RECALL.



Before pressing RECALL RECALL   After pressing RECALL RECALL

EXAMPLE: 3 - 4 x 5 = 17

KEYS	L	U	PRINTOUT
RESET	0	0	
4	4	0	
ENTER	4	0	4.00
5	5	4	5.00
X	20	0	20.00
3	3	20	
RECALL	3	20	
RECALL	20	3	3.00
			20.00
-	-17	0	-
PRINT	-17	0	-17.00

The above problem can be done with fewer keystrokes as follows.

RESET, 4, ENTER	4.00
5, X	5.00
	20.00
CHANGE SIGN	-20.00
3, +	3.00+
PRINT	-17.00

STORED PROGRAMS AND PROGRAMMING MODES

A stored program is a sequence of instructions stored in the program memory of the 1155. An instruction is equivalent to one, two or three keystrokes. Up to 511 keystrokes may be stored in the 1155's program memory.

There are five programming modes.

- WRITE mode is used to store a program in the program memory.
- LIST mode is used to list the stored program on the printer. Instructions are listed as abbreviations.
- RUN mode is used to run the stored program. In RUN mode the 1155 automatically performs the instructions stored in the program memory.
- INSERT mode is used to insert instructions into the stored program.
- DELETE mode is used to delete instructions from the stored program.

The INSERT and DELETE modes are also called edit modes since they are used in editing and debugging a program (see Section 4).

ABBREVIATIONS FOR INSTRUCTIONS DISCUSSED IN SECTION 2

KEYED IN AS			LISTED AS	COMMENTS
$\|A\|^x$			A TO X	
ACCUM STORE () ()	1	2	ACCSTORE 12	Any memory register 00 - 19 may be used instead of "12".
ARC			ARC	
CHANGE SIGN			OPP SIGN	
CLEAR ALL REGS			CLR REGS	
COS			COS	

KEYED IN AS	LISTED AS	COMMENTS
DEG	DEGREES	
e^x	E TO X	
ENTER	ENTER	
ENTER EXPONENT	ENT EXP	
|X|!	FACTRL	
INTGR	INTEGER	
log|X|	log 10	
ln|X|	ln E	
π	PI	
PRINT	PRINT	
PRINT SCI	PRINT SCI	
RAD	RADIANS	
RECALL ( ) ( ) 1 0	RECALL 10	Any memory register 00 - 19 may be used insteas of "10".
RECALL ( ) ( ) 9 9	RECALL 99	Any integer greater than 19 causes L to be copied into U.
RECALL ( ) ( ) RECALL ( ) ( )	EXCG	Exchanges L and U.
1/X	RECIP	

KEYED IN AS	LISTED AS	COMMENTS
SET DEC ( )	SET DEC	
SIN	SIN	
LINE SPACE	SPACE	
$\sqrt{\lvert X \rvert}$	SQ RT	
STORE () () 1 8	STORE 18	Any memory register 00 - 19 may be used instead of "18".
TAN	TAN	
TO POLAR	TO POLAR	
TO RECT	TO RECT	
X^2	X SQ	
+	+	
−	−	
X	X	
÷	÷	
0	0	ZERO
1	1	Any numeric 0 - 9.
•	•	Decimal point.

In this section we will introduce these additional programmable instructions.

KEYED IN AS	LISTED AS	COMMENTS
[ACCUM STORE () ()] [IND (00)]	ACCSTORE N	
[GO TO () ()] [5] [6]	GO TO 56	Any program code 00 - 99 may be used instead of "56".
[GO TO () ()] [IND (00)]	GO TO N	
[IF NEG () ()] [5] [6]	IF - GO TO 56	Any program code 00 - 99 may be used instead of "56".
[IF NEG () ()] [IND (00)]	IF - GO TO N	
[PROG CODE ()()] [5] [6]	56	Any program code may be used.
[RECALL () ()] [IND (00)]	RECALL N	
[STOP]	STOP	
[STORE () ()] [IND (00)]	STORE N	

In addition, the following keys, located on the right side of the keyboard, are used to write, list, run and edit programs.

RESET	DELETE	INSERT	STORE IN CARD
START	WRITE	CLEAR PROG	
ONE STEP	LIST	LOAD FROM CARD	

The above keys are called non-programmable keys since they may not be used as instructions in a stored program.

All other keys on the keyboard, except the CLEAR ENTRY key, may be used as instructions in a stored program.

When the 1155 is running a stored program it behaves exactly as if the keys stored in the program memory were being pressed one after the other in keyboard mode, except:

- No printing is done unless PRINT or PRINT SCI instructions are used in the program.
- An error of any kind causes the 1155 to return to keyboard mode. If this happens L and U are cleared.

SAMPLE PROGRAM

The following is a program to compute the value of

$$\frac{x}{x^2 - 1}$$

for a given value of x. The program is shown as it would appear when listed on the printer in list mode.

```
                     ++678 <———— checksum
program code ——> 27
                    SPACE
                     STOP
                    PRINT
                    STORE 01
                     X SQ
                        1
                        -          instructions
                   RECALL 01
                     EXCG
                        ÷
                    PRINT
                    GO TO 27
```

In the above program, each instruction is equivalent to 1, 2, or 3 keystrokes. Each instruction is listed as the abbreviation for the keystrokes it represents.

To use the above program:

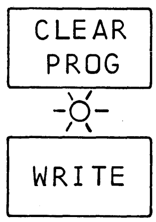
WRITE the program in the 1155's program memory.

LIST the program and verify it.

RUN the program.

WRITE MODE

Write mode is used to store a program in the program memory of the 1155.



Clears program memory. Erases all program steps. The CLEAR PROG key has a lock which prevents accidental erasure of program memory.

Write mode is used to enter programs in the 1155's program memory from the keyboard. To enter write mode from the keyboard mode, press the WRITE key. This will cause the light above the key to go on.

Up to 511 keys may be programmed into the 1155. However, if the key is not programmable (except for RESET) the 1155 will ignore it. That is, it will not be stored in the program memory.

Pressing RESET will cause the 1155 to return to keyboard mode and the light above the WRITE key will go out.

If there is already a program in the program memory. WRITE will cause new program steps to be appended to the program.

To Write a Program

1. Press RESET (to put the 1155 into keyboard mode)
2. Press CLEAR PROG (see note below)
3. Press WRITE (to put the 1155 into write mode)
4. Key in the program.

NOTE: CLEAR PROG is pressed only if it is desired to erase an already stored program or set of programs. Omit this step if the intent is to append steps to a program already stored.

