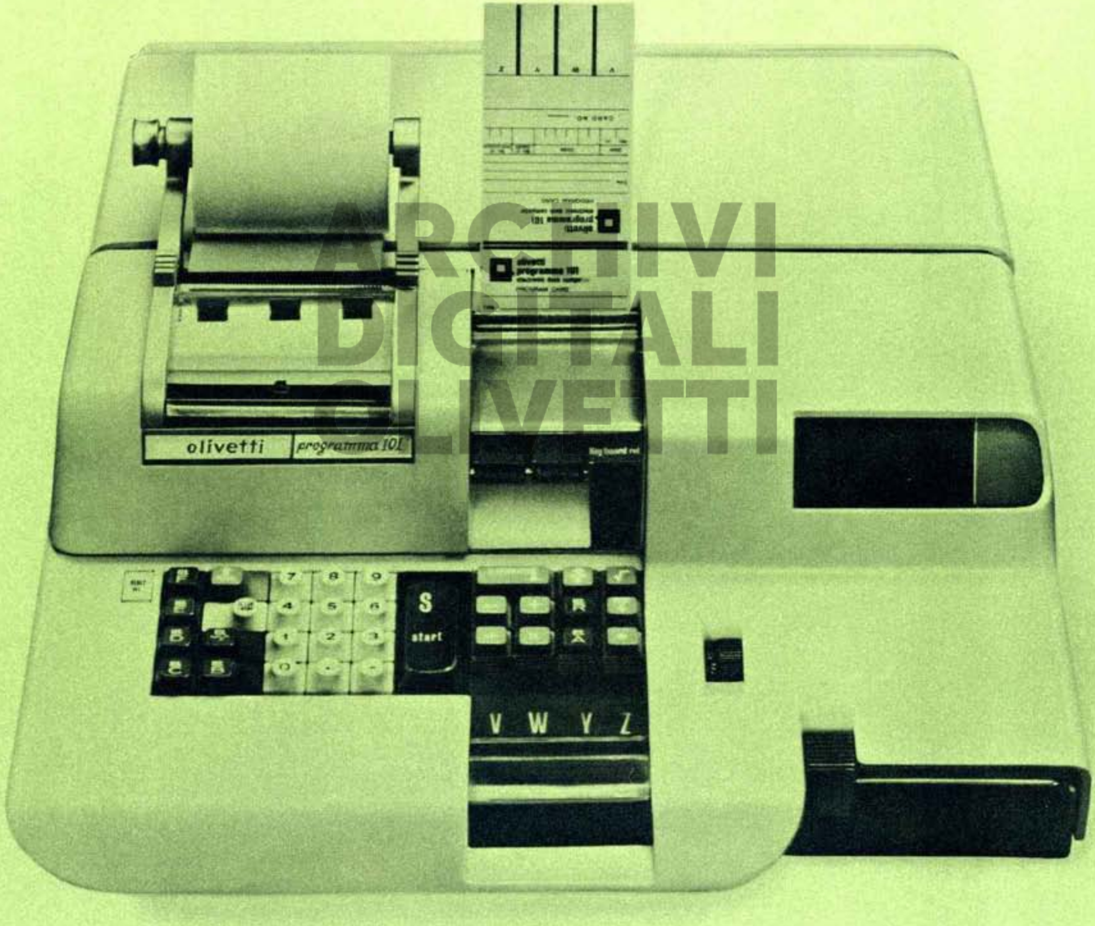
olivetti

PROGRAMMA 101

GENERAL REFERENCE MANUAL

Programma 101
general
reference
manual


olivetti
programma 101
S
start
V W Y Z

Preface

This manual presents the operation, concept and capabilities of the Programma 101. It is divided into a description of the computer, an explanation of its program language and brief presentation of procedures and techniques in programming and using the computer.

Contents

Page

Computer Components 6

Memory 7

Keyboard 11

Arithmetic Operations 15

Addition 16

Subtraction 17

Multiplication 18

Division 19

Square Root 20

Absolute Value 20

Transfer Operations 21

From M 21

To A 22

Exchange 23

D-R Exchange 24

Decimal Part to M 24

General Operations 25

Stop 25

Clear 25

Print 25

Vertical Spacing 25

Jump Operations 26

Unconditional Jumps 27

Conditional Jumps 28

Constants 29

Constants in Registers 29

Constants as Instructions 29

Split Register 30

Decimal Wheel 31

Computer Utilization 32

To Record a Program 32

To Read and Record D and E 33

To Print a Program 33

To Use a Program 34

Automatic Internal Checks 34

Manual Mode 35

Computer Exercise 36

Changing The Ribbon 38

Insertion of Paper Roll 39

Computer Components

The Programma 101 is composed of the following elements:

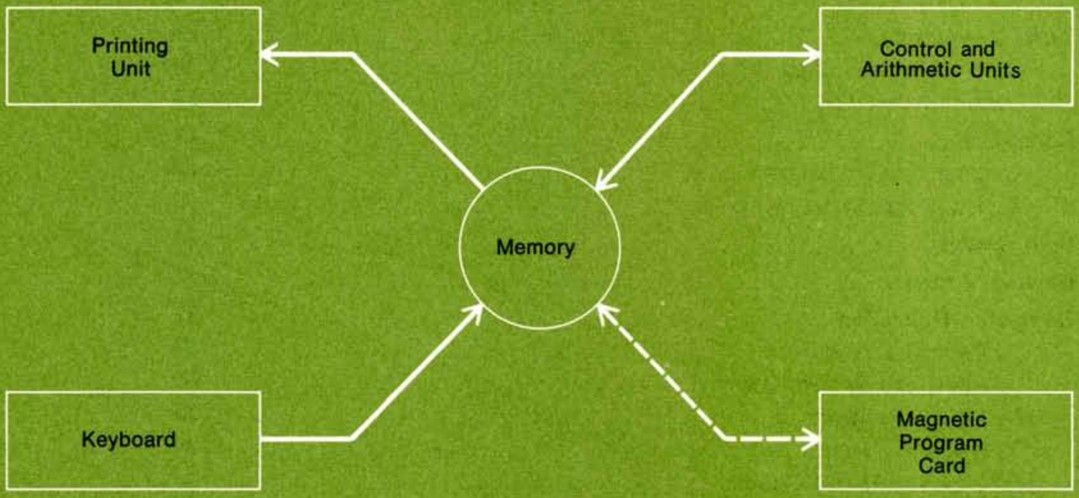
Memory: This unit stores numeric data and program instructions.

Electric Keyboard: For numeric entry, manual operation, and compiling program instructions.

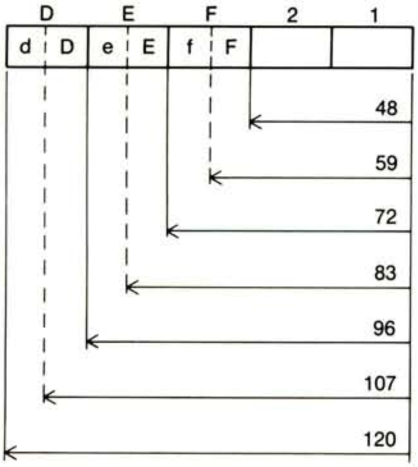
Printing Unit: Serial printing, from right to left, at 30 characters per second; this unit prints all keyboard entries, programmed output and instructions.

