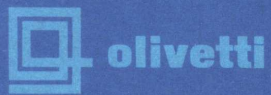


Manuale di programmazione

# Programma 101

Calcolatore elettronico da tavolo



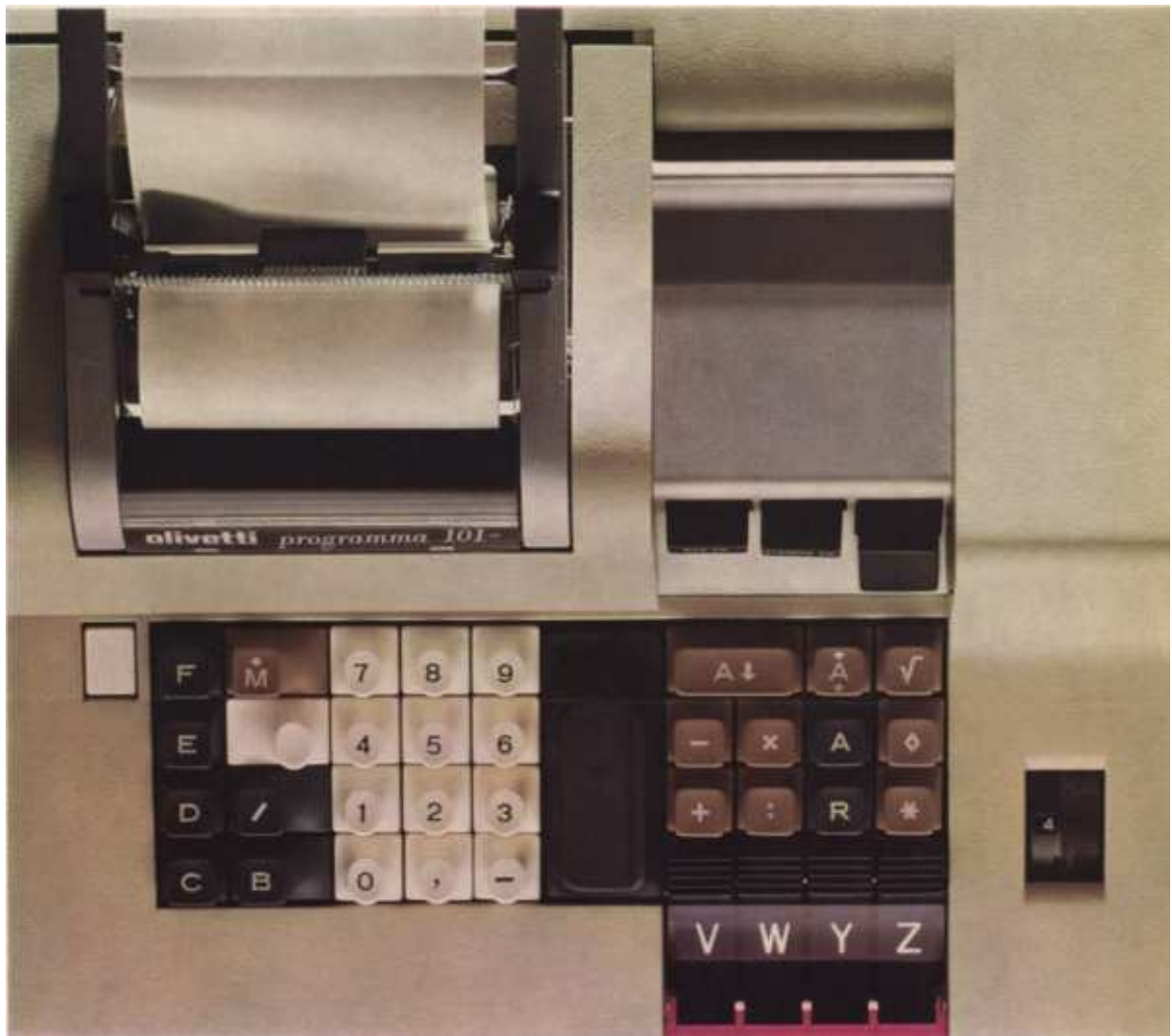
# Manuale di programmazione

## Programma 101

Questo manuale è dedicato alla programmazione del calcolatore Olivetti Programma 101, all'illustrazione delle istruzioni e delle tecniche che è necessario conoscere per programmare la soluzione di un dato problema.

Tutto ciò che si riferisce all'aspetto operativo del calcolatore esula dalle finalità del presente volume ed è trattato nel Manuale Generale. Poiché il calcolatore Olivetti Programma 101, pur essendo di piccole dimensioni, ha le caratteristiche dei calcolatori elettronici numerici, all'inizio del manuale è sommariamente illustrata l'organizzazione di un calcolatore e la funzione del programma.

L'ultima parte contiene consigli utili circa il modo con cui si deve affrontare e risolvere un problema mediante l'uso del calcolatore.



From the original Italian Manual - (L.C. Sept 2017)

## Indice

Struttura di un calcolatore elettronico	9
Organi di entrata e di uscita	9
Memoria	10
Unità aritmetico-logica	11
Unità di governo	11
Caratteristiche del calcolatore Programma 101	12
Organi di entrata	12
Organi uscita	12
Memoria	12
Unità di governo e unità aritmetico-logica	14
Programmazione	14
Le funzioni aritmetiche	16
Operazioni aritmetiche elementari	16
Capacità dei registri operativi	17
Operazioni con dati contenuti nei registri di deposito	17
Uso dei registri divisi nelle operazioni aritmetiche	18
Radice quadrata	19
Operazioni con dati contenuti nei registri operativi	20
Introduzione dei dati e funzioni di trasferimento	23
Posizionamento dei dati nei registri	23
Introduzione dei dati	23
Funzioni di trasferimento	24
Le funzioni di servizio	29
Istruzione di stampa	29
Istruzione di azzeramento	29
Interlinea	30
Le funzioni logiche	31
Principi generali	31
Salti incondizionati	31
Salti condizionati	34
Dati costanti	38
Costanti nei registri di deposito	38
Codificazione di costanti come istruzioni	38
Tecniche di programmazione	40
Inversione di segno di un dato	40
Reciproco di un dato	40
Generazione di costanti	40
Arrotondamento all'unità più vicina	41
Arrotondamento per eccesso	42
Arrotondamento a valori prestabiliti	43
Programmazione di un contatore	43
Interruzione automatica di una sequenza ciclica	44
Registrazione di più dati in unico registro	44
Uso promiscuo dei registri F, E, D, per dati e istruzioni	45
Lunghezza dei dati e decimali nei risultati	47

Programmi particolari	49
Programma su più schede	49
Istruzione RS	49
Utilizzazione del tasto S per l'avvio di un programma	50
Utilizzazione dei sottoprogrammi	50
Il calcolo manuale	52
Differenza tra calcolo manuale e calcolo programmato	52
Utilizzazione del calcolo manuale	52
Norme operative	53
Registrazione dei programmi	53
Stampa dei programmi	53
Lettura ed esecuzione dei programmi	54
Registrazione e lettura di scheda parziale	54
Tempi macchina	55
Analisi di un problema e preparazione del programma	56
Analisi numerica	56
Analisi logica	56
Programmazione	58
Messa a punto e controllo di un programma	59
Codificazione manuale delle costanti	60
Tabella riassuntiva delle funzioni	61
Indice analitico	63

## Struttura di un calcolatore elettronico

L'organizzazione di un calcolatore elettronico numerico può essere sinteticamente illustrata con uno schema semplificato di validità generale, indipendentemente dalle caratteristiche dei singoli tipi.

Il calcolatore elettronico consente elaborazioni di dati ad elevata velocità. I calcolatori possono essere suddivisi in grandi, medi o piccoli, a seconda del volume di dati che possono elaborare in una unità di tempo.

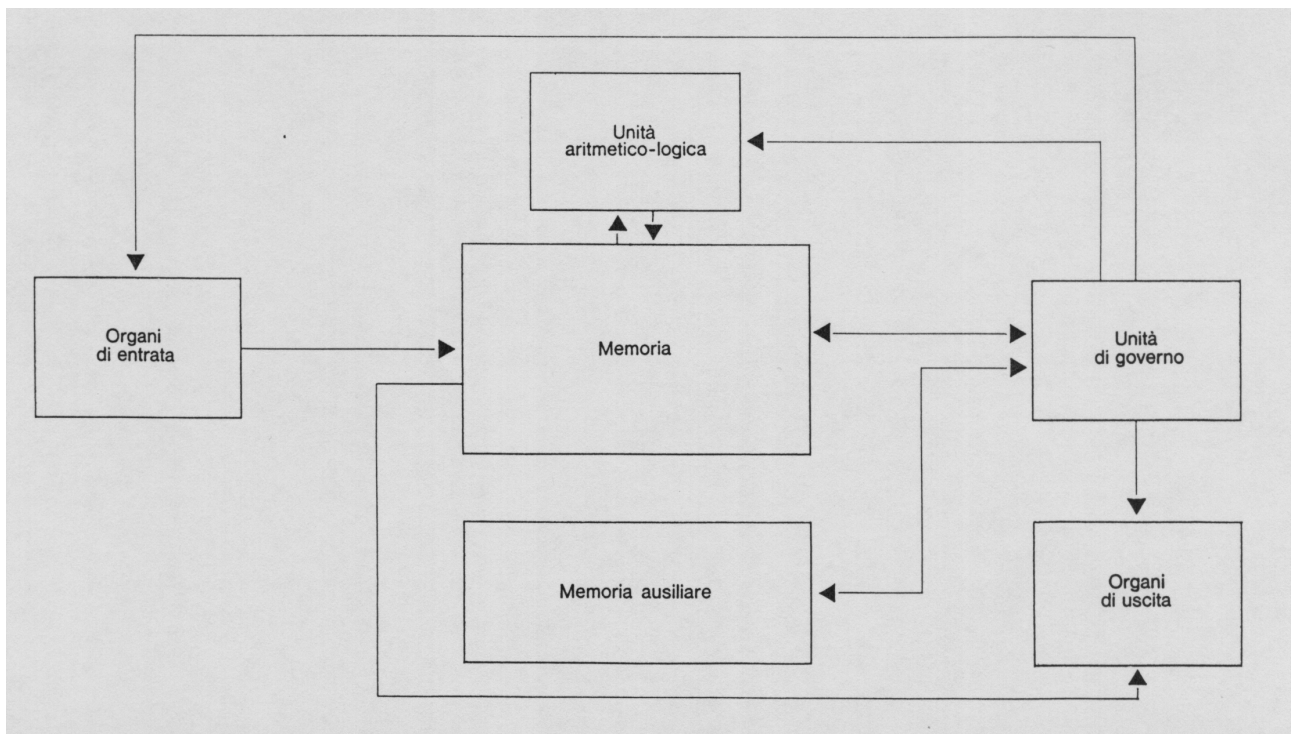
Gli organi di un calcolatore possono essere classificati nel modo seguente:

- *organi di entrata e di uscita*
- *memoria*
- *unità aritmetico-logica*
- *unità di governo*

### Organi di entrata e di uscita

Gli organi di entrata permettono di trasferire dati e comandi dall'esterno all'interno del calcolatore. Sono cioè un mezzo di comunicazione tra l'uomo e la macchina; possono essere di vario genere, dalla semplice tastiera di tipo dattilografico ai più perfezionati lettori di schede, di nastro perforato, di nastro magnetico, ecc. A essi è affidato il compito di trasformare i dati provenienti dall'esterno in impulsi elettrici.