Write the following program. Press, in order, the keys corresponding to the instructions, as follows.

KEYS	COMMENTS
PROG CODE 27	3 keys
LINE SPACE	
STOP	
PRINT	
STORE 01	3 keys
X SQ	x^2 key
1	
-	
RECALL 01	3 keys
RECALL RECALL	2 keys
÷	
PRINT	
GO TO 27	3 keys

The program is stored. Press RESET to terminate write mode and put the 1155 into keyboard mode.

LIST MODE

List mode is used to list (print out on the printer tape) the program in the 1155's program memory. The program may be listed in full or one step at a time.

List mode is also used to compute and print a checksum of the stored program. A checksum is a number which "represents" the program. It is used to verify that a program has been correctly stored, either by hand or using the Model 511 Magnetic Card Reader.

To enter list mode from keyboard mode, press the LIST key. The light above the key will come on indicating that the 1155 is in list mode.

LIST	Puts the 1155 in list mode. It is now ready to list instructions beginning with the one at which it is now stopped.
LIST +	Causes the 1155 to compute and print a checksum of the program then stop at the first instruction in program memory.
START	In list mode, causes the 1155 to start listing, beginning with the step at which the machine is stopped and continuing through the last instruction in program memory.
ONE STEP	In list mode, causes the 1155 to list one key step.

When listing one step at a time, instructions that are programmed by two keystrokes (such as EXCG) will require two depressions on ONE STEP. Instructions that are programmed by three keystrokes (for example, GO TO 23) require three depressions of ONE STEP.

To list the instructions in program memory:

1. Press RESET (omit if already in keyboard mode)
2. Press LIST, + (checksum is printed)
3. Press START

For example:

```
RESET, LIST +
START
```

checksum

```
      ++678
27
     SPACE
      STOP
     PRINT
     STORE 01
     X SQ
         1
         -
    RECALL 01
     EXCG
         ÷
     PRINT
     GO TO 27
```

The following shows the first few steps of the above program listed using ONE STEP instead of START.

```
RESET, LIST +        ++678
ONE STEP         PROG CODE
ONE STEP                 2
ONE STEP                 7
ONE STEP             SPACE
ONE STEP              STOP
ONE STEP             PRINT
ONE STEP             STORE
ONE STEP                 0
ONE STEP                 1
```

Unused Program Steps

If ONE STEP is pressed immediately following a listing the 1155 prints the number of unused program storage positions. The 1155 has 511 program storage positions, each equivalent to one keystroke. The program below requires 20 positions. Therefore, there are 511 - 72 or 489 positions still available. Instructions such as SPACE, PRINT and + require one position. PROG CODE and GO TO instructions require three positions.

Press LIST,+,START

```
          ++678
27
      SPACE
       STOP
      PRINT
      STORE 01
       X SQ
          1
          -
     RECALL 01
       EXCG
          ÷
      PRINT
      GO TO 27
```

```
PRESS ONE STEP ──────►
Unused program space        N489
```

RUN MODE

In run mode, the 1155 executes instructions previously stored in the program memory. Run mode is entered from keyboard mode by pressing either the START key or the ONE STEP key.

START

If the 1155 is in keyboard mode, pressing START causes it to start executing instructions stored in the program. The 1155 continues executing instructions until one of the following occurs:

- If the 1155 encounters a STOP instruction it stops executing the program and returns to keyboard mode. If START is pressed, the 1155 will resume execution of the program, beginning with the instruction following the STOP.
- The operator may manually press STOP. The 1155 will complete the instruction in progress, stop and return to keyboard mode. Pressing START will cause the 1155 to resume program execution from the point at which it was stopped.
- If an error results from the execution of a stored instruction the 1155 prints an error message and returns to keyboard mode. If this happens the L and U registers are cleared.
- The user may press the RESET key while the program is running. This will cause the 1155 to return to keyboard mode immediately and clear L and U. Pressing START will not always restart the 1155 at the place at which it was interrupted.

After executing the last instruction in program memory, the 1155 will automatically "wrap around" to the first step in program memory and continue execution.

Assume the following program has been entered into memory.

```
27
    SPACE
     STOP
    PRINT
    STORE 01
     X SQ
        1
        -
   RECALL 01
     EXCG
        ÷
    PRINT
    GO TO 27
```

Computes $\frac{x}{x^2 - 1}$

To RUN the program:

- Press RESET (omit if already in keyboard mode)
- Press SET DEC 5
- Press GO TO 27 (3 keystrokes)
- Press START

The 1155 searches for PROG CODE 27, finds it and begins running the program. It performs a SPACE then stops at the STOP instruction.

- Key in x = 2, press START.

The 1155 prints the value of x	2.00000
and the value of $x/(x^2 - 1)$	.66667

- Key in x = 3, press START.

The 1155 prints the value of x	3.00000
and the value of $x/(x^2 - 1)$	.37500

- Key in x = .7, press START

The 1155 prints the value of x	.70000
and the value of $x/(x^2 - 1)$	-1.37255

This may be repeated for as many values of x as desired. When finished, press RESET.

STOP

Halts automatic execution of a program and returns the 1155 to keyboard mode. All normal keyboard operations may be performed. Automatic execution of the program may be continued by pressing START.

The most common use of the STOP instruction is to stop execution of the program to permit entry of a number by the operator. After keying in the number, the operator presses START and the 1155 resumes execution of the program.

EXAMPLE: Given r and h, compute $V = \pi r^2 h$.

PROGRAM	COMMENTS
23	
SPACE	
STOP	Operator keys in r and presses START.
PRINT	
X SQ	
STOP	Operator keys in h and presses START.
PRINT	
X	
π	
X	
PRINT	
GO TO 23	

PROGRAM CODE AND GO TO

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.

A GO TO instruction instructs the 1155 to search for a particular program code and, on finding it, to begin executing instructions beginning with the instruction immediately following the program code.

The following program has two program codes and one GO TO instruction.

```
program code ————————→ 10
                           SPACE
                            STOP
                           PRINT
                           STORE 01
program code ————————→ 07
                           SPACE
                            STOP
                           PRINT
                            X SQ
                          RECALL 01
                               X
                           PRINT
GO TO instruction ————————→ GO TO 07
```

In the above program, the program code 10 is used to mark the beginning of the program. The program code 07 is referenced by the GO TO 07 instruction.

PROG CODE ( ) ( )

The PROGRAM CODE key is used to store a two digit program code at a particular point in a program. Programming this operation requires three keystrokes:

PROG CODE, digit, digit.