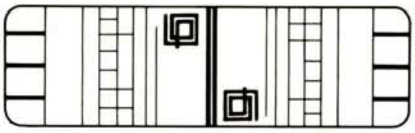
Magnetic Card Reader/Recorder: A device through which instructions for a program can be stored on, and retrieved from, a magnetic card.

Control and Arithmetic Units: The *Control Unit* is the administrative section of the computer. It receives the incoming information and decides the computation by telling the *Arithmetic Unit* what to compute and where to find the information.



Memory





The memory is composed of ten registers. Eight are storage registers and two are used exclusively for instructions.

The two instruction registers can store a total of 48 instructions.

The eight storage registers, M, A, R, B, C, D, E and F, have a capacity of 22 digits, plus a decimal point and sign.

Three of these registers, M, A and R, are operating registers and take part in all arithmetic operations.

The M register *is the Median or distributive register. ALL keyboard figure entries are held in the M register and distributed to the other registers as instructed.*

The A register *functions with the arithmetic unit to form the Accumulator. Arithmetic results are developed and retained in the A register. A result of up to 23 digits can be achieved in the A register.*

The R register retains:

- *The complete results in addition and subtraction.*
- *The complete product in multiplication.*
- *The remainder in division.*
- *A nonfunctional remainder in square root.*

The five remaining registers, B, C, D, E, and F, are storage registers. Each can be split into two registers with a capacity of 11 digits, plus decimal point and sign.

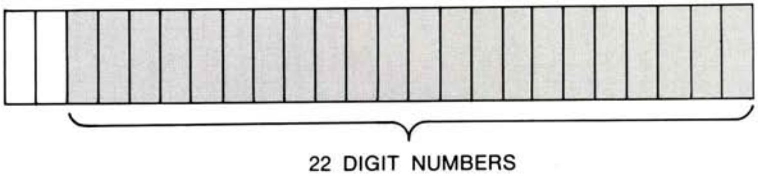
When storage registers are split, the right portion of the split register retains its original designation, while the left side is identified with the corresponding lower case letter. The lower case designation is obtained by entering the corresponding upper case letter and depressing the "/" key, e.g. c = C/.

The registers F, E and D and their splits have the additional capability of storing instructions and constants.

When these registers or their splits are used as instruction registers, the instructions follow an overflow pattern, so that after the instruction registers are at capacity, the remaining instructions will be received first by F, then f, then E, then e, then D, and finally d. Programs of up to 120 instructions can be stored. This concept is represented above. When registers D, E and F and their splits are not used for instructions, they may be used to contain constants or as regular storage registers.

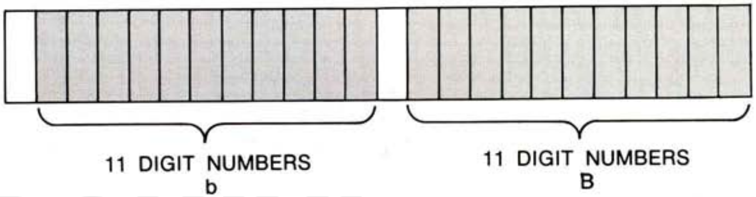
The following formats show the separation and capacities of the ten registers.

1



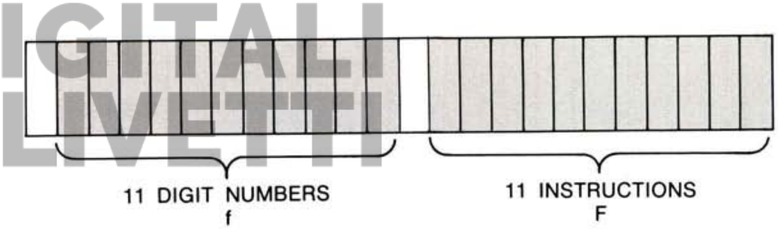
Format 1 – This represents the storage registers used as complete registers: A, M, R: always complete; B, C, D, E, F: complete in this case.

2



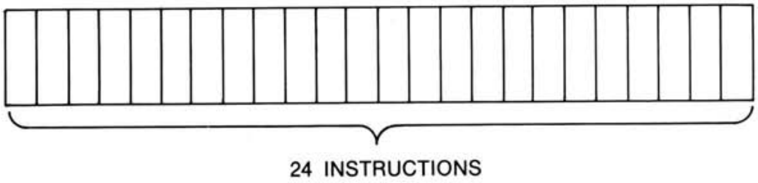
Format 2 – This represents the storage registers when split: b, B; c, C; d, D; e, E; and f, F.