Gli organi di uscita hanno una funzione opposta e simmetrica; costituiscono cioè il mezzo di comunicazione tra la macchina e l'uomo, trasferendo dati o risultati dall'interno all'esterno del calcolatore mediante trasformazione di impulsi elettrici in stampa, in perforazioni su scheda o nastro, o in registrazioni magnetiche.



(pag. 10)

### Memoria

La memoria può essere considerata la parte centrale di un calcolatore, registra ciò che viene comunicato dall'esterno per mezzo degli organi di entrata e lo restituisce opportunamente el-



borato tramite gli organi di uscita. Ogni dato, risultato o comando deve passare attraverso la memoria.

La memoria è qualificata in funzione della capacità e del tempo di accesso.

*La capacità* indica quanti dati sono registrabili in memoria. E' espressa in varie unità di misura, a seconda del tipo di calcolatore e dell'organizzazione della memoria.

*Il tempo di accesso* indica il tempo intercorrente tra il momento in cui si richiede un dato contenuto in memoria ed il momento in cui si può disporne. Spesso uno stesso calcolatore, specie se si di medie o di grandi dimensioni, è dotato di diversi tipi di memoria.

Le memorie a tempo di accesso rapido (i nuclei magnetici, thin film, ecc.) sono caratterizzate da un tempo di acceso dell'ordine dei microsecondi o dei nanosecondi, ma hanno un costo elevato. I dati in esse contenuti possono essere direttamente sfruttati dall'unità aritmetico-logica.

Le memorie a tempo di accesso medio o basso (tamburi magnetici, dischi, nastri, ecc.) sono caratterizzate da un tempo di accesso molto più elevato, ma hanno un costo notevolmente minore. In generale i dati in esse contenuti non possono essere sfruttati direttamente dall'unità aritmetico-logica, ma devono essere trasportati preventivamente nella memoria ad accesso rapido che ha la funzione di memoria principale del calcolatore e che interviene nell'elaborazione vera e propria. Le altre memorie hanno una funzione ausiliaria e sono utilizzate per l'immagazzinamento di una considerevole quantità di dati.

I calcolatori a programma registrato consentono la registrazione in memoria, sotto forma codificata, del programma relativo all'elaborazione che si vuole eseguire. L'unità di governo, ovvero gli organi di comando, prelevano dalla memoria le istruzioni che devono essere via via eseguite.

## PROGRAMMA

Per risolvere un determinato problema ogni calcolatore elettronico esegue una serie di operazioni elementari su dati specifici. L'indicazione dell'operazione da eseguire e dei dati su cui operare costituisce un'«istruzione»; una serie di istruzioni che risolve un determinato problema costituisce un «programma».

Preparare un programma significa dunque indicare nell'ordine le operazioni che il calcolatore deve compiere, attribuendo a ciascuna un codice determinato nel linguaggio della macchina.

Il programma codificato viene registrato in memoria prima di iniziare l'elaborazione dei dati per la quale è stato preparato. In tal modo è poi sufficiente fornire al calcolatore solo i dati sui quali il calcolo si deve svolgere.

Dopo che il programma è stato registrato, è sufficiente indicare alla macchina dove si trova la prima istruzione da eseguire; a partire da quel punto, il programma si svolge in modo automatico. Il calcolatore prende in carico, infatti, le singole istruzioni sequenzialmente, ameno che non incontri istruzioni particolari che costringono l'unità di governo a interrompere l'esecuzione sequenziale dei comandi.

Ciò si verifica quando il calcolatore deve operare automaticamente delle scelte in base a risultati (pag. 11) ottenuti a al tipo di dati su cui opera (salti condizionati), o quando, indipendentemente da qualsiasi scelta, deve prendere in considerazione che si trova registrata in memoria, ma che non è quella adiacente a quella che si sta esaminando (salti incondizionati).

Si consideri, ad esempio, di dover risolvere un problema nel corso del quale sia necessario confrontare due dati, «a» e «b». Le esigenze del problema impongono calcoli diversi a seconda che i due valori da confrontare siano eguali o differenti.

Se il confronto dimostrerà l'uguaglianza dei due valori ( $a=b$ ), si eseguirà una serie di operazioni che potrà essere sinteticamente indicata come «sequenza n. 1». In caso contrario ( $a \neq b$ ), la serie di calcoli che si dovrà eseguire sarà differente e verrà indicata come «sequenza n. 2».

Per risolvere il problema il programmatore deve scrivere, dopo un'istruzione di confronto, sia la prima che la seconda sequenza, poiché il programma deve sempre prevedere tutti i casi possibili, ma dovrà far sì che l'esecuzione di ciascuna sequenza escluda l'altra.

Ciò si ottiene nel caso del calcolatore Programma 101, con l'utilizzazione di un particolare tipo di istruzioni, mediante le quali la macchina può raggiungere l'una o l'altra sequenza.

Ogni programma preparato per risolvere un certo problema è valido indipendentemente dai valori numerici dei dati su cui opera.

Dopo la registrazione in memoria esso può essere quindi automaticamente e ripetutamente eseguito dal calcolatore per diverse serie di dati che devono subire un'elaborazione identica a quella programmata.

Poiché capita spesso che una stessa serie di calcoli sia ripetuta più volte durante uno stesso programma, è inutile riscrivere la stessa sequenza di istruzioni, basta scriverla una volta sola ed eseguirla ciclicamente tutte le volte che sia necessario.

### **Unità aritmetico-logica**

Questa unità costituisce la parte del calcolatore capace non solo di eseguire operazioni aritmetiche, ma anche di operare trasferimenti di dati all'interno della memoria e di prendere decisioni logiche.

La parte aritmetica di questa unità prende di solito il nome di «accumulatore», ed è dotata di un organo calcolante detto «addizionatore-sottrattore» che contiene, a fine operazione, il risultato.

### **Unità di governo**

L'unità di governo è costituita dall'insieme degli organi di comando; presiede quindi al funzionamento delle singole parti del calcolatore e prende ordinatamente in carico le istruzioni del programma, le interpreta e le esegue, inviando i corrispondenti comandi agli organi interessati. L'unità di governo è collegata a tutti gli organi del calcolatore.

## Caratteristiche del calcolatore Programma 101

L'Olivetti Programma 101 è un calcolatore elettronico numerico, cioè una macchina capace di elaborare dati e di fornire risultati a elevata velocità operativa. In esso, come illustrato schematicamente, si ravvisano tutti gli organi caratteristici di un calcolatore elettronico:

- *organi di entrata*
- *organi di uscita*
- *memoria*
- *unità di governo e unità aritmetico-logica*

### Organi di entrata

Gli organi di entrata sono:

- la tastiera, composta di tasti numerici e tasti operativi, mediante la quale sono impostati sia i dati che i comandi
- l'unità di lettura e di registrazione delle scheda magnetica che, in assetto di lettura, consente la registrazione del programma e di dati da scheda magnetica in memoria.

### Organi di uscita

Gli organi di uscita sono:

- l'unità di stampa, che scrive i dati serialmente da destra a sinistra su 28 colonne alla velocità di 30 caratteri al secondo
- la già menzionata unità di lettura e di registrazione della scheda magnetica che, in assetto di registrazione, consente il trasferimento del programma e di dati da memoria su scheda magnetica.

### Memoria

La memoria dell'Olivetti Programma 101 è suddivisa in 10 registri.

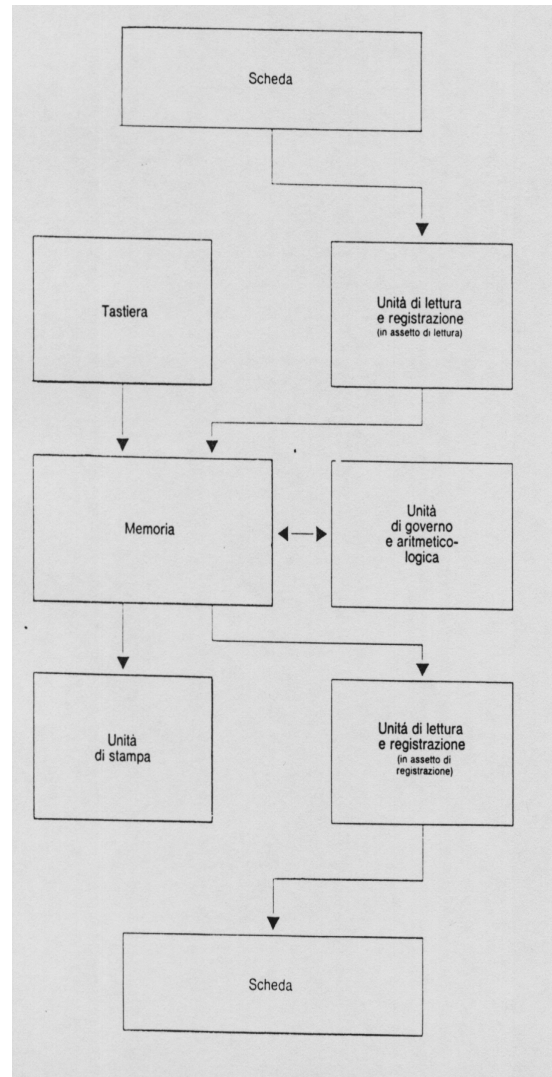
Due registri sono dedicati alla memorizzazione del programma e possono contenere ciascuno 24 istruzioni, o comandi elementari.