For example, the program code 10 is programmed as follows:

PROG CODE ( ) ( ) | 1 | 0

GOTO ( ) ( )

Causes the 1155 to search for a specific program code. Programming this operation requires three keystrokes:

GO TO, digit, digit

For example, GO TO 07 is programmed as follows:

GOTO ( ) ( ) | 0 | 7

In run mode, execution of the GO TO instruction causes program execution to continue from the instruction immediately following the indicated program code.

In keyboard mode, a GO TO instruction causes the 1155 to find the indicated program code, then return to keyboard mode. Automatic execution can then be initiated by pressing the START key.

If the 1155 is instructed to search for a nonexistent program code, it will continue searching indefinitely. The search can be terminated by pressing the RESET key.

It is customary to begin a program with a program code. Some programs may have several "starting points," each one used for a different purpose. In this case, a different program code is used for each starting point. It may also be desired to store two or more programs in the 1155 at the same time. Each program should begin with a different program code.

EXAMPLE: Assume that four programs are stored in the program memory of the 1155. Program 1 begins at PROG CODE 10, Program 2 begins at PROG CODE 20, Program 3 begins at PROG CODE 30 and Program 4 begins at PROG CODE 40.

To Run Program 1	To Run Program 2
RESET	RESET
GO TO 10	GO TO 20
START	START

To Run Program 3 ______________ To Run Program 4 ________________

```
RESET                    RESET
GO TO 30                 GO TO 40
START                    START
```

A listing may begin at any program code. For example, suppose the following program is stored in the 1155.

```
              ++200
10
         CLR REGS
            SPACE
13
             STOP
            PRINT
         ACCSTORE 01
                1
         ACCSTORE 02
            GO TO 13
17
            SPACE
           RECALL 01
            PRINT
           RECALL 02
            PRINT
                ÷
            PRINT
             STOP
            GO TO 10
```

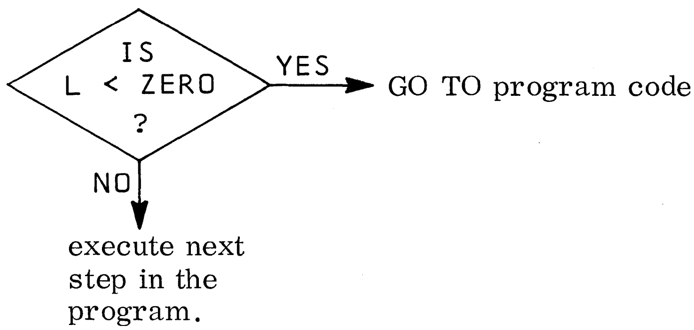
- List the program, beginning at PROG CODE 17.

```
RESET ←———————— (if not in keyboard mode)
GO TO 17
LIST
START                    SPACE
                        RECALL 01
                         PRINT
                        RECALL 02
                         PRINT
                             ÷
                         PRINT
                          STOP
                         GO TO 10
```

The listing begins with the SPACE instruction following PROG CODE 17 and continues to the end of the program.

IF NEG ( ) ( )	If the content of L is negative (less than zero) a GO TO is executed to the designated program code. If L is positive or zero, GO TO is not executed but the program code is checked for validity, just as if a GO TO instruction had been used. An IF - GO TO instruction is sometimes termed "conditional transfer."

The IF - GO TO instruction works as follows:



The IF - GO TO instruction may be used to:

1. Terminate a loop
2. Perform logic operations (conditional GO TO)

Terminating a Loop

EXAMPLE: Print the integers from 0 to 10, in descending order, then stop.

PROGRAM:

```
            ++921
10
                1      ┐
                0      ├── Store 10 in register 05
         STORE 05      ┘
20
        RECALL 05          Put content of 05 in L
     IF - GO TO 99 ──────────────────────────────────┐
         PRINT             Print L                    │
                1      ┐                              │ L < 0
      OPP SIGN         ├── Subtract 1 from register 05│
     ACCSTORE 05       ┘                              │
         GO TO 20  (to 20)                            │
99 ◄──────────────────────────────────────────────────┘
        STOP
```

The IF - GO TO instruction causes the program to jump to program code 99 when the content of register 05 becomes negative as a result of successive subtraction. The STOP instruction is then executed. If L is positive or zero, the IF - GO TO 99 instruction is not executed, and the program continues to the step following the IF - GO TO 99.

Condition GO TO

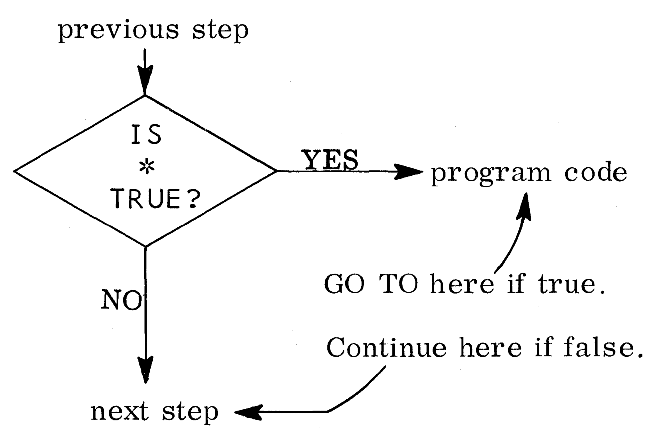
The IF - GO TO key may be used to program instructions of the form

IF * GO TO

where the "*" may stand for a condition other than "-" (for example, IF L = 0 GO TO or IF L > U GO TO).

An IF * GO TO is called a conditional GO TO; where the "*" represents the condition under which the GO TO is executed. Conditional GO TO's are also called branches.

The following flowchart symbol diagrams the operation of a conditional GO TO.



Since many of the following branching techniques change one or both of the numbers to be compared, it is assumed that they will be stored in memory registers when it is desired to save them.

Examples of Conditional GO TO

Conditions comparing L to 0 (zero).

CONDITION	STEPS TO PRODUCE BRANCH
IF L < 0 GO TO prog code	IF - GO TO prog code
IF L > 0 GO TO prog code	CHG SIGN IF - GO TO prog code
IF L = 0 GO TO prog code*	IF - GO TO 25 CHG SIGN IF - GO TO 25 GO TO prog code
← 25	(next step follows here)
IF L ≤ 0 GO TO prog code*	IF - GO TO prog code CHG SIGN IF - GO TO 25 GO TO prog code
← 25	(next step follows here)
IF L ≥ 0 GO TO prog code*	IF - GO TO 25 GO TO prog code
← 25	(next step follows here)
IF L ≠ 0 GO TO prog code	IF - GO TO prog code CHG SIGN IF - GO TO prog code

Conditions comparing L to U.