3

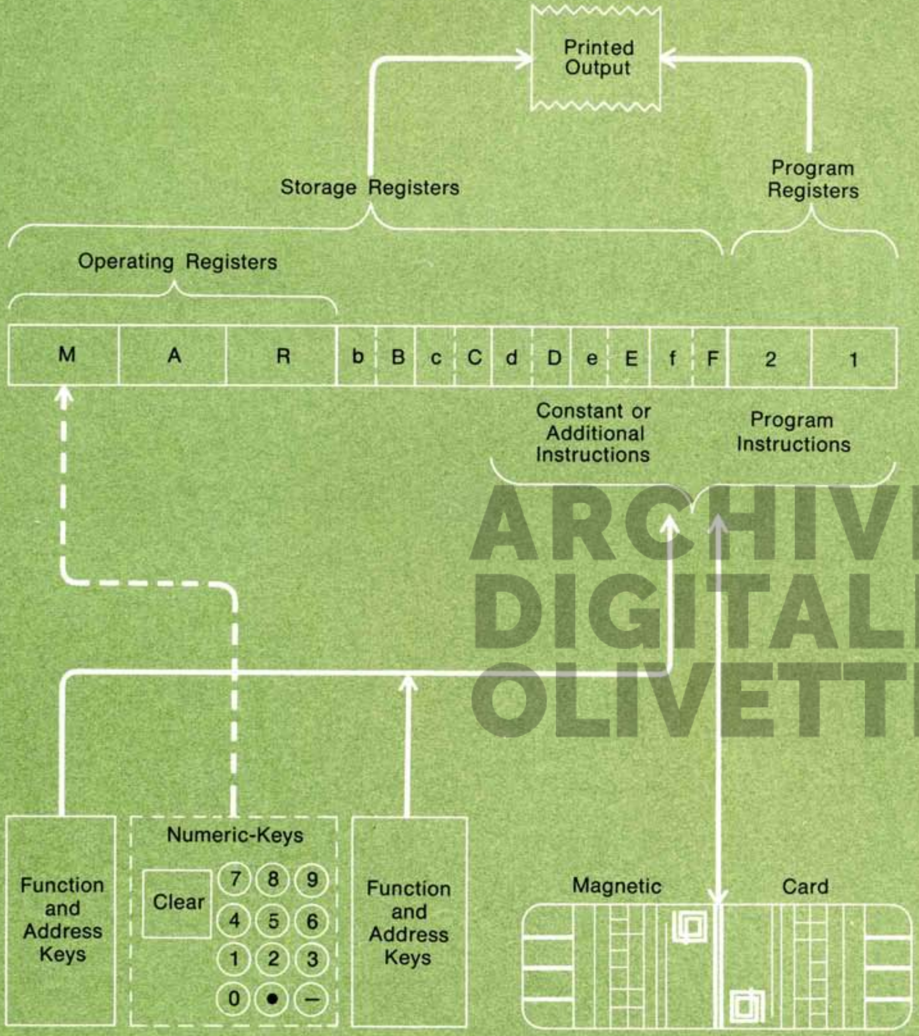


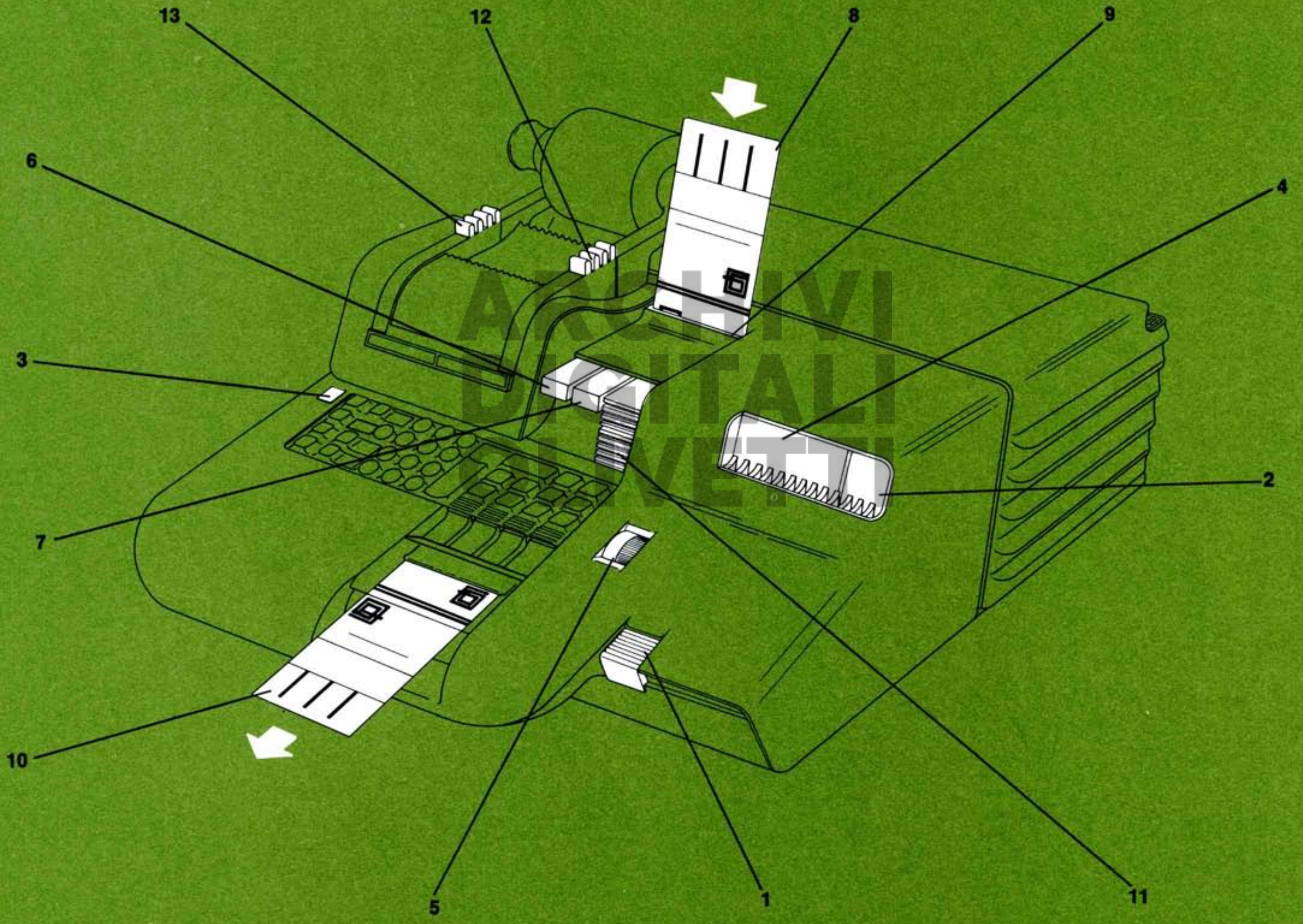
Format 3 – This represents the split registers with numeric and instruction storage: e.g. d, D; e, E; and f, F.

4



Format 4 – This represents the complete instruction registers. Registers 1 and 2; registers F, E and D only when used for instructions.


Printed Output
Storage Registers
Program Registers
Operating Registers
M
A
R
b
B
c
C
d
D
e
E
f
F
2
1
Constant or Additional Instructions
Program Instructions
Function and Address Keys
Numeric-Keys
Clear
7
8
9
4
5
6
1
2
3
0
●
–
Function and Address Keys
Magnetic
Card


13
12
8
9
6
4
3
2
7
10
5
1
11

Keyboard

1 **The ON-OFF** Key is a dual purpose switch for both the ON and OFF positions. (Note: the OFF position automatically clears all stored data and instructions.)

2 **The ERROR LIGHT** lights when the computer is turned on, and whenever the computer detects an operational error; e.g. exceeding capacity, division by zero.

3 **The GENERAL RESET KEY** erases all data and instructions from the computer.

4 **The CORRECT PERFORMANCE LIGHT** indicates the computer is functioning properly. The light is steady when the computer is ready for operation; a flickering light indicates the computer is performing programmed instructions.

5 **The DECIMAL WHEEL** determines the number of decimal places to which computations will be carried out in the A register, and the decimal places in the printed output, except for results from the R register.

6 **The RECORD PROGRAM SWITCH,** when **ON** (in), directs the computer to store instructions either in memory from the keyboard, or onto a magnetic program card from memory.

The RECORD PROGRAM SWITCH must be OFF (out) to load instructions from a magnetic program card into the memory.

7 **The PRINT PROGRAM SWITCH,** when **ON** (in), directs the computer to print out the instructions stored in memory from its present location in the program to the next Stop instruction (S), whenever the Print key (20) is depressed.