Nei restanti otto registri, ciascuno dei quali ha la capacità di 22 cifre più virgola e segno, si possono invece registrare i dati. Se un dato ha un numero di cifre inferiore alla capacità del registro, quest'ultimo non verrà interamente occupato.

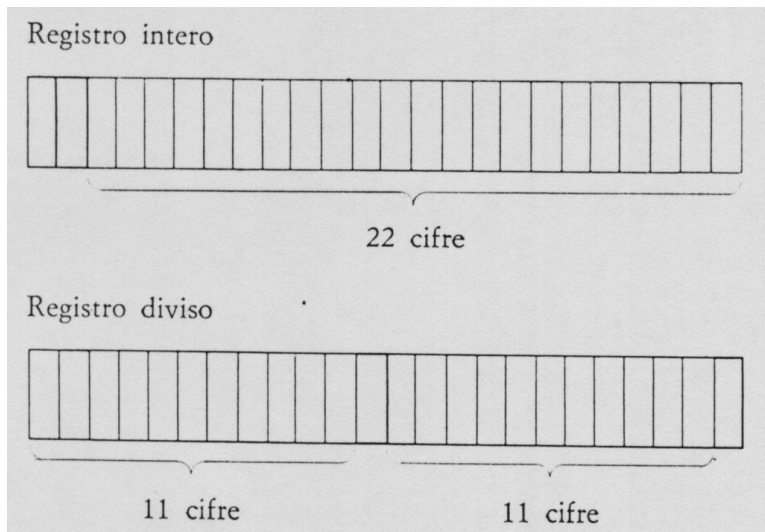
I registri sono denominati M, A, R, B, C, D, E, F, e si possono considerare distinti in due gruppi. (pag. 13) I registri M, A, R sono detti *operativi* in quanto intervengono direttamente nell'esecuzione delle operazioni aritmetiche. Il registro M ha anche l'importante funzione di accogliere tutti i dati impostati in tastiera.

Gli altri registri sono detti *di deposito* e sono utilizzati per memorizzare dati, risultati intermedi di un calcolo e dati costanti.

Ogni registro può contenere un unico dato, di lunghezza massima pari alla sua capacità. Ciascun registro di deposito può tuttavia essere diviso in due parti, in ognuna delle quali si può registrare un dato avente una lunghezza massima di 11 cifre più virgola e segno.



Un registro di deposito può quindi essere utilizzato per la registrazione di dati, secondo una delle seguenti configurazioni:



Nel primo caso il registro è utilizzato nella sua interezza ed ha una capacità di 22 cifre, più virgola e segno; nel secondo, invece, viene utilizzato diviso in due registri ridotti ed indipendenti, ciascuno della capacità di 11 cifre più virgola e segno.

Il registro viene in tale circostanza sdoppiato; la parte destra conserva la denominazione originaria (ad esempio B) e quella sinistra la stessa, scritta in minuscolo (ad esempio b). Durante la stesura dei programmi però si usa scrivere, anziché la lettera minuscola, la maiuscola seguita dal simbolo / (ad esempio B/ per B, D/ per d) in quanto la combinazione

di lettera maiuscola col simbolo / consente di formare da tastiera la corrispondente minuscola. Poiché l'Olivetti Programma 101 è un calcolatore a programma registrato, nella sua memoria si introducono anche i comandi elementari, o istruzioni di programma. I due registri programma, che accolgono esclusivamente istruzioni, hanno una capacità complessiva di 48 istruzioni. Qualora il programma sia costituito da più di 48 istruzioni, quelle che eccedono questo numero possono essere registrate nei registri F, f, E, e, D, d, fino ad un massimo di 120.

(pag. 14)

Ciascun registro interno ha la capacità di 24 istruzioni. Un semiregistro ha la capacità di 11 o 13 istruzioni a seconda che si tratti di della parte destra (contrassegnata con lettera maiuscola) o sinistra (contrassegnata con lettera minuscola).

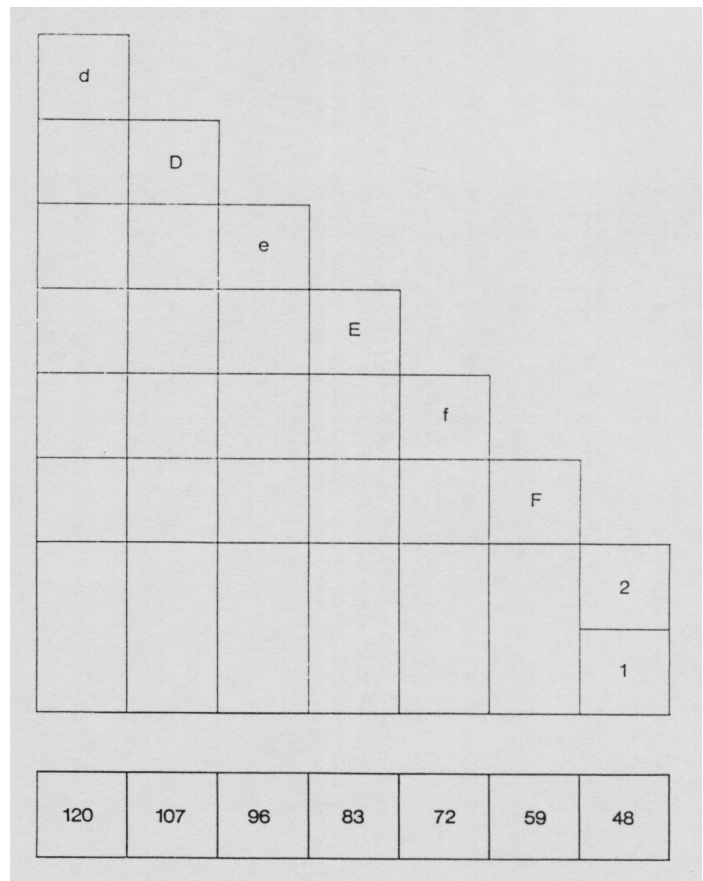
I registri impegnati per il programma non possono essere ovviamente utilizzati per la memorizzazione dei dati.

Esiste però anche la possibilità di utilizzare in modo promiscuo i registri F, f, E, e, D, d, registrando simultaneamente dati ed istruzioni.

#### SCHEDA MAGNETICA

I calcolatori elettronici, come già detto, oltre alla memoria di elaborazione collegata all'unità aritmetico-logica, dispongono in generale di altre memorie dette «ausiliarie», più lente, di elevate capacità e basso costo, utilizzate soltanto a scopo di immagazzinamento di dati. Nel caso dell'Olivetti Programma 101 si può considerare come ausiliaria la scheda magnetica, che ha le dimensioni di una normale scheda meccanografica e sulle quale sono registrate magneticamente istruzioni di programma e dati numerici.

Su una scheda magnetica si può registrare in modo semplice e automatico un programma. E' possibile in tal modo ottenere la registrazione permanente dei programmi su un supporto esterno al calcolatore, nonché la loro semplice e rapida introduzione in memoria ogniqualvolta si desidera utilizzarli.



Nella registrazione della scheda magnetica, cioè nel trasferimento del programma dalla memoria sulla scheda, viene registrato integralmente il contenuto dei registri 1, 2, F, E, D (e di conseguenza anche gli eventuali dati contenuti nei registri F, E, D).

Nelle lettura della scheda, cioè nell'operazione di trasferimento da scheda a memoria, il contenuto della scheda andrà a occupare tutti i registri 1, 2, F, f, E, e, D, d annullandone il precedente contenuto.

## **Unità di governo e unità aritmetico-logica**

L'unità di governo riceve dalla memoria le istruzioni, cioè i comandi che essa interpreta ed esegue attraverso l'unità aritmetico-logica.

Per l'esecuzione dei calcoli il registro operativo A è dotato di un organo detto «addizionatore», che è l'unico organo calcolante della macchina. Per questa sua caratteristica il registro A viene anche denominato «accumulatore».

## **Programmazione**

Un programma è costituito da una successione di istruzioni atte a definire, nel linguaggio proprio del calcolatore, l'ordine secondo cui devono essere effettuate le singole operazioni al fine di risolvere un determinato problema.

Ogni istruzione si divide in due parti: indirizzo e funzione. Le istruzioni composte dalla sola parte di funzione sottintendono l'indicazione del registro M.

Le istruzioni del calcolatore Programma 101 comandano varie funzioni, classificabili nel modo seguente:

- *funzioni aritmetiche*: le quattro operazioni elementari, la radice quadrata, la determinazione del valore assoluto
- *introduzione dei dati e funzioni di trasferimento*: istruzioni di Stop, trasferimenti semplici, scambio tra registri
- *funzioni di servizio*: azzeramento, stampa, interlinea
- *funzioni logiche*: salti incondizionati, salti condizionati.

(pag. 15)

Le istruzioni di un programma devono essere scritte sequenzialmente dal programmatore, sui moduli appositamente studiati per evidenziare ad ogni istruzione la situazione di tutti i registri. Si evitano in tal modo errori dovuti ad una inesatta attribuzione di dati ai singoli registri, soprattutto nei casi in cui si abbiano frequenti trasferimenti.

E' inoltre sempre possibile controllare l'esatta ubicazione delle singole istruzioni che vengono man mano ad occupare i registri programma.