IF L < U GO TO prog code For example, U - L > 0	- ** CHG SIGN IF - GO TO prog code
IF L > U GO TO prog code For example, U - L < 0	- IF - GO TO prog code

*Program code 25 may be replaced throughout by any other program code.

**The subtraction key subtracts L from U and places the result in L.

CONDITION	STEPS TO PRODUCE BRANCH
IF L = U GO TO prog code* For example, U - L = 0	- IF - GO TO 10 CHG SIGN IF - GO TO 10 GO TO prog code 10 (next step follows here)
IF L ≤ U GO TO prog code* For example, U - L ≥ 0	- IF - GO TO 10 GO TO prog code 10 (next step follows here)
IF L ≥ U GO TO PROG code* For example, U - L ≤ 0	- IF - GO TO prog code CHG SIGN IF - GO TO 10 GO TO prog code 10 (next step follows here)
IF L ≠ U GO TO prog code* For example, U - L ≠ 0	- IF - GO TO prog code CHG SIGN IF - GO TO prog code

Looping

When a sequence of instructions is to be repeated, a loop may be used. A loop is formed using GO TO (or IF - GO TO) followed by the program code that marks the beginning of the key sequence to be repeated.

The following program illustrates the use of a loop.

```
10
    SPACE
     STOP
    PRINT
     X SQ
    PRINT
    GO TO 10
```

This is the loop. To terminate the loop, press RESET.

*Program code 10 may be replaced throughout by some other program code.

Complex Loops

The following diagrams illustrate some of the more complex loops that can be constructed using GO TO and IF - GO TO. Three dots in a vertical line stand for a set of programmed instructions.

- Two loops in one program:

```
10
      .
      .
      .
      GO TO 10
23
      .
      .
      .
      GO TO 23
```

- Transfer to loop on condition:

```
  12
      .
      .
      .
-----IF - GO TO 30
  20
      .
      .
      .
      GO TO 12
 --30
      .
      .
      .
      GO TO 20
```

This is the main loop.

This is the conditional loop.

- Loop within a loop:

```
        08  <-----------------+
         .                    |
         .                    |
         .                    |
        20  <-----------+     |
         .              |     |
         .              |     |
         .              |     |
 +------IF - GO TO 30   |     |
 |          GO TO 20 ---+     |
 +-->30                       |
         .                    |
         .                    |
         .                    |
            GO TO 08 ---------+
```

IND (00)

The INDIRECT key is used in two wasy:

- Following STORE, RECALL, and ACCSTORE, to use the content of register 00 as a memory register.

After STORE, RECALL or ACCSTORE, the IND key replaces the address of a memory register which normally follows these keys. When the 1155 encounters an IND after any of these keys, while executing a program, it uses the number in memory register 00 at that time for the address. When used as the address of a register, the number in 00 must be a number between 1 and 19 inclusive.

- Following GO TO and IF - GO TO, to use the content of register 00 as a program code.

After GO TO or IF - GO TO, the IND key replaces the program code which normally follows these keys. When the 1155 encounters an IND after either of these keys while executing a program, it uses the number in memory register 00 at that time for the program code. When used as a program code, the content of 00 must be a number between 0 and 99 inclusive, and must refer to an actual stored program code.

NOTES: If the content of 00 is not an integer, the 1155 will take that number rounded to the nearest integer as the memory address or program code; the number in 00 will not itself be changed by this operation.

In a program listing, the IND key appears as the letter N. For example, the instruction GO TO, IND is listed as "GO TO N."

STORE ( ) ( ) / IND (00)	Transfers the number in L to the memory register whose address is the number in register 00. Does not change the number in L; does not change the number in register 00.
RECALL ( ) ( ) / IND (00)	The number in register 00 is used to address another memory register, 01 through 19. The number in the register is transferred to the L register. The number previously in L is moved to U; the number previously in U is lost. The numbers in memory registers 00 through 19 are not changed.
ACCUM STORE ( ) ( ) / IND (00)	Adds the number in L to the number in the memory register given by register 00. The result is stored in the referenced memory register. Registers L, U and 00 are not changed.

KEYS	L	U	00	09	PRINTOUT
SET DEC 0					
CLR REGS	0	0	0	0	
9	09	0	0	0	
STORE 00	09	0	09	0	
10	10	09	09	0	
STORE N	10	09	09	10	
2	2	10	09	10	
PRINT	2	10	09	10	2
RECALL N	10	2	09	10	
PRINT	10	2	09	10	10
1	1	10	09	10	
ACCSTORE N	1	10	09	11	
RECALL N	11	1	09	11	
PRINT	11	1	09	11	11

GO TO ( ) ( ) / IND (00)	Uses the content of register 00 as a program code; executes a GO TO to the indicated program code, and continues running the program from the instruction following that program code.

EXAMPLES:

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

PROGRAM	COMMENTS
10	
SPACE	Key in N. N can be 1, 2, 3, or 10.
STOP	
PRINT	
STORE 00	Store N in register 00.
GO TO N	
01 ←	If N = 1, execution continues from program code 01.
1	
PRINT	
GO TO 10	
02 ←	If N = 2, execution continues from program code 02.
2	
2	
PRINT	
GO TO 10	
03 ←	If N = 3, execution continues from program code 03.
3	
3	
3	
PRINT	
GO TO 10	

Subroutine to compute

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

The value of x is assumed to be in U and the "return" program code in L. (See SUBROUTINES).

PROGRAM	COMMENTS
90	
STORE 00	Store the return address in 00
EXCG	Exchange L and U; X is now in L.
E TO X	
STORE 19	Compute e^{x} and store result in 19.
RECIP	Compute e^{-x}.
RECALL 19	
+	Compute $e^{x} + e^{-x}$
2	
÷	Divide by 2
GO TO N	Go to the return address.

Subroutines

If the same (nearly the same) sequence of instructions is used in several different places in a program, a programming technique called a subroutine may be used to shorten the program and to clarify its function.

For convenience in discussion, the program exclusive of subroutines is termed main program. GO TO instructions are used to reach ("call") the subroutine from the main program, and program codes are used to mark the points where the subroutine is to return to the main program.

To call a subroutine, use GO TO followed by the subroutine program code.