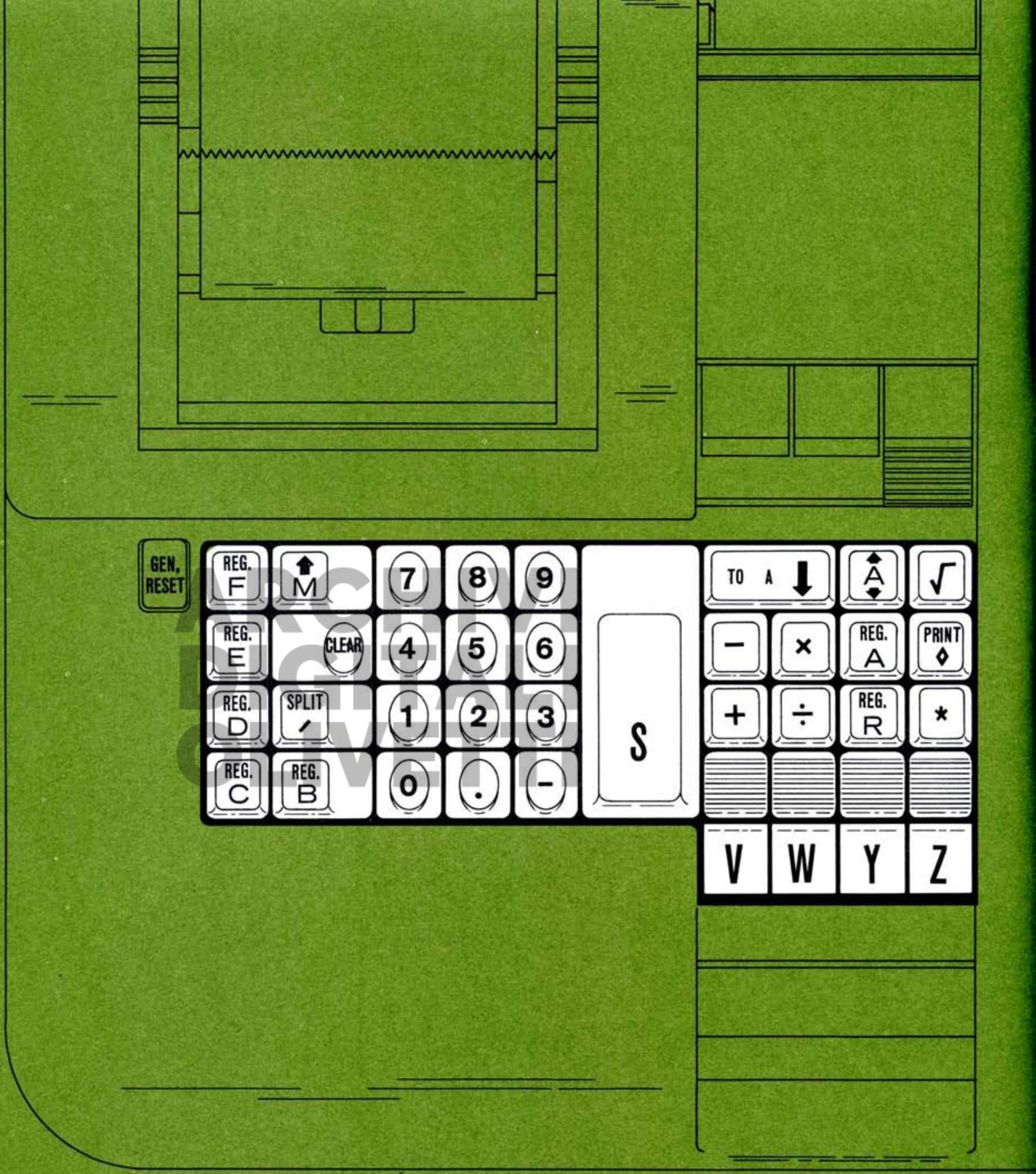
8 **The MAGNETIC PROGRAM CARD** is a plastic card, with a ferrous oxide backing, used to magnetically record programs for external storage. The card is inserted into a read/write device **(9)** to record instructions and/or constants to or from the computer memory. Once inserted, the card may be removed from the computer **(10)** without disturbing the stored instructions.

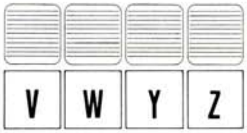
(Note: the read/write device uses only half the magnetic card; consequently, two sets of 120 instructions and/or constants may be stored on a single card.)

11 **The KEYBOARD RELEASE KEY** reactivates a locked keyboard. If two or more keys are depressed simultaneously, the keyboard will lock to indicate a misoperation. Because the operator does not know what entry was accepted by the computer, after touching the Keyboard Release Key the Clear Entry Key (16) MUST also be depressed and the complete figure entered.

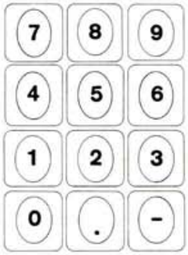
12 **TAPE ADVANCE:** for advancing the paper tape.

13 **TAPE RELEASE LEVER** enables precise finger-tip adjustment when changing tape rolls.


GEN. RESET
REG. F
M
7
8
9
TO A
A
√
REG. E
CLEAR
4
5
6
−
×
REG. A
PRINT ◊
REG. D
SPLIT /
1
2
3
S
+
÷
REG. R
*
REG. C
REG. B
0
.
−
V
W
Y
Z



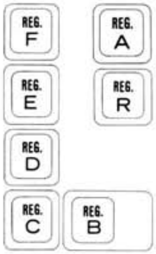
14 ROUTINE SELECTION KEYS: V, W, Y and Z activate the computer into the proper program or subroutine.



15 NUMERIC KEYBOARD: Ten-key entry system with provision for entry of a decimal point and a negative value. Keyboard figures are immediately stored in the M register at the time of entry.

16 CLEAR ENTRY KEY: clears the entire keyboard entry.

17 START/STOP KEY: Reactivates the computer into the programmed sequence and is used to code a stop instruction when entering program instructions.

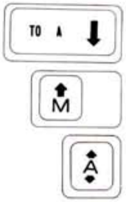


18 REGISTER KEYS: Storage registers B, C, D, E, F and the arithmetic registers A and R are identified by their corresponding letters. The operating register M has no keyboard identification.

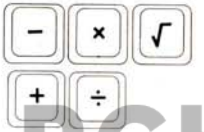
19 SPLIT KEY: The designation for a split register is obtained by entering the register key and depressing the split key; e.g. C/ = c. When storage registers are split, the right portion of the split register retains the original designation while the left side is identified on the tape with the corresponding lower case letter.

20 PRINT KEY: Prints the contents of a selected register.

21 CLEAR KEY: Clears the contents of a selected register. This key will also cause printing when operated manually.



22 TRANSFER KEYS: These keys perform transfer operation between the storage registers and the operating registers.



23 ARITHMETIC KEYS: These keys perform their indicated arithmetic function.

Arithmetic Operations

All arithmetic operations are performed in the operating registers M, A and R. An arithmetic operation is performed in two phases:

1 The contents of the selected register are automatically transferred to the M register. The M register is selected automatically if no other register is indicated.

2 The operation is carried out in the M, A and R registers.

The Programma 101 can perform these arithmetic operations: +, −, ×, ÷, √, and absolute value. Programma accepts and computes figures algebraically. A negative value is entered on the keyboard with the figure entry; the negative key can be touched at any time during entry of a figure. If there is no negative indication, the computer will accept the figure as a positive value.

The Subtract operation is separate from the numeric keyboard and is used exclusively for Subtraction.

These operations are described in detail in the following pages.