**olivetti**

CALCOLATORE ELETTRONICO  
MODULO DI PROGRAMMAZIONE

Scheda No. \_\_\_\_ di \_\_\_\_ decimali

TITOLO	DATA	CODICE		SCH	ISTRUZ
	a m.	classe	No		

N.	ISTRUZIONI		M	A	B		C		D		E		F		OSSERVAZIONI
	indi- rizzo	fun- zione			B/ B	C/ C	D/ D	E/ E	F/ F	F/ F					

## Le funzioni aritmetiche

### Operazioni aritmetiche elementari

Nelle quattro operazioni aritmetiche elementari (addizione, sottrazione, moltiplicazione, divisione) intervengono sempre due termini o «operandi». La programmazione di queste operazioni richiede la scrittura di uno o più simboli che, registrati nella memoria del calcolatore, sono da questo interpretati come comandi, rispettivamente, di addizione, sottrazione, moltiplicazione o divisione.

*Le operazioni avvengono tra i registri A ed M.* Il calcolatore opera considerando il contenuto di A come primo termine dell'operazione (primo addendo nell'addizione, minuendo nella sottrazione, moltiplicando nella moltiplicazione e dividendo nella divisione) ed il contenuto di M come secondo termine (secondo addendo nell'addizione, sottraendo nella sottrazione, moltiplicatore nella moltiplicazione e divisore nella divisione).

Basta quindi programmare:

+ per comandare un'addizione  
- per comandare una sottrazione  
x per comandare una moltiplicazione  
: per comandare una divisione

Le singole operazioni sono eseguite algebricamente su numeri positivi o negativi, interi o decimali.

Dopo l'esecuzione di un'operazione aritmetica il contenuto dei registri A e M è il seguente:

- *A contiene il risultato che cancella e sostituisce il primo operando*
- *M contiene inalterato il secondo operando.*

#### Esempi

	M	A
prima	5	7
+		
dopo	5	12
prima	4	6
-		
dopo	4	2
prima	8	3
x		
dopo	8	24
prima	2	10
:		
dopo	2	5

Nel caso si eseguano operazioni con numeri decimali, la struttura delle istruzioni non cambia; occorre però stabilire, posizionando opportunamente *l'indicatore dei decimali*, il numero dei decimali che si desidera ottenere nel risultato. Le possibili posizioni dell'indicatore sono 16 (da 0 a 15).

Il calcolatore esegue automaticamente gli incolonnamenti necessari e, nel caso di addizione, sottrazione o moltiplicazione, prende in considerazione tutti i decimali degli operandi, calcolando il risultato completo dell'operazione con eliminazione successiva dei decimali in eccesso rispetto a quelli prefissati.

Nel caso della divisione, invece, il calcolatore aggiunge automaticamente al dividendo tanti zeri, quanti sono necessari per ottenere un risultato con il numero di decimali richiesto.

Dopo l'esecuzione di un'operazione aritmetica con numeri decimali, il contenuto dei registri A e M è il seguente:

- *A* contiene il risultato con il numero di decimali prefissato
- *M* contiene il secondo operando inalterato.

Oltre ai due registri *A* e *M*, nell'esecuzione delle operazioni aritmetiche interviene anche il registro *R* che, al termine di un'addizione, sottrazione o moltiplicazione, contiene il risultato completo dell'operazione, con tutti i decimali che gli competono, indipendentemente dalla posizione dell'indicatore dei decimali.

Al termine della divisione il registro *R* contiene l'eventuale resto aritmeticamente completo nella sua struttura decimale.

Esempi

decimali → 4

	M	A	R
prima	4,7938	7,6329	
+			
dopo	4,7938	12,4267	12,4267
prima	4,7938	7,6329	
x			
dopo	4,7938	36,5905	36,59059602
prima	-5,12	-4,2	
:			
dopo	-5,12	0,8203	0,000064

## Capacità dei registri operativi

I registri *A*, *M*, *R* possono contenere ciascuno un numero di 22 cifre più virgola e segno eventuali. In *A*, nel caso dell'addizione e della moltiplicazione, può esservi anche un risultato di 23 cifre. Tale risultato però non può essere ulteriormente operato. Qualora le singole operazioni comportino risultati con più di 23 cifre, la macchina non esegue il calcolo e segnala «overflow», cioè supero di capacità.

Per determinare il numero di cifre del risultato è opportuno osservare che nel caso di addizioni, sottrazioni o moltiplicazioni, il calcolatore segue l'operazione completa (con tutte le cifre degli operandi) e solo ad operazione avvenuta vengono scartati i decimali non richiesti. Il calcolatore segnala «overflow» se il risultato completo di una delle operazioni sopra specificate supera la capacità massima.

Nel caso della divisione occorre invece tener presente che il calcolatore modifica automaticamente la composizione decimale del dividendo se questo non è atto a fornire un quoziente col numero di decimali richiesto.

Ciò si traduce, in pratica, nell'attribuzione al dividendo di un maggior numero di cifre decimali, come accade del resto nell'esecuzione manuale delle divisioni.

Così ad esempio il calcolatore non può eseguire l'operazione  $54536 : 567,7842$  se i decimali richiesti sono quindici, poiché occorrerebbe considerare il dividendo come avente 24 cifre, di cui 15 decimali.

E' necessario infatti aggiungere al dividendo altri 4 zeri per annullare la virgola del divisore e altri 15 zeri per ottenere anche i 15 decimali richiesti, per cui si ha

$$5 + 4 + 15 = 24 \text{ cifre}$$

## Operazioni con dati contenuti nei registri di deposito

Come si è visto, i due operandi di un'operazione aritmetica debbono essere contenuti nei due registri *A* e *M*.

Qualora invece i dati sui quali si desidera operare siano contenuti in altri registri (*B*, *C*, *D*, *E*, *F*) essi dovranno essere preventivamente trasferiti nei suddetti registri operativi. Nel caso del primo termine è sempre necessario prevedere un'istruzione di trasferimento in *A*; nel caso del secondo termine, sia il trasferimento in *M* sia l'operazione sono comandati con una sola istru-



zione contenente, accanto all'indicazione dell'operazione, il simbolo del registro da cui si richiama il secondo termine.

Ad esempio, volendo sommare ad un numero contenuto in A un numero contenuto in C, si programmerà l'istruzione C +, volendo invece sottrarre dal numero contenuto in A il numero contenuto in F, si programmerà l'istruzione F -.

L'esecuzione dell'operazione avviene in due fasi:

**1<sup>a</sup> fase:** il calcolatore trasferisce automaticamente il contenuto del registro indicato nell'istruzione nel registro M, mantenendolo inalterato anche nel registro d'origine

**2<sup>a</sup> fase:** si esegue l'operazione programmata tra i contenuti dei registri A ed M. Il risultato, con il numero di decimali desiderato, è contenuto in A, quello completo (oppure il resto in caso di divisione) è contenuto in R.

Istruzione	1 <sup>a</sup> fase	2 <sup>a</sup> fase	Registro A	Registro R
B+	B→M	A+M	risultato troncato	risultato completo
B-	B→M	A-M	risultato troncato	risultato completo
Bx	B→M	AxM	risultato troncato	risultato completo
B:	B→M	A:M	quoziente	resto

La logica descritta per il registro B vale per tutti i registri di deposito (F, E, D, C) nonché per i registri operativi (A, R) come sarà illustrato in seguito.

Esempi

decimali → 0

	M	A	R	B
prima	5	7	9	3
B+				
dopo	3	10	10	3
	M	A	R	C
prima	5	7,5	9	3,6
C-				
dopo	3,6	3	3,9	3,6
	M	A	R	D
prima	7	5,2	9	6,4
Dx				
dopo	6,4	33	33,28	6,4
	M	A	R	B
prima	5	7	9	3
B:				
dopo	3	2	1	3

## Uso dei registri divisi nelle operazioni aritmetiche

Per eseguire operazioni aritmetiche nelle quali il secondo termine sia contenuto in un registro diviso (e il primo sia già contenuto in A) occorre che l'istruzione programmata contenga, oltre al simbolo dell'operazione, anche l'indicazione della parte del registro in cui è contenuto il secondo operando. Se la parte è quella di destra, nell'istruzione si scrive semplicemente il nome del registro; se invece è quella di sinistra, si scrive lo stesso nome seguito dal simbolo / (Split).

## Esempi

	M	A	R	b	B
prima	2	7	3	8	10
B/+ dopo	8	15	15	8	10
prima	2	7	3	8	10
B+ dopo	10	17	17	8	10

I registri A, M e R non possono mai essere divisi.

## Radice quadrata

Il calcolatore esegue automaticamente e direttamente la radice quadrata.

Diverse sono le istruzioni da programmare, a seconda che il radicando si trovi nel registro M, oppure in un altro registro, intero o diviso:

a) quando il radicando si trova nel registro M, l'istruzione corrispondente è costituita solo dal simbolo  $\sqrt{\quad}$ .

Il calcolatore, nell'eseguire l'istruzione  $\sqrt{\quad}$ , estrae la radice quadrata del numero contenuto nel registro M, dopo averlo preventivamente trasferito in A. Il risultato si forma ancora in A, con il numero di decimali richiesto.

Alla fine dell'operazione, nel registro M si trova il risultato moltiplicato per due, nel registro R invece un dato non significativo.

Operando su un numero decimale il calcolatore corregge automaticamente la composizione decimale del radicando (analogamente a quanto avviene nella divisione per il dividendo) in funzione del numero di decimali richiesto.

### Esempi

decimali  $\rightarrow$  0

	M	A
prima	10	7
$\sqrt{\quad}$ dopo	6	3

decimali  $\rightarrow$  1

prima	12	8
$\sqrt{\quad}$ dopo	6,8	3,4

b) quando il radicando si trova in un registro diverso da M, l'istruzione corrispondente deve essere programmata facendo precedere al simbolo  $\sqrt{\quad}$  l'indicazione del registro in cui si trova il radicando.