To return to the main program, store the program code marking the return point in register 00, and use GO TO N at the end of the subroutine.

In addition to the call and return program codes, it is frequently desired to store the values of variables to be manipulated by the subroutine, and recall them during the execution of the subroutine. The result of the subroutine operation may also be stored for use later on by the main program.

Sample Subroutine

Below is a sample subroutine to calculate

$$\cosh x = (e^{x} + e^{-x})/2$$

An instruction in the main program stores the value of x before calling the subroutine and another instruction stores the subroutine result (the computed value of cosh x) for use later in the main program.

PROGRAM		COMMENTS	
10			
	SPACE	Key in x	
	STOP		
	PRINT		
	STORE 01	Store x in 01	MAIN PROGRAM
	1	Enter return program code in L. (Go to sub-	
	1	routine.)	
	GO TO 90		
11		Program code for return from subroutine	
•			
•			
•			
90			
	STORE 00	Store return program code. Recall x.	
	RECALL 01		
	E TO X		
	STORE 19	Compute cosh x	
	RECIP		SUBROUTINE
	RECALL 19		
	+		
	2		
	÷		
	STORE 02	Store cosh x	
	GO TO N	Return to main program.	

Below is another subroutine to compute cosh x, using another way to "pass" the value of x to the subroutine.

PROGRAM		COMMENTS	
•			
•			
•			
10			
	STORE		
	STOP	Key in x	
	PRINT		
	1	Return program code.	MAIN
	1	Return program code	PROGRAM
	GO TO 90	is now in L and x is in U.	
11			
	•		
	•		
	•		
90			
	STORE 00	Store return program	
	EXCG	code in 00. Exchange	
	E TO X	L and U; x is now in L.	
	STORE 19		
	RECIP		SUBROUTINE
	RECALL 19		
	+		
	2		
	÷	cosh x is now in L.	
	GO TO N	Return to main program.	

IF NEG ( ) ( ) IND (00)

The IF - GO TO N instruction may be used to return from a subroutine which contains a loop; for example, an operation which is to be performed a specified number of times before the return to the main program is executed.

EXAMPLE: In the following example, the desired number of iterations is stored in Register 19; the return program code in in Register 00.

INSTRUCTIONS	REMARKS
25	Subroutine program code.
1 OPP SIGN ACCSTORE 19	Decrease the loop counter by 1.
EXCG RECALL 19 IF - GO TO N EXCG	Save the content of U (if desired). If the loop counter is negative, return to main program. Save U register (if desired).
. . . GO TO 25	Operation to be performed x times, $x \geq 0$.

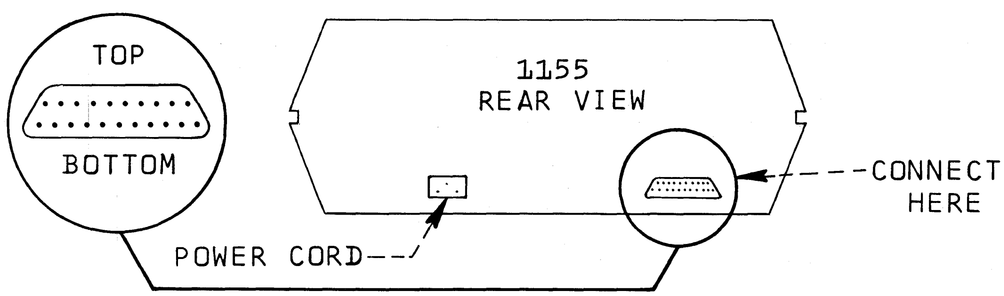
MAGNETIC CARD READER

The Model 511 magnetic card reader provides permanent external storage of programs written for the 1155.

To connect the card reader to the 1155 calculator:

1. BE SURE THAT THE POWER SWITCH ON THE 1155 IS OFF.

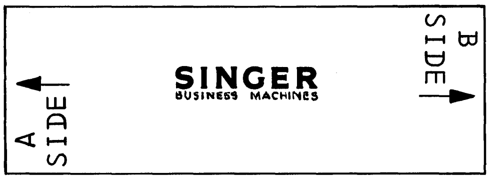
2. Connect the cord attached to the card reader to the socket in the rear of the 1155. Make sure the two rows of prongs on the cord attachment are aligned with the slots in the 1155; the shorter row of prongs should be on top.



3. Turn the 1155 power switch to ON. The card reader is now connected and ready to operate.

Magnetic Cards

Cards used with the Model 511 are plastic, and are magnetically coated (sensitive) on one face only. Each card will store or read instructions in two directions or "sides", labeled A and B. The card reader feeds the cards in one direction only; the arrows printed on the sensitive face of the card show which way the card should be inserted to read or store on the desired side (A or B).



Either side will store the maximum number of keystrokes (511 or FEWER) which the program memory can contain.

LOAD FROM CARD

Allows the user to enter instructions into the program memory from the magnetic card reader.

To load a program from a card:

1. Press RESET to reach keyboard mode.
2. Press LOAD FROM CARD; the keylight will GO ON.
3. Insert the card, printed face UP, into the slot in the front face of the card reader. This activates the drive mechanism in the reader, and the card feeds automatically.

Programs may be read in two directions or sides, labeled A and B; make sure that the card is inserted in the correct direction for the program you wish to load. When the reader has read the entire card (for example, one side), it stops automatically; the card may then be removed.

The instructions on the card are appended to any instructions already in the program memory. To erase any instructions from previous programs, press CLEAR PROG before pressing LOAD FROM CARD.

STORE IN CARD

Allows the user to store instructions in the program memory onto a magnetic card, using the magnetic card reader.

To store a program on a card:

1. Press RESET to reach keyboard mode.
2. Press STORE IN CARD; the keylight will GO ON.
3. Insert the card, printed face UP, into the slot in the front face of the card reader. This activates the drive mechanism in the card reader, and the card feeds automatically.

Programs may be stored on a card in two directions or sides, labeled A and B; make sure that the card is inserted in the correct direction. Any instructions already on the side of the card used for storing the program will be erased.

When the reader has fed the card entirely, storing all instructions in program memory on one side of the card (A or B), it stops automatically. The card may then be removed.

FRACTIONAL PART OF X

The fractional part of a number may be defined as:

$$y = FP(x) = x - \text{INTEGER } (x)$$

The following program computes and prints y after x is keyed in by the user.

PROGRAM	COMMENTS
10	
CLR REGS STOP PRINT	key in x
RECALL 99	Copy L into U
INTEGER - PRINT STOP	Compute and print y.