Addition

An instruction containing the operation "+" directs the computer to:

1ST PHASE *Transfer the contents of the selected register to M, while retaining them in the original register.*

2ND PHASE *Add the contents of M to the contents of A, obtaining in A the sum truncated according to the setting of the Decimal Wheel. The complete sum is in R. M retains its contents.*

			RESULTS	
INSTRUCTION	**PHASE 1**	**PHASE 2**	**TRUNCATED**	**COMPLETE**
+		A + M	A	R
R +	R → M	A + M	A	R
A +	A → M	A + M	A	R
b +	b → M	A + M	A	R
B +	B → M	A + M	A	R
c +	c → M	A + M	A	R
C +	C → M	A + M	A	R
d +	d → M	A + M	A	R
D +	D → M	A + M	A	R
e +	e → M	A + M	A	R
E +	E → M	A + M	A	R
f +	f → M	A + M	A	R
F +	F → M	A + M	A	R

EXAMPLE:

	M	A	R	B
BEFORE	5	7	9	3
B +				
AFTER	3	10	10	3

Subtraction

An instruction containing the operation "–" directs the computer to:

1ST PHASE — *Transfer the contents of the selected register to M, while retaining them in the original register.*

2ND PHASE — *Subtract the contents of M from the contents of A, obtaining in A the difference truncated according to the setting of the Decimal Wheel. The complete difference is in R. M retains its contents.*

			RESULTS	
INSTRUCTION	PHASE 1	PHASE 2	TRUNCATED	COMPLETE
–		A – M	A	R
R –	R → M	A – M	A	R
A –	A → M	A – M	A	R
b –	b → M	A – M	A	R
B –	B → M	A – M	A	R
c –	c → M	A – M	A	R
C –	C → M	A – M	A	R
d –	d → M	A – M	A	R
D –	D → M	A – M	A	R
e –	e → M	A – M	A	R
E –	E → M	A – M	A	R
f –	f → M	A – M	A	R
F –	F → M	A – M	A	R

EXAMPLE:

	M	A	R	B
BEFORE	5	7	9	3
B –				
AFTER	3	4	4	3

×

Multiplication

An instruction containing the operation "×" directs the computer to:

1ST PHASE	*Transfer the contents of the selected register to M, while retaining them in the original register.*
2ND PHASE	*Multiply the contents of M by the contents of A, obtaining in A the product truncated according to the setting of the Decimal Wheel. The complete product is in R. M retains its contents.*

			RESULTS	
INSTRUCTION	**PHASE 1**	**PHASE 2**	**TRUNCATED**	**COMPLETE**
×		A × M	A	R
R ×	R → M	A × M	A	R
A ×	A → M	A × M	A	R
b ×	b → M	A × M	A	R
B ×	B → M	A × M	A	R
c ×	c → M	A × M	A	R
C ×	C → M	A × M	A	R
d ×	d → M	A × M	A	R
D ×	D → M	A × M	A	R
e ×	e → M	A × M	A	R
E ×	E → M	A × M	A	R
f ×	f → M	A × M	A	R
F ×	F → M	A × M	A	R

EXAMPLE:

	M	A	R	B
BEFORE	5	7	9	3
B ×				
AFTER	3	21	21	3

Division

An instruction containing the operation "÷" directs the computer to:

1ST PHASE *Transfer the contents of the selected register to M, while retaining them in the original register.*

2ND PHASE *Divide the contents of M into the contents of A, obtaining in A the quotient truncated according to the setting of the Decimal Wheel. The decimally correct fractional remainder is in R. M retains its contents.*

INSTRUCTION	PHASE 1	PHASE 2	QUOTIENT	REMAINDER
÷		A ÷ M	A	R
R ÷	R → M	A ÷ M	A	R
A ÷	A → M	A ÷ M	A	R
b ÷	b → M	A ÷ M	A	R
B ÷	B → M	A ÷ M	A	R
c ÷	c → M	A ÷ M	A	R
C ÷	C → M	A ÷ M	A	R
d ÷	d → M	A ÷ M	A	R
D ÷	D → M	A ÷ M	A	R
e ÷	e → M	A ÷ M	A	R
E ÷	E → M	A ÷ M	A	R
f ÷	f → M	A ÷ M	A	R
F ÷	F → M	A ÷ M	A	R

EXAMPLE:

	M	A	R	B
BEFORE	5	7	9	3
B ÷				
AFTER	3	2	1	3

Square Root

An instruction containing the operation "√ " directs the computer to:

1ST PHASE	*Transfer the contents of the selected register to M, while retaining them in the original register.*
2ND PHASE	*Extract the square root of the contents of M, as an absolute value, obtaining in A the result truncated according to the setting of the Decimal Wheel. The R register contains a nonfunctional remainder. At the end of the operation, M contains double the square root.*

INSTRUCTION	PHASE 1	PHASE 2	TRUNCATED RESULT
√		√ M → A	A
R √	R → M	√ M → A	A
A √	A → M	√ M → A	A
b √	b → M	√ M → A	A
B √	B → M	√ M → A	A
c √	c → M	√ M → A	A
C √	C → M	√ M → A	A
d √	d → M	√ M → A	A
D √	D → M	√ M → A	A
e √	e → M	√ M → A	A
E √	E → M	√ M → A	A
f √	f → M	√ M → A	A
F √	F → M	√ M → A	A

EXAMPLE:

	M	A	R	B
BEFORE	5	7	7	9
B √				
AFTER	6	3	–	9

Absolute Value

The absolute value instruction "A↕" changes the contents of the A register, if negative, to positive.

EXAMPLE:

	M	A	R	B
BEFORE	3	–6	4	8
A↕				
AFTER	3	6	4	8

From M

Transfers the contents of M to the selected register while retaining them in M.

INSTRUCTION	RESULT
↑	Inoperative
R ↑	Inoperative
A ↑	See Constants as Instructions
b ↑	M → b
B ↑	M → B
c ↑	M → c
C ↑	M → C
d ↑	M → d
D ↑	M → D
e ↑	M → e
E ↑	M → E
f ↑	M → f
F ↑	M → F

EXAMPLE:

	M	A	R	B
BEFORE	19	5	16	30
B ↑				
AFTER	19	5	16	19

To A

Transfer the contents of the selected register to A while retaining them in the original register. The contents of M are not affected.

INSTRUCTION	RESULTS
↓	M→A
R ↓	R→A
A ↓	Inoperative
b ↓	b→A
B ↓	B→A
c ↓	c→A
C ↓	C→A
d ↓	d→A
D ↓	D→A
e ↓	e→A
E ↓	E→A
f ↓	f→A
F ↓	F→A

EXAMPLE:

	M	A	R	B
BEFORE	24	36	48	12
R ↓				
AFTER	24	48	48	12

Exchange

Exchange the contents of the A register with the contents of the selected register. The contents of M are not affected except by the exchange between A and M

INSTRUCTION	RESULT
↕	M ⇌ A
R ↕	operates as R ↓ (R → A)
A ↕	Absolute Value
b ↕	b ⇌ A
B ↕	B ⇌ A
c ↕	c ⇌ A
C ↕	C ⇌ A
d ↕	d ⇌ A
D ↕	D ⇌ A
e ↕	e ⇌ A
E ↕	E ⇌ A
f ↕	f ⇌ A
F ↕	F ⇌ A

EXAMPLE:

	M	A	R	B
BEFORE	6	9	12	15
B ↕				
AFTER	6	15	12	9

RS

D-R Exchange

Exchanges the contents of D (both D and d registers) with the contents of R register

This instruction has a special use in the following cases exclusively:

- In multicard programs, to temporarily store the contents of the D register in R, when a new card has to be read to continue the program. During this temporary storage no instruction with an R address can be executed.
- To use the D register for data storage in the second phase of a program when, in the first phase, this register has been used to store instructions which are no longer needed. In this case the RS instruction must be preceded by R ★ (usually inoperative).

INSTRUCTION	RESULT
RS	d, D ⇄ R

EXAMPLE:

	d	D	R
BEFORE	2.25	45.50	135798624211345
RS			
AFTER	135798624211345		2.25 45.50
BEFORE	135798624211345		2.25 45.50
RS			
AFTER	2.25	45.50	135798624211345

/↕

Decimal Part To M

Transfers the decimal portion of the A register contents to the M register while retaining the original contents in A.