L'istruzione poi è eseguita dal calcolatore in due fasi:

**1ª fase:** il contenuto del registro indicato dall'istruzione viene trasferito nel registro A, rimanendo inalterato nel registro d'origine

**2ª fase:** si estrae la radice quadrata del numero contenuto in A (come nel caso precedentemente esaminato).

<i>Istruzione</i>	<i>1<sup>a</sup> fase</i>	<i>2<sup>a</sup> fase</i>	<i>Registro A</i>
B√	B→A	√ A→A	risultato troncato
C√	C→A	√ A→A	risultato troncato
R√	R→A	√ A→A	risultato troncato
B:	B→M	A:M	quoziente                      resto

Esempi

decimali → 0

	M	A	B
prima B√	8	4	27
dopo	10	5	27

decimali → 1

	M	A	C
prima C√	12	8	12
dopo	6,8	3,4	12

c) quando il radicando si trova in un registro diviso, l'istruzione corrispondente deve essere programmata facendo precedere al simbolo √ l'indicazione della parte di registro in cui si trova il radicando.

L'esecuzione dell'operazione avviene come nel caso precedente.

<i>Istruzione</i>	<i>1<sup>a</sup> fase</i>	<i>2<sup>a</sup> fase</i>	<i>Registro A</i>
B/√	b→A	√ A→A	risultato troncato
F/√	f→A	√ A→A	risultato troncato

(pag. 20)

Esempi

decimali → 0

	M	A	b	B
prima B/√	4	7	17	38
dopo	8	4	17	38
prima B√	4	7	17	38
dopo	12	6	17	38

### Operazioni con dati contenuti nei registri operativi

E' possibile che il contenuto di A debba costituire anche il secondo termine di un'operazione, oppure che questo si trovi nel registro operativo R.

In tal caso A ed R, mantenendo sempre la funzione di registri operativi, assumono anche una funzione analoga a quella dei registri di deposito. Occorre quindi indicarli nell'istruzione programmata prima del simbolo dell'operazione da eseguire.

La logica di esecuzione delle operazioni è uguale a quella vista per i registri di deposito, con la differenza che in questo caso nessuno dei due operandi si mantiene inalterato nel registro originario in quanto in esso si forma, a fine operazione il risultato.

**1° caso:** il registro indicato nell'istruzione è il registro A.

<i>Istruzione</i>	<i>1<sup>a</sup> fase</i>	<i>2<sup>a</sup> fase</i>	<i>Registro A</i>	<i>Registro R</i>
A+	A→M	A+M	risultato troncato	risultato completo
A-	A→M	A-M	risultato troncato	risultato completo
Ax	A→M	AxM	risultato troncato	risultato completo
A:	A→M	A:M	esultato troncato	resto
A√	A→A	√ A	risultato troncato	dato non significativo

Con le suddette operazioni si ottengono i risultati seguenti:

A+ viene raddoppiato il contenuto di A

A- viene azzerato il contenuto di A

Ax il contenuto di A viene moltiplicato per se stesso

A: si ottiene come quoziente 1 (purché il contenuto di A sia ≠ 0)

**2° caso:** il registro indicato nell'istruzione è il registro R.

<i>Istruzione</i>	<i>1<sup>a</sup> fase</i>	<i>2<sup>a</sup> fase</i>	<i>Registro A</i>	<i>Registro R</i>
R+	R→M	A+M	risultato troncato	risultato completo
R-	R→M	A-M	risultato troncato	risultato completo
Rx	R→M	AxM	risultato troncato	risultato completo
R:	R→M	A:M	esultato troncato	resto
R√	R→A	√ A	risultato troncato	dato non significativo

## DETERMINAZIONE DEL VALORE ASSOLUTO

L'istruzione speciale A↑ elimina l'eventuale segno negativo del contenuto di A rendendolo sempre positivo.

Due casi sono possibili:

- il contenuto di A è positivo. L'istruzione di valore assoluto non opera alcuna trasformazione (pag. 21)
- il contenuto di A è negativo. L'istruzione di valore assoluto lo trasforma in positivo.

Tale operazione si definisce determinazione del valore assoluto di A, è l'istruzione relativa A↑ lascia inalterato il contenuto degli altri due registri M ed R e di tutti i registri di deposito

## Esempio

	M	A	R	B
prima	3	-6	4	8
A ↑				
dopo	3	6	4	8

## SEQUENZE DI OPERAZIONI ARITMETICHE

Un calcolo che richieda l'esecuzione di più operazioni aritmetiche successive da luogo a una sequenza di istruzioni che, eseguite dal calcolatore una dopo l'altra, permettono di pervenire al risultato finale.

Il risultato di un'operazione può essere direttamente sfruttato come primo termine di un'operazione successiva, in quanto già si trova nel registro A.

Si debba ad esempio calcolare:

$$a) \quad \frac{(36+12) \cdot 17}{18}$$

o in generale un'espressione del tipo:

$$\frac{(x_1+x_2) \cdot x_3}{x_4}$$

supponendo che 36 (o in generale  $x_1$ ) sia nel registro A, 12 ( $x_2$ ) in M, 17 ( $x_3$ ) in b, 18 ( $x_4$ ) in d. Le istruzioni da programmare saranno:

decimali → 0

Istruzione	M	A	b	d	Descrizione
	12	36	17	18	
+	12	48			Esecuzione della somma $36 + 12 = 48$ . Il risultato si forma in A e può essere considerato come fattore della moltiplicazione successiva
B/x	17	816	17		Moltiplicazione di $48 \times 17 = 816$
D/:	18	45		18	Il risultato del prodotto, 816, rappresenta il dividendo della divisione per 18; per cui si ha $816:18=45$

$$b) \quad \left( \frac{36}{9} \cdot 25 \right) + 22$$

o in generale un'espressione del tipo:

$$\left( \frac{x_1}{x_2} \cdot x_3 \right) + x_4$$

supponendo che  $x_1$  sia in A,  $x_2$  in C,  $x_3$  in D,  $x_4$  in B. Nel caso specifico  $x_1 = 36$ ,  $x_2 = 9$ ,  $x_3 = 25$ ,  $x_4 = 22$ .

Le istruzioni da programmare saranno:

decimali → 0

Istruzione	M	A	B	C	D	Descrizione
		36	22	9	25	
C :	9	4		9		Esecuzione della divisione $36 : 9 = 4$
D x	25	100			25	Moltiplicazione di 4 (quoziente della divisione precedente) per 25
B +	22	122			22	Somma di 100 (risultato del prodotto precedente) più 22, ottenendo il risultato finale

c) 
$$\frac{2 \cdot 8 + 12 - 7}{5}$$

o in generale un'espressione del tipo:

$$\frac{2 \cdot x_1 + x_2 - x_3}{x_4}$$

supponendo che  $x_1$  sia nel registro A,  $x_2$  in B,  $x_3$  in F,  $x_4$  in b.  
Le istruzioni da programmare saranno:

decimali → 0

Istruzione	M	A	b	B	F	Descrizione
		8	5	12	7	
A +	8	16				Somma di $8 + 8 = 16$
B +	12	28		12		Somma di $16 + 12 = 28$
F -	7	21			7	Sottrazione di 7 dal risultato precedente, ottenendo 21
B :	5	4	5			Esecuzione della divisione $21 : 5 = 4$ , ottenendo così il risultato finale

## Introduzione dei dati e funzioni di trasferimento

### Posizionamento dei dati nei registri

Nella trattazione delle operazioni aritmetiche si sono considerati vari casi, semplici o complessi, ma si è sempre supposto che i dati fossero già introdotti nella memoria del calcolatore.

In caso contrario è opportuno definire come si programmi l'introduzione di dati in memoria, e come questi possano essere posizionati nei singoli registri.

Come si è visto, il risultato di un'operazione si forma nel registro A, ma se questo risultato deve essere provvisoriamente conservato e ripreso in uno stadio più avanzato del calcolo, occorre evidentemente toglierlo da A per poter eseguire le operazioni che precedono la sua riutilizzazione.

Occorre perciò conoscere in particolare come, e con quali istruzioni, si possa trasferire in un registro di deposito un risultato intermedio e quali siano le varie operazioni di trasferimento.

### Introduzione dei dati

Generalmente ogni volta che si deve introdurre un dato nel calcolatore occorre un intervento manuale dell'operatore attraverso la tastiera.

Poiché può essere necessario introdurre dati in diversi momenti del programma, che si svolge automaticamente, è ovvio che questo debba contenere, in punti prestabiliti, un'istruzione particolare che faccia arrestare l'esame sequenziale delle istruzioni da parte del calcolatore e permetta all'operatore di impostare sulla tastiera il dato desiderato.

E questa l'istruzione di Stop, che si programma abbassando il tasto «S» (Start-Stop), che in questo caso ha la funzione di interrompere l'esecuzione delle istruzioni.

Dopo l'arresto della macchina, l'operatore imposta sulla tastiera il dato da introdurre abbassando i tasti corrispondenti alle cifre volute, cominciando da quella più significativa come nella normale scrittura di un numero.

Se il numero da introdurre è decimale, si preme il tasto «,» (virgola) dopo l'impostazione della sua parte intera: in tal modo viene introdotta la virgola nella posizione che le è propria. Se il numero è solo decimale, senza parte intera, deve essere sempre impostato uno zero prima della virgola.

Se il numero è negativo, l'abbassamento del tasto «-» (meno) sulla tastiera numerica può indifferentemente precedere o seguire l'impostazione del numero stesso.

Un numero non accompagnato dal segno «-» è considerato sempre positivo.

*Il numero impostato sulla tastiera è automaticamente introdotto nel registro M.*

Dopo l'introduzione di un dato, il normale svolgimento del programma può essere ripreso al punto in cui era stato interrotto, abbassando di nuovo il tasto « S », che in questo caso ha la funzione di riavviare l'esecuzione delle istruzioni (Start).

Il dato introdotto viene automaticamente stampato col simbolo S a destra.

Ad esempio, si debba calcolare:

$$(x_1 + x_2) x_3$$

supponendo che:

$x_1 = 3$  si trovi già nel registro A

$x_2 = 4$  si trovi già nel registro C

e si debba introdurre  $x_3 = 8$ .