CUBE ROOT

To compute the cube root of the number in L, use

```
    3
RECIP
A TO X
```

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

To compute $c = \sqrt{a^2 + b^2}$, use

```
        a
    PRINT
        b
    PRINT
 TO POLAR
     EXCG
```

c is now in the L register.

SIGNUM FUNCTION

The signum function may be defined as:

$$y = SGN(x) = \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

The following program computes and prints y after the user keys in x.

PROGRAM	COMMENTS
10	
CLR REGS	
STOP	Key in x
PRINT	
IF - GO TO 20	
0	If $x > 0$, $y = x^0 = 1$
A TO X	If $x = 0$, $y = 0^0 = 0$*
GO TO 30	
20	
1	If $x < 0$, $y = -1$
OPP SIGN	
30	
PRINT	
STOP	

*The 1155 defines $0^0 = 0$.

MODULUS FUNCTION

The modulus function may be defined in the following way:

$$r = p \text{ MOD } q = p - q \text{ times INTEGER } (p \div q),$$

with the following restrictions:

$$p \geq 0 \qquad q > 0 \qquad 0 \leq r < q$$

Note that if p is not an integer, r will not be an integer. The quantity r is also called "the remainder of p divided by q." The following program computes and prints r after the user has keyed in p and q (in that order).

PROGRAM	COMMENTS
10	
CLR REGS	
STOP	Key in p and store it.
PRINT	
STORE 01	
STOP	Key in q and store it.
PRINT	
STORE 02	
÷	
INTGR	Compute r and print it.
RECALL 02	
X	
OPP SIGN	
RECALL 01	
+	
PRINT	
STOP	

INSERT MODE

INSERT

Puts the 1155 in insert mode. Instructions may be inserted into the program. The light above the INSERT key will come on.

To Insert Instructions into a Stored Program

1. List the program, one step at a time, until the place is reached where instructions are to be inserted.
2. Press RESET, then press INSERT.
3. Insert the new instructions (key them in).
4. Press RESET to terminate insert mode.

The above procedure will be applied to the following incorrectly stored program which is missing three instructions.

```
        INCORRECTLY STORED
             PROGRAM                      MISSING INSTRUCTIONS

10
        SPACE
         STOP  <------------------------- [PRINT
         STOP                             [STORE 01
        PRINT
            1
            +
        STORE 02
20
        RECALL 02
         SPACE
         PRINT <------------------------- STOP
        A TO X
        RECALL 01
            X
        PRINT
         GO TO 20
```

The missing instructions are inserted as follows:

1. List the program to the place where the first instruction is to be stored.

```
RESET, GO TO 10, LIST
                    ONE STEP          SPACE
                    ONE STEP           STOP
```

2. Insert PRINT and STORE 01 instructions.

```
RESET, INSERT, PRINT, STORE 01
```

3. List the program to the next place where an instruction is to be inserted.

```
RESET, GO TO 20, LIST
                    ONE STEP         RECALL
                    ONE STEP              0
                    ONE STEP              2
                    ONE STEP          SPACE
```

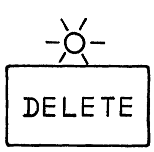
4. Insert STOP instruction.

```
RESET, INSERT, STOP
```

5. Press RESET to return to keyboard mode.

The modified program should be listed and proofread to make certain that the program is now correctly stored.

DELETE MODE



Puts the 1155 in delete mode and allows instructions to be deleted from a program one keystroke at a time, while the program is being entered or after it has been stored. DELETE is used with INSERT to edit stored programs.

To Delete an Instruction During Entry

If an error is discovered while a program is being entered (for example, in write mode):

1. Press RESET to terminate write mode.
2. Press DELETE to delete the last keystroke.
3. Press START.

NOTE: If DELETE has been pressed in error (for example, the step is not to be deleted), pressing RESET instead of START will return the 1155 to keyboard mode without deleting any steps.

EXAMPLE: The volume of a sphere. $V = 4/3\pi r^3$

Program:

```
          ++492
  10
         SPACE
          STOP
         PRINT
             3
        A TO X
            PI
             X
             4
             X
             3
             ÷
         PRINT
```

Enter the program:

```
RESET, CLEAR PROG, WRITE, PROG CODE, 1, 0,
SPACE, STOP, PRINT, 3, A TO X, 4
```

wrong key pressed

Delete the incorrect key:

```
RESET, DELETE, START
```

When DELETE is pressed, the light above it goes on; pressing START returns the 1155 to keyboard mode, and the delete light goes out. The incorrect instruction has been deleted. To continue storing the program, press:

```
WRITE, π, X, 4, X, 3, ÷, PRINT
```

To Delete Stored Instructions

1. List the program, one step at a time, up to and including the step to be deleted.

2. Press RESET, DELETE, START for each keystroke to be deleted.

EXAMPLE: Suppose that the user, in storing a program to compute

$$\frac{x}{x^2-1}$$

loses his place and inadvertently keys in some instructions twice. Then the listing of the program might be as follows:

```
            ++794 <---- checksum
27
           SPACE
            STOP
           PRINT
           STORE 01
            X SQ
           STORE 01 }
            X SQ    }  Here are the duplicated steps.
               1
               -
          RECALL 01
            EXCG
               ÷
           PRINT
           GO TO 27
```

To Delete the Duplicated Steps:

KEYS	PRINTOUT	
RESET, LIST, +,	++794	
ONE STEP	PROG CODE	
ONE STEP	2	
ONE STEP	7	
ONE STEP	SPACE	
ONE STEP	STOP	
ONE STEP	PRINT	
ONE STEP	STORE	duplicated steps
ONE STEP	0	
ONE STEP	1	
ONE STEP	X SQ	

There are four keystrokes to be deleted, as follows:

```
RESET
DELETE, START          (deletes X SQ)
DELETE, START          (deletes 1)
DELETE, START          (deletes 0)
DELETE, START          (deletes STORE)
```

NOTE: The DELETE key works backward from the last key reached by LIST, ONE STEP.

Now, LIST the program:

```
RESET, LIST, +, START
                   ++678        note altered checksum
         27
               SPACE
                STOP
               PRINT            duplicated keystrokes deleted
               STORE 01
                X SQ
                   1
                   -
              RECALL 01
                EXCG
                   ÷
               PRINT
               GO TO 27
```

To Edit Stored Programs

1. List the program in full.
2. Note errors: steps to be deleted and inserted.
3. Press RESET.
4. List the program one step at a time until the first error is reached. Use GO TO to begin the LIST, ONE STEP at a program code other than the beginning of the program.
5. DELETE and/or INSERT steps as desired.