INSTRUCTION	RESULT
/↕	DECIMAL PART OF A → M

EXAMPLE:

	M	A	R	B
BEFORE	2.6	5.231	20	18
/↕				
AFTER	0.231	5.231	20	18

General Operations

Stop

Stops the program and releases the keyboard for the entry of figures or the selection of a subroutine. After figure entry, the program is restarted by touching the Start key (S).

The program can also be restarted by touching a Routine Selection key.

When the "S" instruction stops the program, the computer may also be operated in the manual mode.

S

Clear

As a programmed instruction "★" clears the selected register. The M and R registers can not be cleared with this instruction.

✱

Print

Instructs the computer to print the contents of the selected register while retaining them in the register.

◇

Vertical Spacing

Is used to advance the tape one vertical space, without printing.

/◇

Jump Operations

The Jump operation directs the computer to depart from the normal sequence of step-by-step instructions and jump to a pre-selected point in the program.

These instructions provide both internal and external (manual) decision capability and are useful to:

Create "loops" that allow repetitive sequences in a program to be executed.

Select alternate routines or subroutines at the discretion of the operator.

Select alternate routines or subroutines according to the value of an obtained result.

The jump operation requires two related instructions:

1 Point of origin: where to start the jump, interrupting the sequence.

2 Reference point: where the jump will arrive, restarting the sequence.

There are two types of jump instructions:

Unconditional jumps

Conditional jumps

Unconditional Jumps

these jumps are executed whenever the instruction is read

List of Unconditional Jump Instructions:

	POINT OF ORIGIN	REFERENCE POINT
normally used to initiate program or to select a Subroutine manually.	V	AV
	W	AW
	Y	AY
	Z	AZ
	CV	BV
	CW	BW
	CY	BY
	CZ	BZ
	DV	EV
	DW	EW
	DY	EY
	DZ	EZ
	RV	FV
	RW	FW
	RY	FY
	RZ	FZ

An abbreviated form of list is to combine V, W, Y or Z as follows:

△ → A △
C △ → B △
D △ → E △
R △ → F △

△ = V, W, Y or Z

All programs normally begin with the reference point of an unconditional jump instruction. Reference Points AV, AW, AY, AZ are used so the program sequence can be started simply by touching Routine Selection key V, W, Y or Z.

△ → A△
C△ → B△
D△ → E△
R△ → F△

/△ ⟶ a△
c△ ⟶ b△
d△ ⟶ e△
r△ ⟶ f△

Conditional Jumps

these jumps choose one of two alternatives by testing the contents of the A register for the following condition:

If the contents of the A register are:

GREATER THAN 0 – the program jumps to the corresponding Reference Point.

ZERO OR LESS – the program continues with the next instruction in sequence.

The Conditional Jump can not directly differentiate between a negative value and zero. If the A register has a negative value, the distinction may be made by a two-test procedure: 1. The Value of A is tested. If the jump occurs, this value is positive; if the jump does not occur, the A ↕ (absolute value) instruction is executed. 2. The value of A is again tested. If the jump occurs, the value was negative whereas if the jump does not occur, the value is zero. If necessary, the program will re-establish the initial algebraic condition of A.

List of Conditional Jump Instructions:

POINT OF ORIGIN	REFERENCE POINT
/V	a V
/W	a W
/Y	a Y
/Z	a Z
c V	b V
c W	b W
c Y	b Y
c Z	b Z
d V	e V
d W	e W
d Y	e Y
d Z	e Z
r V	f V
r W	f W
r Y	f Y
r Z	f Z

The lower case letters shown in the above lists are obtained by entering the corresponding upper case letter and touching the "/" key, e.g. r = R/.

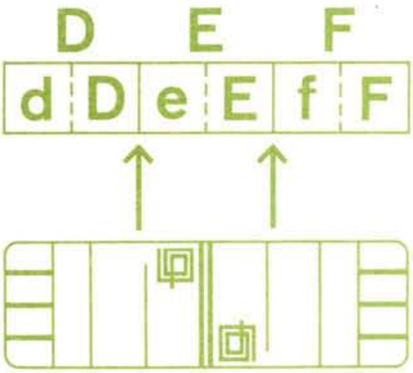
An abbreviated form of the above chart is created by combining V, W, Y, Z with the following:

/ △ → a △
c △ → b △
d △ → e △
r △ → f △

△ = V, W, Y or Z.

Constants

Programma 101 provides the capability to store on the magnetic card, in addition to program instructions, numeric values (e.g. constants) to be used within the program.



Constants in Registers

Numeric values entered into the registers D, E and F, or their splits, can be directly stored together with the program on the magnetic card when the RECORD PROGRAM switch is ON (in). When the card is read, the same values are retrieved from the card and recorded into the original registers.

This feature provides the capability of storing on the card:

- up to 3 values of 22 digits each; or
- up to 6 values of 11 digits each (or any combination thereof).

When Program instructions overflow from F towards D obviously values can be stored only in those registers, or their splits, not utilized by instructions. This method is primarily useful for values with many digits and/or repeatedly utilized throughout the program.

A/↑

Constants as Instructions

This is a method for storing values in the program by converting the required number into computer instructions, one for each digit.

These instructions are preceded by the instruction a↑ (A/↑) and are entered as part of the program.

When the computer carries out this special series of instructions, the value is generated in the M register.

Before recording the program, the programmer generates the instruction for the required value in the following manner:

1 Turn PRINT PROGRAM switch ON (in).
2 Turn RECORD PROGRAM switch OFF (out).
3 Touch GENERAL RESET.
4 Enter the value to be generated.
5 Touch the Keys A ↑.
6 Touch PRINT (◇).
Repeat steps 3 through 6 for each value.

The computer will print the sequence of instructions that corresponds to the entered value. Then the programmer enters these instructions as a part of the program, following the special instruction a↑ (A/↑); the special instruction must precede each value.

This method is primarily useful for values with few digits and/or utilized only once within a program routine.

Split Register

As was pointed out in the preceding pages, each of the B, C, D, E and F registers can be split into two parts, each with a capacity of 11 digits plus decimal point and sign. The left part of the register is identified with the lower case letter corresponding to its companion's designation. The lower case designation is obtained by entering the corresponding upper case letter and touching the "/" key, e.g. c = C/. The right side of the split register retains its original upper case letter designation.

The computer has built-in self-regulating circuitry that alerts the operator if an attempt has been made to transfer a value larger than 11 digits to a split register or to split a register that already contains a value greater than 11 digits.

Nevertheless, a register can be used as both a whole and a split register at varying times throughout a program by using the clear (★) instruction before shifting from whole to split or vice versa. For example:

In phase one of a program, register B is used as a split register (b,B); then in phase two, register B is to be used as a whole register:

1 At the end of phase 1, a b★ instruction frees the left side of the register so that, in phase 2, it can be used in conjunction with the right side as a whole register.

2 At the end of phase 2, a B★ instruction frees the whole register so that it can be used as a split register when the program starts again with phase 1.

Decimal Wheel

As it affects COMPUTATION:

The Decimal Wheel determines the number of decimal places to which the result in the A register will be carried out. The Decimal Wheel has the following effect on these operations.