Le istruzioni da programmare saranno:

Istruzione	M	A	C	Descrizione
	3	4		
C+	4	7	4	Somma di 3 + 4. = 7
S	8	7	4	Impostazione di 8 che si posiziona nel registro M
x	8	56		Moltiplicazione di 7 (risultato della somma precedente) per 8 (nuovo dato impostato)

Dato che il numero 8 verrà impostato all'istruzione S, e si troverà quindi in M, basta programmare l'istruzione x per ottenere il risultato finale.

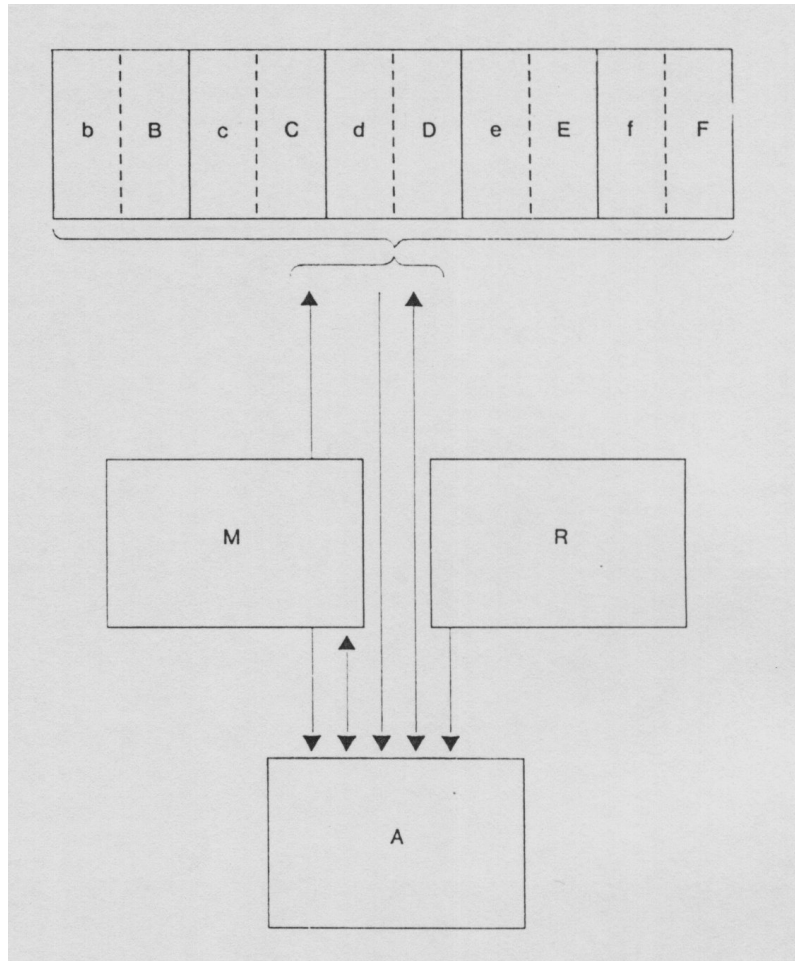
### Funzioni di trasferimento

Le istruzioni di trasferimento comandano lo spostamento di dati da un registro a un altro e sono di quattro tipi:

- trasferimento del contenuto di M in un altro registro
- trasferimento del contenuto di qualsivoglia registro in A
- scambio del contenuto di A con il contenuto di un altro registro
- trasferimento della parte decimale di A in M.

Nello schema che segue sono illustrati i possibili collegamenti fra i registri.





Per ogni tipo di trasferimento vale una considerazione di carattere generale: *la memoria del calcolatore Programma 101 è di tipo conservativo in estrazione e sostitutivo in immissione; di conseguenza il registro da cui si estrae un dato continuerà a conservare il data stesso, mentre il registro che riceve un dato perde il suo precedente contenuto.*

### TRASFERIMENTO DA M

Poiché ogni dato impostato sulla tastiera viene introdotto in M, e indispensabile poter spostare altrove il dato stesso prima che una successiva immissione lo distrugga.

L'istruzione mediante la quale viene programmata l'operazione di trasferimento ha il simbolo (pag. 25)  $\uparrow$ , preceduto dall'indicazione del registro in cui si vuole trasferire il contenuto di M. Questa operazione di trasferimento lascia inalterato il contenuto di M.

Esempio

	M	A	C
prima	10	36	15
$C\uparrow$			
dopo	10	36	10

Il registro specificato nell'istruzione può essere sia intero sia diviso.  
Non sono però utilizzate le istruzioni:

$\uparrow$ ,  $R\uparrow$ ,  $A\uparrow$

La prima avrebbe il significato logico di trasferire il contenuto di M in M e non ha quindi alcuna utilità, la seconda non opera e l'ultima, che avrebbe il significato logico di trasferire M in A, non

è utilizzata come tale, ma con un significato particolare che verrà esaminato più avanti.

### TRASFERIMENTO IN A

Tutte le operazioni aritmetiche, come è noto, richiedono che il primo operando si trovi in A. Se il primo operando non è in A, ma in un altro registro, occorre trasferirlo.

A ciò serve l'istruzione di «trasferimento in A» che introduce direttamente in A il contenuto di un determinato registro senza alterare il contenuto del registro stesso e di M.

L'istruzione si programma col simbolo  $\downarrow$  preceduto dall'indicazione del registro il cui contenuto si desidera trasferire.

Esempi

$\downarrow$   
R  $\downarrow$

B/ $\downarrow$   
F  $\downarrow$

Come si può osservare dagli esempi riportati, il registro specificato nell'istruzione può essere sia intero sia diviso.

L'istruzione  $\downarrow$  trasferisce il contenuto di M in A. L'istruzione A  $\downarrow$  non opera, poiché avrebbe il significato logico di trasferire il contenuto di A in A e non sarebbe quindi di alcuna utilità.

Esempi

	M	A	R	B
prima	24	36	48	12
$R\downarrow$				
dopo	24	48	48	12

	M	A	R
prima	9	4	7
$\downarrow$			
dopo	9	4	7

(pag. 26)

Si debba aggiungere 4 a 5, partendo da questa situazione:

	M	A	B
	4	6	5

Istruzione	M	A	B	Descrizione
	4	6	5	
$B\downarrow$	4	5	5	Trasferimento di 5 nel registro A
+	4	9	5	Somma di 5 + 4

(pag. 26)

Si debba aggiungere il contenuto del registro B a quello di C, partendo da questa situazione:

	M	A	B	C
	4	6	5	10

Istruzione	M	A	B	C	Descrizione
	4	6	5	10	
C↓	4	10		10	Trasferimento in A di 10 (primo operando dell'addizione successiva)
B+	5	15	5	10	Somma di 10 + 5

Se si fosse invece considerato il contenuto di B come primo operando e si fosse voluto aggiungere a esso il dato in C, si sarebbero programmate nell'ordine le istruzioni B↓ e C +.

Il contenuto finale dei registri sarebbe stato:

M	A	C	B
10	15	5	10

Cambiando l'ordine degli addendi il risultato non cambia; cambia però il contenuto di M. Può essere importante ricordare ciò quando si programma una sequenza di calcoli, onde evitare errori derivanti da un'erronea valutazione del contenuto dei registri.

## SCAMBIO CON A

Per spostare in un altro registro un risultato formatosi in A, si impiega l'istruzione di scambio  $\updownarrow$  che consente di trasferire il contenuto di A in un registro determinato e con temporaneamente il contenuto di questo in A.

L'istruzione viene programmata scrivendo il simbolo  $\updownarrow$  preceduto dall'indicazione del registro con cui viene eseguito lo scambio. Come al solito, se il registro interessato all'operazione è M, il suo simbolo non viene indicato.

### Esempi

$\updownarrow$  M $\leftrightarrow$ A  
C  $\updownarrow$  C $\leftrightarrow$ A  
E  $\updownarrow$  E $\leftrightarrow$ A

L'istruzione R  $\updownarrow$  non opera e non va usata. L'operazione di scambio non altera il contenuto del registro M, tranne che nello scambio tra M ed A. Va rilevato che lo scambio è l'unica operazione che permette di trasferire un dato da A in altro registro (di deposito).

### Esempio

	M	A	R	B
prima	6	9	12	15
B $\updownarrow$				
dopo	6	15	12	9

Supposto che i registri abbiano il seguente contenuto:

M	A	C	D
4	10	8	7

per memorizzare 7 in D e sottrarre 4 da 10, (pag. 27) le istruzioni da programmare saranno:

Istruzione	M	A	C	D	Descrizione
	4	7	10	8	
D↑	8			7	Scambio del contenuto di A con quello di D
C↓		10	10		Trasferimento di 10 nel registro A
-	4	6	10	7	Esecuzione della sottrazione 10 - 4 = 6

La prima istruzione programmata invia 7 in D e 8 in A, Ai fini del calcolo, il trasferimento di 8 in A non ha importanza, poiché occorre avere 10 come primo operando nella successiva sottrazione; ciò si ottiene trasferendo il contenuto di C in A mediante l'istruzione C↓.

Poiché spesso un problema dal punto di vista della programmazione non ha una soluzione unica, si possono avere diverse sequenze di istruzioni che portano allo stesso risultato. Nel caso in esame, ad esempio, si può giungere a un risultato identico programmando:

Istruzione	M	A	C	D	Descrizione
	4	7	10	8	
D↑	8			7	Scambio del contenuto di A con quello di D
C↑		10		8	Scambio del contenuto di A con quello di C, richiamando così 10 in A
-	4	6	8	7	Esecuzione della sottrazione 10 - 4 = 6

Occorre osservare che in tal modo risulta cambiato il contenuto di C poiché il trasferimento del primo operando in A si è ottenuto mediante uno scambio.

La situazione finale del contenuto dei registri è pertanto diversa.