EXAMPLE: Compute the mean of a set of scores, x.

$M = (\Sigma x_i)/N$, where N is the number of scores.

CORRECT PROGRAM:

```
              ++200
10
         CLR REGS
            SPACE
13
             STOP
            PRINT
       ACCSTORE 01
                1
       ACCSTORE 02
          GO TO 13
17
            SPACE
        RECALL 01
            PRINT
        RECALL 02
            PRINT
                ÷
            PRINT
             STOP
          GO TO 10
```

Suppose, however, that several incorrect steps had been inadvertently stored, so that the listing is as follows:

```
10
         CLR REGS
            SPACE
13
             STOP
            PRINT
         ACCSTORE 10 <——— incorrect step
                1
         ACCSTORE 02
            GO TO 13
17
            SPACE
          RECALL 01
          RECALL 02 <——— step omitted
            PRINT
```

To edit the program:

KEYS	PRINTOUT
RESET, GO TO 13, LIST	
ONE STEP	STOP
ONE STEP	PRINT
ONE STEP	ACCSTORE
ONE STEP	1
ONE STEP	0
RESET, DELETE, START	(DELETE 0)
DELETE, START	(DELETE 1)
INSERT, 0, 1	(INSERT 0, 1)
RESET, GO TO 17, LIST	
ONE STEP	SPACE
ONE STEP	RECALL
ONE STEP	0
ONE STEP	1
RESET, INSERT, PRINT	(INSERT PRINT)

Now, list the edited program:

RESET, LIST, +, START

CHECKSUMS

A number called the checksum is used to identify each program. To obtain the checksum in a program listing, press

RESET, LIST, +,

The checksum will be printed, preceded by the symbols "++". Modifying the program usually changes the checksum. It is possible, but unlikely, that two different programs will have the same checksum.

The checksum is used to verify that a program has been correctly loaded, either by hand or using the Model 511 Magnetic Card Reader. If the checksums of the original program listing and of the program loaded are not the same; it is not the same program or the card has been loaded or stored incorrectly.

If the checksums of the original program listing and the program just loaded do not match follow the procedure below.

1. List the first few steps of the program using LIST, ONE STEP, to see if they match the original program. Be sure to list several steps since many programs begin in similar ways.

2. If the program has been loaded from a card, check to see that the cable connecting the 1155 to the card reader is attached properly, reload the card, and print the checksum.

3. Using the GO TO key, jump to a few program codes selected at random and list the program in their vicinity as an additional check on the identify of the loaded program.

4. List the entire program and check it against the original listing. If the program was loaded manually from the keyboard look for missing or extra steps as well as incorrect ones.

5. If the listings match, run the program as a test for malfunctions. If the program runs properly, the original checksum may be in error.

SUMMARY

KEYS	L	U	ERROR CODES
+	U + L	0	ERROR 7 IF $\lvert U + L \rvert > 10^{100}$
−	U − L	0	ERROR 7 IF $\lvert U - L \rvert > 10^{100}$
X	U X L	0	ERROR 7 IF $\lvert U \times L \rvert > 10^{100}$
÷	U ÷ L	0	ERROR 5 IF L = 0
$\lvert A \rvert^{X}$	U^{L}	0	ERROR 1 IF $\lvert U^{L} \rvert > 10^{100}$
X^2	L^2	U	ERROR 7 IF $\lvert L \rvert > 10^{50}$
$\sqrt{\lvert X \rvert}$	$\sqrt{\lvert L \rvert}$	U	
1/X	1/L	U	ERROR 8 IF L = 0
INTGR	Integer part of L	U	
$\lvert X \rvert !$ *	$\lvert L \rvert !$	U	ERROR 7 IF L $\geq$ 69.5 ERROR 6 IF L $\geq$ 100
e^{X}	e^{L}	U	ERROR 1 IF L $\geq$ 231
$\ln \lvert X \rvert$	$\log_e \lvert L \rvert$	U	ERROR 3 IF L = 0

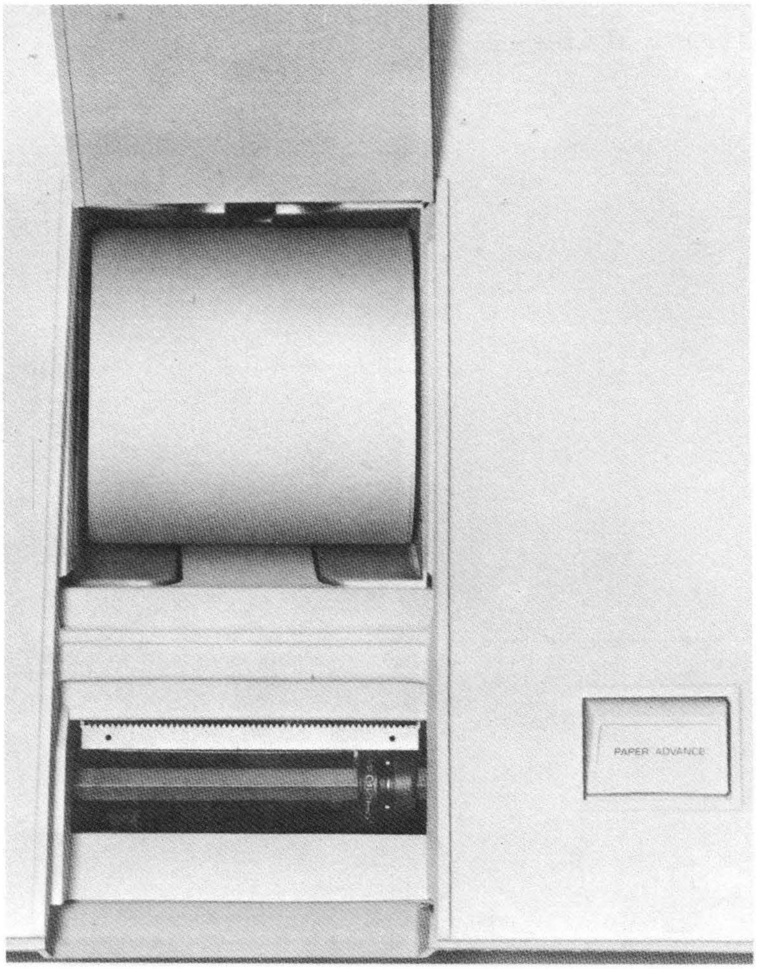
*The number in L is rounded before $\lvert X \rvert !$ is computed.