ADDITION, SUBTRACTION, MULTIPLICATION: *After the computation, the result in the A register is truncated according to the number of decimal places indicated by the setting of the Decimal Wheel. The complete result is retained in the R register.*

DIVISION: *The quotient is retained in the A register and is carried out only to the number of decimal places indicated by the setting of the Decimal Wheel.*

The decimally correct remainder is retained in the R register.

SQUARE ROOT: *The root is retained in the A register and is extracted to the number of decimal places indicated by the setting of the Decimal Wheel.*
The R register contains a non-functional remainder.

As it affects KEYBOARD ENTRIES

An additional feature of the ten key entry system is the Decimal Point key. When entering decimal numbers, the Decimal Point key is touched in its proper position; e.g. to enter 12.6; you should enter 1, then 2, then touch the Decimal Point key, and finally enter 6.

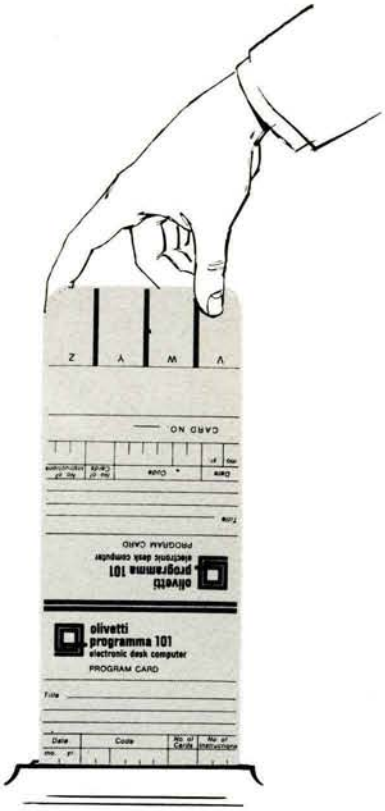
To enter numbers less than 1, a zero must be entered before the decimal point; e.g. .07 would be entered as 0.07.

Regardless of the setting of the Decimal Wheel, the complete figure entered on the keyboard will be printed when the Start key or an operation key is touched.

As it affects OUTPUT PRINTING:

All printed output, except that of the R register, is truncated to the setting of the Decimal Wheel.

Computer Utilization



To Record a Program

Once the sequence of steps to solve a problem has been developed on coding sheets, the operator enters the program into the computer by depressing the keys corresponding to the instruction sequence.

1 Turn computer ON

2 Touch GENERAL RESET

3 Turn RECORD PROGRAM switch ON (in).

4 Enter, in order, the instructions that comprise the program. See example: Touch A, Touch V, etc.

5 The computer will print when the complete instruction has been entered.

6 To correct an error, the CLEAR ENTRY key is used. Touching the CLEAR ENTRY key will clear ONLY:
a. the last instruction printed; or
b. that portion of an instruction entered on the keyboard.

7 After writing the program, turn RECORD PROGRAM switch OFF (out).

8 If constants are to be used by the program, enter the constant figure and transfer it to the appropriate register.

EXAMPLE: to store 89.14 in d register, enter 8 9 . 1 4 on the keyboard in order; touch keys D / ↑ in succession. The constant 89.14 is now stored in d register.

9 The computer is now ready to use the completed program.

10 To store the program and constants permanently on a magnetic card turn RECORD PROGRAM switch ON (in) and insert the magnetic card into the computer.

11 Turn RECORD PROGRAM switch OFF (out).

PROGRAM FORM

NO	INSTRUCTIONS			M
		A	V	
		C	✱	
			S	
	B	/	↑	
			S	
		B	↑	
	B	/	↓	
		B	+	
		A	◊	
		C	↕	
	B	/	↓	
		B	−	
		A	◊	
		C	+	
		C	↕	
	B	/	↓	
		B	×	
		A	◊	
		C	+	
		C	↕	
	B	/	↓	
		B	÷	
		A	◊	
		C	+	
		C	↕	
	B	/	√	
		A	◊	
		C	+	
		C	↕	
		C	◊	
			V	

Read/Record D and E

Constants or instructions can be recorded only from Registers D and E onto a card, or from a card only into Registers D and E, without affecting the contents of other program registers:

1 If the information is recorded in the D and E registers:
BOTH the Record Program switch and the Print Program switch are ON.

A program card is inserted.

The transfer of information is from the D and E registers of the memory to the program card.

2 If the contents of a card must be stored only in the D and E registers:

The Print Program switch is turned ON and the program card is inserted. Information is read into the D and E registers exclusively.

Subroutines or module programs can be utilized effectively by this technique.

To Print a Program

Once a program has been stored in the computer, the entire sequence of instructions forming the program can be printed out. This is normally used to analyze a new program to insure correct entry, or to locate an error in the program.

1 To locate the initial instruction of a program, turn RECORD PROGRAM ON (in) momentarily, then OFF (out).

2 Turn PRINT PROGRAM ON (in); touch the Print key (◇).

3 The computer will start printing the instructions in their programmed sequence.

4 The computer will stop at each Stop instruction (S). To continue the printing of the instructions, touch the Print key (◇).

5 When the entire program has been printed, turn PRINT PROGRAM OFF (out).

To locate an error within a program, after the Error light has gone on, follow steps 2 through 5 above. This prints the instructions from the present location in the program. The instruction immediately preceding the first printed instruction caused the activation of the Error light.

To Use a Program

1 Turn computer ON.

2 Touch GENERAL RESET.

3 Turn RECORD PROGRAM and PRINT PROGRAM switches OFF (out).

4 Set DECIMAL WHEEL to desired position.

5 Insert the program card into the computer. The program card can be removed from the computer at any time, since the program is transferred directly to the Memory. If the program is to be repeated, the program card need not be reinserted since the program is already in the computer.

6 Depress the proper ROUTINE SELECTION key.

7 Operations will occur as explained in the operating procedure. The keyboard will lock during computations. When variable factors are to be entered, the program will stop and the keyboard will be unlocked to permit data entry.

Automatic Internal Checks

The Programma 101 provides a system of automatic internal checks to assure the operator that all components are functioning properly. The red ERROR light is the visual indication that a misoperation has been detected.

When a magnetic program card is misread, the error indicated is usually either a damaged or a dirty card, or the improper insertion of the card.

examine the physical condition of the card and if there is no evidence of damage, reinsert the card properly.

The following misoperations are indicated by the Error light:

- Capacity exceeded by computation.
  The original contents of registers A and R are destroyed.
- Capacity of a split register exceeded by a transfer.
  The transfer is not performed.
- Division by zero.
  The original contents of registers A and R are destroyed.
- Keyboard entry capacity exceeded.
  To repeat the entry correctly, the clear entry key must first be depressed.

the following steps provide the information basic to the analysis of the above errors:

- *print out all registers*
- *print out program from point of error (the instruction immediately preceding the first printed instruction is the operation that caused the error).*
- *print out the complete program*

Programma 101 can be operated manually. The rules given in the preceding pages for computer instructions apply also to manual operations with the following exceptions:

1 The results of multiplication, division and square root will be printed automatically.

2 The results of addition and subtraction will be printed only by touching the keys A and ◇ in sequence.

3 The ★ key when touched will also cause printing, clearing the selected register (except M and R).

The computer can be operated manually also when it is used in program mode, at any stop instruction in the program. In this second case, the operator should be careful when entering figures in the registers not to affect the contents of the registers used by the program.