Si debba calcolare:

$$(124 : 12) \times 15 + (157 : 6)$$

Le istruzioni richieste saranno:

Istruzione	M	A	R	B	Descrizione
S	124				Impostazione del dividendo
↓	124	124			Trasferimento del dividendo in A
S	12	124			Impostazione del divisore
:	12	10	4		Esecuzione della prima divisione
S	15	10	4		Impostazione del moltiplicatore
x	15	150	150		Esecuzione della moltiplicazione
B↑	15		150	150	Memorizzazione del risultato della moltiplicazione in B
S	157		150	150	Impostazione del dividendo della seconda divisione
↓	157	157	150	150	Trasferimento del dividendo in A
S	6	157	150	150	Impostazione del divisore
:	6	26	1	150	Esecuzione della seconda divisione
B+	150	176	176	150	Richiamo del risultato della moltiplicazione sommandolo al contenuto di A

decimali → 0

(pag. 28)

## TRASFERIMENTO DELLA PARTE DECIMALE

L'istruzione /↑ comanda il trasferimento in M della parte decimale della cifra contenuta in A, che mantiene inalterato il suo contenuto.

Non si deve attribuire ai simboli / e  $\uparrow$  il significato logico che essi hanno normalmente; la loro combinazione ha solo il valore di una particolare codificazione.

Esempio

	M	A	R	B
prima	2,6	5,231	21,8	12
/ $\uparrow$				
dopo	0,231	5,231	21,8	12

Per illustrare più efficacemente l'utilità della istruzione in esame, si supponga di avere un tempo espresso in ore e frazione decimale di ora (ad esempio 5,31) e di volerlo trasformare in ore, minuti e secondi. La trasformazione viene programmata tenendo presente che un'ora corrisponde a 60 minuti e un minuto a 60 secondi.

Istruzione	M	A	B	Descrizione	decimali $\rightarrow$ 1
S	60			Impostazione di 60	
B $\uparrow$	60		60	Trasferimento di 60 in B	
S	5,31			Impostazione di 5,31	
$\downarrow$	5,31	5,31		Trasferimento di 5,31 in A	
/ $\uparrow$	0,31	5,31		Parte decimale di A in M	
-	0,31	5,0		$5,31 - 0,31 = 5$ (numero delle ore)	
$\downarrow$	0,31	0,31		Trasferimento della frazione decimale di ora nel registro A	
B x	60	18,6	60	$0,31 \times 60$ (determinazione dei minuti)	
/ $\uparrow$	0,6	18,6		Parte decimale di A in M	
-	0,6	18,0		$18,6 - 0,6 = 18$ (numero dei minuti)	
$\downarrow$	0,6	0,6		Trasferimento della frazione decimale di minuti nel registro A	
B x	60	36,0	60	$0,6 \times 60$ (determinazione dei secondi)	
/ $\uparrow$	0,0	36,0		Parte decimale di A in M	
-	0,0	36,0		$36,0 - 0,0 = 36$ (numero dei secondi)	

Si osserva che con le istruzioni /  $\uparrow$ , -, si ottiene in A il dato senza la sua parte decimale; con la sequenza /  $\uparrow$ ,  $\downarrow$ , si ottiene in A il dato senza la sua parte intera. E' così possibile operare indipendentemente sia sulla sola parte intera sia sulla sola parte decimale.

## Le funzioni di servizio

### Istruzione di stampa

Come in ogni calcolatore elettronico, anche nell'Olivetti Programma 101 la stampa dei risultati deve essere appositamente comandata.

A ciò provvede l'istruzione di stampa mediante la quale si può stampare non solo un risultato, ma anche un dato contenuto in qualsiasi registro.

Essa viene programmata col simbolo  $\diamond$  preceduto dall'indicazione del registro il cui contenuto si deve stampare. Il calcolatore stampa i risultati con il numero di decimali richiesto. Il contenuto del registro R viene sempre stampato per intero.

Esempio

- Con A  $\diamond$  si stampa il contenuto di A
- Con C/ $\diamond$  si stampa il contenuto di c

Il contenuto del registro interessato all'operazione di stampa non viene alterato.

In stampa si ottiene, oltre al dato richiesto, anche l'indicazione dell'istruzione corrispondente.

Esempio

123,45 b ◇  
2589,90 A ◇

Ciò permette di differenziare e individuare più facilmente i dati stampati. A questo scopo, per distinguere l'una dall'altra due stampe del contenuto di A, si possono utilizzare le istruzioni A◇ e A/◇ che vengono interpretate dall'unità di governo come due comandi analoghi, pur provocando la stampa di codici diversi.

Supposto che il contenuto di A sia 4284,5 si ottiene:

4284,5 A ◇    programmando A ◇  
4284,5 a ◇    programmando A /◇

Esempio

$(a \times b)^2$   
-----  
c

Le istruzioni sono le seguenti:

S        Impostazione di a  
↓        Trasferimento di a in A  
S        Impostazione di b  
x        Moltiplicazione di a x b  
A x       $(a \times b)^2$   
S        Impostazione di c  
:        Divisione del contenuto di A per c  
A ◇     Stampa del risultato.

## Istruzione di azzeramento

Per azzeramento di un registro s'intende la operazione di cancellazione del contenuto del registro stesso.

Questa operazione si programma con il simbolo \*, preceduto dall'indicazione del registro che si vuole azzerare.

I registri M ed R non possono mai essere azzerati utilizzando questa istruzione. L'azzeramento di M si può realizzare con una operazione di scambio ↓, qualora il contenuto di A sia nullo.

L'azzeramento di R, invece, si può ottenere come risultato di una moltiplicazione con un fattore nullo; il prodotto infatti (evidentemente nullo) si forma sia in A sia in R.

L'istruzione \* è necessaria quando si vuole utilizzare nella sua capacità intera un registro precedentemente diviso.

Se ad esempio si vuole disporre di B come registro intero, dopo aver utilizzato i registri (pag. 30) divisi b e B, per evitare l'«overflow», occorre azzerare preventivamente b, programmando B/\*.

Nel caso opposto, per passare dal registro intero ai registri divisi, si deve programmare l'azzeramento del registro intero se il suo contenuto supera le 11 cifre.

Valgono, per ulteriore chiarimento, gli esempi seguenti:

Istruzione	M	b	B
	15.000.000.000.000	2450	3320
B/*		0	3320
B↑		15.000.000.000.000	

Istruzione	M	b	B
	3500	24.000.000.000	.000.000
B*		0	0
B/ ↑		3500	0

## Interlinea

L'istruzione /◇ fa avanzare di una interlinea, senza stampa, la striscia di carta. Viene utilizzata per ottenere una maggiore spaziatura verticale tra due successive righe di stampa, per distanziare, ad esempio, la stampa del risultato da quella dei dati di calcolo.

Anche in questa istruzione i simboli / e ◇ non hanno alcun riferimento con il loro significato logico normale.

## Le funzioni logiche

### Principi generali

Le istruzioni di un programma sono esaminate dal calcolatore sequenzialmente.

Spesso però si presenta la necessità di dover accedere a una o più istruzioni che non seguono ordinatamente l'ultima eseguita.

In tal caso si ricorre alle istruzioni di salto, che comandano al calcolatore di interrompere l'esame sequenziale delle istruzioni e di saltare a un diverso punto prestabilito dal programma. Le istruzioni di salto vengono qualificate come funzioni logiche.

Due sono gli elementi che caratterizzano un salto: il punto in cui si interrompe la normale sequenza di istruzioni, detto *origine del salto*, e il punto di arrivo, a partire dal quale viene ripreso il normale esame delle istruzioni, detto *destinazione del salto*.

I salti sono di due tipi:

- *salti incondizionati*, che vengono eseguiti, indipendentemente da qualsiasi condizione, alla sola lettura dell'istruzione corrispondente
- *salti condizionati*, che alla lettura dell'istruzione corrispondente vengono eseguiti solo se sussistono certe condizioni.

### Salti incondizionati

La programmazione di un salto si effettua mediante due istruzioni.

La prima, che si chiama *istruzione di origine*, indica il punto del programma in cui ha inizio il salto; la seconda, detta *istruzione di destinazione*, indica il punto di arrivo del salto a partire dal quale le successive istruzioni sono di nuovo prese in esame sequenzialmente.

Nel caso di salto incondizionato il calcolatore, quando legge nel programma un'istruzione di origine, salta sempre alla corrispondente istruzione di destinazione, indipendentemente da qualsiasi condizione.

In un programma si possono trovare, per esigenze diverse, più salti; e indispensabile, di conseguenza, individuare in modo univoco la destinazione corrispondente a ciascun punto di origine.

Data l'origine del primo salto, la sua destinazione deve poter essere riconosciuta in modo tale che non si confonda con la destinazione del secondo, del terzo, ecc.

Le due istruzioni di un salto devono perciò essere collegate tra loro in modo che, fissata l'origine, risulti determinata la destinazione corrispondente.

Quando il calcolatore legge nel programma una istruzione di origine, ricerca automaticamente l'istruzione di destinazione corrispondente, a partire dalla prima istruzione del programma. Se per errore sono state programmate due istruzioni di destinazione eguali, il calcolatore si porterà sempre sulla prima ignorando la seconda.

Le coppie di istruzioni di salto incondizionato di cui si può disporre sono 16.