KEYS	L	U	ERROR CODES
log|X|	$\log_{10}$ L	U	ERROR 3 IF L = 0
DEG	L	U	
RAD	L	U	
SIN	SIN L	U	
COS	COS L	U	
TAN	TAN L	U	ERROR 2 IF L IS ODD MULTIPLE OF 90°
ARC SIN*	SIN^{-1} L	U	ERROR 4 IF |L| > 1
ARC COS*	COS^{-1} L	U	ERROR 4 IF |L| > 1
ARC TAN*	TAN^{-1} L	U	
TO POLAR	TAN^{-1}(U/L)	$\sqrt{L^2 + U^2}$	ERROR 10 IF U = L = 0
TO RECT	U COS L	U SIN L	

*The inverse functions compute the principal values:

$$-90^\circ \le \sin^{-1} L \le 90^\circ$$
$$0^\circ \le \cos^{-1} L \le 180^\circ$$
$$-90^\circ \le \tan^{-1} L \le 90^\circ$$

LOADING PAPER TAPE



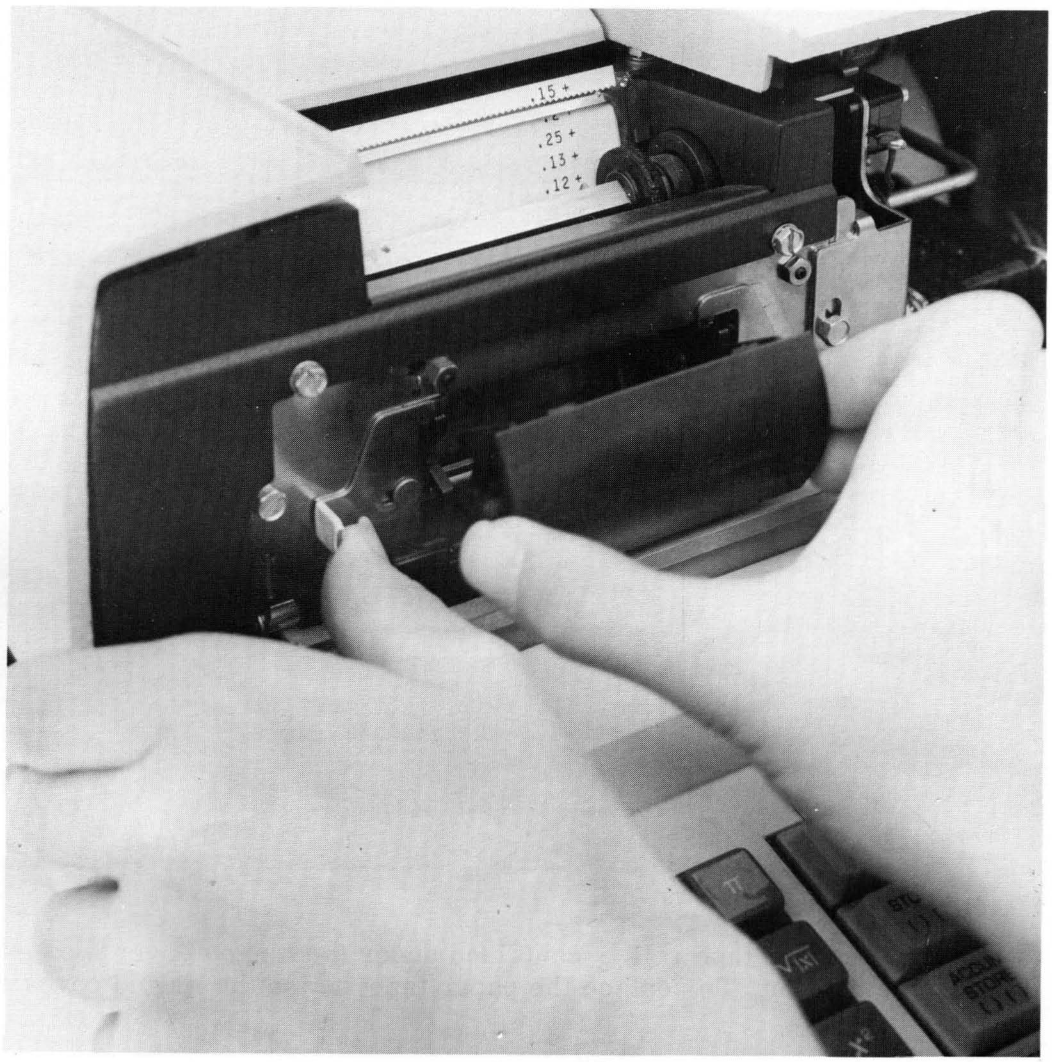
The paper tape roll is contained under the hinged cover located on top of the calculator. To replace the paper tape, raise the hinged cover and proceed as follows:

1. Remove the remaining roll by lifting the roll from the well and tearing the tape to free the spool.
2. Depress PAPER ADVANCE until the remaining paper is free.
3. Tear leading edge of new roll to form a smooth, flat edge.
4. Drop the roll in the well, leading edge facing down.

5. Depress PAPER ADVANCE key until paper appears above the paper tear off.

6. After closing the cover, the machine is ready to operate.

CHANGING INK ROLLER



The 1155 contains a unique ink roller that eliminates ribbons. Designed for maximum convenience, each ink roller lasts much longer than a ribbon, and eliminates ribbon threading and stained hands.

Simply replace the old ink cartridge with a new one according to the following procedure. The new cartridge is conveniently packaged in a plastic container so you never have to touch the ink.

1. Remove the snap-on plate on the front of the machine.

2. Push Locking Tab (yellow) to left and hold.

3. Grasp ink cartridge with right hand and extract.

4. Release Locking Tab (yellow). Calculator is now ready to receive new ink cartridge.

5. Hold Locking Tab (yellow) to left. Grasp new ink cartridge by ends. Insert new cartridge, checking to see that both ends of the inker shaft are inserted in the yokes.

6. Release Locking Tab (yellow). Make sure ink cartridge is locked in place.

7. Replace the snap-on front plate.

FRIDEN MODEL 1155 ADVANCED PROGRAMMABLE CALCULATOR
REFERENCE MANUAL
Publication No. 55-538

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.

- 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)

DETACH HERE

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

SINGER
BUSINESS MACHINES

Publication No. 55-538