Computer Exercise

The following exercise shows you how to:

- use the Programma manually
- write a program

The problem used is:

a + b = ans.
a − b = ans.
a × b = ans.
a ÷ b = ans.
√a = ans.
grand total (GT)

Assume a = 9 and b = 3;
The following operations solve the problem manually:

	Programma prints
Turn the computer ON.	
Depress the GENERAL RESET key.	
Turn the RECORD PROGRAM switch OFF (out).	
Turn the PRINT PROGRAM switch OFF (out).	
Set DECIMAL WHEEL to desired number.	
Enter 1st factor and Transfer to B/.	9 b ↑
Enter 2nd factor and Transfer to B.	3 B ↑
Move 9 (B/) to A register.	b ↓
Add 3 (B) to 9 (A).	B +
Print answer in A.	12 A ◇
Transfer answer to GT (C).	C ↕
Move 9 (B/) to A register.	b ↓
Subtract 3 (B) from 9 (A).	B −
Print answer in A.	6 A ◇
Add answer (A) and GT (C).	C +
Store new GT in C.	C ↕
Move 9 (B/) to A register.	b ↓
Multiply 3 (B) × 9 (A).	B ×
Answer prints automatically.	27 A ◇
Add answer (A) and GT (C).	C +
Store GT in C.	C ↕
Move 9 (B/) to A register.	b ↓
Divide 3 (B) into 9 (A).	B ÷
Answer prints automatically.	3 A ◇
Add answer (A) and GT (C).	C +
Store new GT in C.	C ↕
Take square root of 9 (B/).	b √
Answer prints automatically.	3 A ◇
Add answer (A) and GT (C).	C +
Store new GT in C.	C ↕
Print GT.	51 C ◇

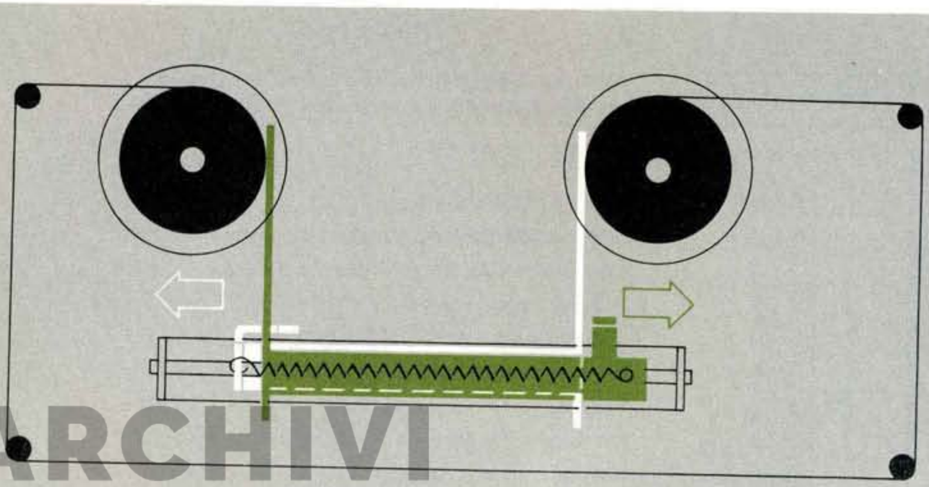
The following instructions solve the same problem in program mode:

PROGRAM FORM

NO	INSTRUCTIONS			M	
		A	V		Starting Point of Program.
		C	✱		Clear register for accumulation of answers.
			S		Stop program for entry of factor a.
	B	/	↑		Transfer factor a to B /.
			S		Stop program for entry of factor b.
		B	↑		Transfer factor b to B.
	B	/	↓		Transfer factor a to A register.
		B	+		Add factor B to factor a in A register.
		A	◇		Print answer in A register.
		C	↕		Transfer answer to C register (GT).
	B	/	↓		Transfer factor a to A register.
		B	−		Subtract factor b from factor a in A register.
		A	◇		Print answer in A register.
		C	+		Add answer and GT (C register).
		C	↕		Transfer new GT to C register.
	B	/	↓		Transfer factor a to A register.
		B	×		Multiply factor b X a.
		A	◇		Print answer in A register.
		C	+		Add answer and GT (C register).
		C	↕		Transfer new GT to C register.
	B	/	↓		Transfer factor a to A register.
		B	÷		Divide factor b into factor a.
		A	◇		Print answer in A register.
		C	+		Add answer and GT (C register).
		C	↕		Transfer new GT to C register.
	B	/	√		Take square root of factor a.
		A	◇		Print answer in A register.
		C	+		Add answer and GT (C register).
		C	↕		Transfer new GT to C register.
		C	◇		Print Final GT in C register.
			V		Jump to Start of Program.

Note that the manual solution to the problem and the programmed solution are quite similar, with exceptions only for the instructions to create the loop, and the entry of variables. Consequently, a useful technique for testing and debugging a new program is the step-by-step manual operation of the written sequence.

Changing The Ribbon



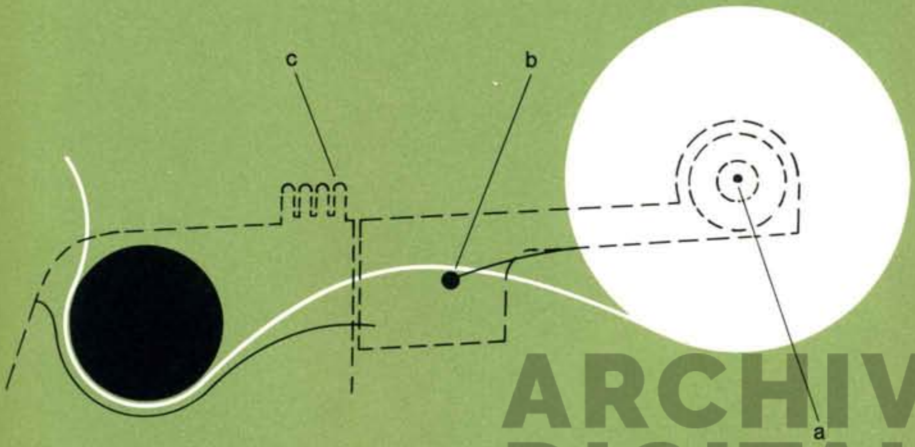
1 Remove print unit cover and pull paper tape holder forward.

2 Move the ribbon reverse levers away from the spools (see diagram).

3 Unwind the ribbon completely from one of the spools.

4 Fasten the end of the new ribbon to the empty spool and wind it a few turns.

5 Replace the spools and thread the ribbon as indicated in the diagram. Restore the ribbon Reverse levers.



1 Remove the paper roll bar (a) and discard the old spool.

2 Place the new roll between the two holders, so the tape feeds from underneath, and insert paper roll bar.

3 Turn Programma on.

4 Feed the tape over the. guide bar (b) and under the print cylinder. Depress the tape advance lever (c) to feed tape thru the printing unit.

5 Depress tape release lever to adjust tape alignment.

ING. C. OLIVETTI & C., S.p.A.—IVREA, ITALY

PRINTED IN U.S.A.