Istruzione di origine	Istruzione di destinazione
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

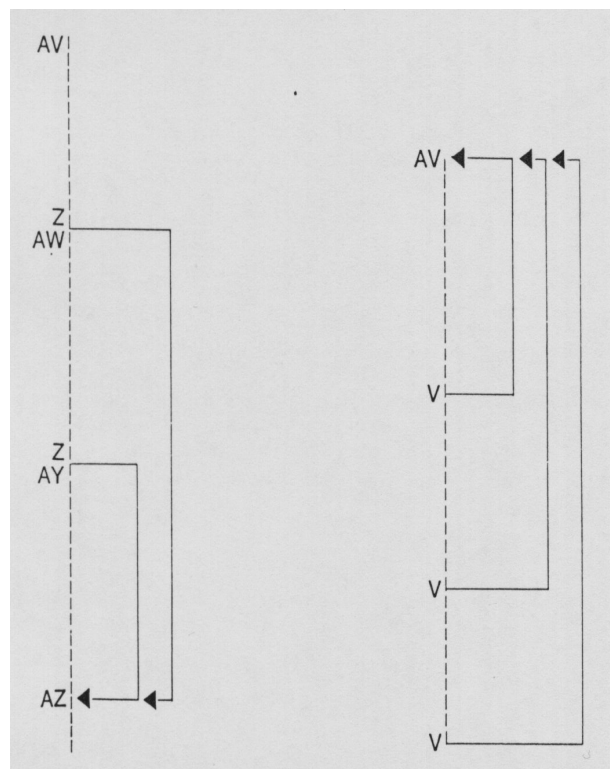
(pag. 32)

In forma abbreviata le predette coppie di istruzioni possono venire rappresentate come segue:

Origine del salto	Destinazione del salto
<b>Δ</b>	<b>A Δ</b>
<b>C Δ</b>	<b>B Δ</b>
<b>D Δ</b>	<b>E Δ</b>
<b>R Δ</b>	<b>F Δ</b>

dove: **Δ = V, W, Y, Z**

In taluni programmi è possibile avere più punti di origine per un'unica destinazione. L'esempio che segue indica due sequenze di istruzioni aventi più punti di origine per uno stesso punto di destinazione.





## AVVIO DELL'ESECUZIONE DI UN PROGRAMMA

Per consentire al calcolatore di accedere alla prima istruzione di un programma è necessario indicare all'unità di governo dove essa si trova registrata. Il calcolatore inizierà da lì l'esecuzione di tutte le istruzioni programmate. A questo scopo esistono sulla tastiera i tasti V, W, Y, Z, che selezionano ciascuno un punto di destinazione di salto incondizionato.

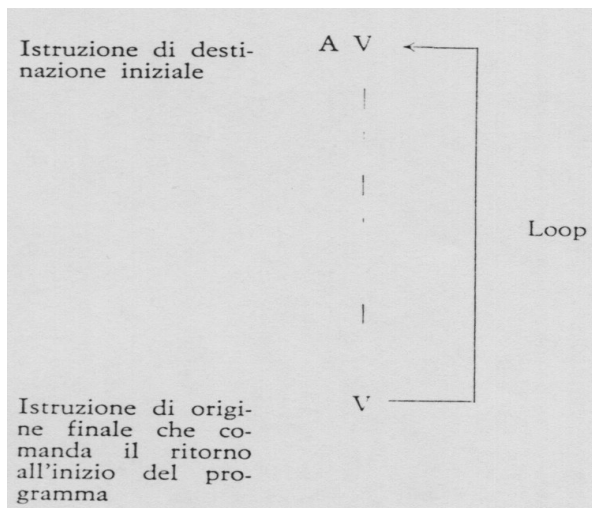
Nella codificazione di ogni programma è perciò opportuno mettere come prima istruzione una destinazione di salto incondizionato (AV, AW, AY, AZ) e come ultima un'istruzione di origine (V, W, Y, Z) che riporti il calcolatore al punto di inizio.

Se, ad esempio, si preme il tasto «V», il calcolatore ricerca l'istruzione AV e comincia da questa l'esame sequenziale delle istruzioni successive.

## ESECUZIONE RIPETUTA DI UN PROGRAMMA

Si supponga di dover calcolare:

$$x_1 + x_2 + x_3$$



su n serie di dati (ad esempio 2 + 3 + 4, ecc.). E' chiaro che non è pensabile di scrivere n volte le istruzioni che risolvono questo calcolo, perché si supererebbe facilmente la capacità della memoria; basterà programmarle una sola volta e utilizzarle n volte.

Ciò si ottiene collocando alla fine della sequenza che si desidera ripetere, un'istruzione di origine e al principio di tale sequenza l'istruzione di destinazione corrispondente, affinché il calcolatore ritorni di nuovo al punto di partenza per ripeterla. Si crea con tale tecnica ciò che viene generalmente chiamato un «loop» o «ciclo».

(pag. 33)

Un loop permette di ripetere più volte le stesse istruzioni di un programma.

Esecuzione ripetuta di un programma

L'utilizzazione del loop può riferirsi all'intero programma o a parte di essa, qualora sia necessaria la ripetizione di una sola sua parte. Le considerazioni 'suesposte hanno un valore meramente esemplificativo e teorico, in quanto nella programmazione di una sequenza di loop occorre prevedere una possibilità di interruzione del ciclo, senza di che il calcolatore ripeterebbe all'infinito la sequenza. Tale interruzione, che deve essere sempre prevista nel programma, si comanda con l'istruzione S (o con la programmazione di un salto condizionato).

## APPLICAZIONI RELATIVE AI SALTI INCONDIZIONATI

Alcuni esempi consentono di chiarire quanto precedentemente trattato. Per calcolare  $(a \times b)^2 + c$ , le istruzioni saranno le seguenti:

Istruzione	Descrizione
A V	Destinazione iniziale: abbassando il tasto V il calcolatore ricerca la destinazione AV, posta in testa al programma, e da inizio all'elaborazione
S	Introduzione di a
↓	Trasferimento di a in A
S	Introduzione di b
x	$a \times b$
A x	$(a \times b)^2$

S                    Introduzione di c  
+                     $(a \times b)^2 + c$   
A ◇                Stampa del risultato contenuto in A  
V                    Salto di fine programma: il calcolatore ricerca la destinazione A V, riportandosi in tal modo all'inizio del programma per ripetere tutte le istruzioni che lo compongono, utilizzando i nuovi dati che verranno introdotti dall'operatore

Per calcolare la media aritmetica di un numero nota di dati, ad esempio 100

$$\frac{a_1 + a_2 + a_3 + \dots + a_{100}}{100}$$

occorre ripetere 100 volte una stessa serie di operazioni e accumulare i dati in un registro di deposito. Le istruzioni da ripetere sono:

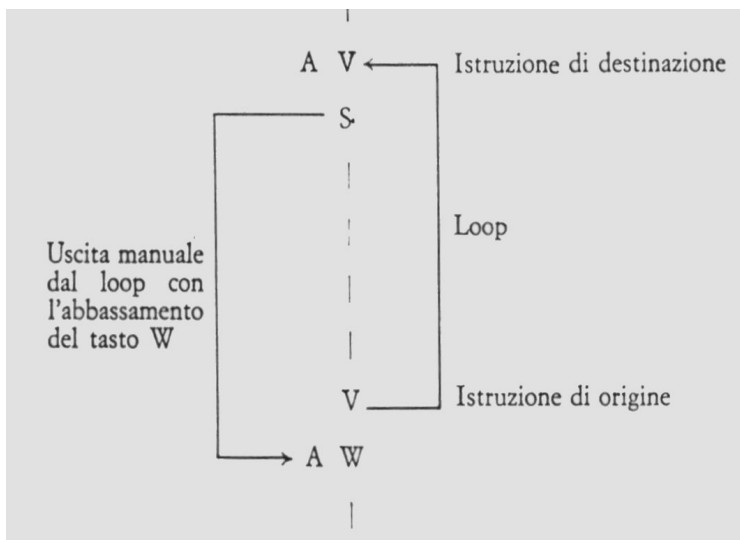
S        Impostazione di  $a_i$   
↓        Trasferimento di  $a_i$  in A  
B +     Accumulo di  $a_i$  sul totale precedente contenuto in B  
B ↑     Memorizzazione dell'accumulo.

(pag. 34)

La sequenza è ciclica ed è perciò opportuno ricorrere alle istruzioni di salto incondizionato al fine di programmarla una sola volta.

Si viene in tal modo a realizzare un loop all'interno del programma poiché una sua determinata sequenza viene più volte ripetuta prima di passare alle istruzioni successive.

Dopo l'introduzione dell'ultimo data la macchina esegue ancora l'istruzione di salto che chiude il ciclo e si arresta sullo Stop iniziale. Ciò avviene perchè la meccanica dei salti incondizionati è rigida e non comporta un'operazione di scelta da parte del calcolatore. E' perciò l'operatore che deve intervenire per comunicare al calcolatore di non ripetere ulteriormente il ciclo, non essendovi altri dati da elaborare, ordinandogli di eseguire invece le ulteriori istruzioni del programma.



A tal fine è indispensabile che le istruzioni successive al loop siano precedute da un'istruzione di destinazione per fare in modo che l'operatore, impostando l'istruzione di origine corrispondente, comandi al calcolatore di eseguire le istruzioni ad essa successive. Più chiaramente, quando la macchina si arresta sullo Stop dopo l'introduzione del centesimo dato, l'operatore deve abbassare uno dei tasti V, W, Y, Z, a seconda che l'istruzione di destinazione che segue il ciclo sia AV, AW, AY o AZ.

Il programma completo per risolvere il problema in esame e il seguente:

Istruzione	Descrizione
AV	Istruzione di destinazione di inizio programma
S	Impostazione di 100
C ↑	Memorizzazione di 100
B *	Azzeramento di B, registro in cui si fa l'accumulo
BV	Istruzione di destinazione della sequenza ciclica
S	Impostazione di $a_i$
↓	Trasferimento di $a_i$ in A
B +	Accumulo di $a_i$
B ↓	Memorizzazione dell'accumulo
C V	Istruzione di salto che chiude la sequenza ciclica
AW	Istruzione di destinazione per la sequenza finale
B ↓	Richiamo del totale in A
C :	Divisione del totale per 100
A ◇	Stampa della media
V	Istruzione di salto che chiude il programma

L'istruzione B\*, situata al quarto posto, è necessaria in quanto prima d'iniziare il loop il contenuto di B deve essere zero.

## Salti condizionati

La struttura delle istruzioni di salto condizionato è del tutto simile a quella dei salti incondizionati.

E' richiesta infatti, anche in questo caso, una coppia di istruzioni (l'origine e la destinazione) (pag. 35) per definire un salto, ma la differenza sostanziale sta nel fatto che, incontrata un'istruzione di origine, il calcolatore ricerca la destinazione corrispondente solo a certe condizioni.

Il calcolatore esamina il valore numerico contenuto in quell'istante nel registro A, e in base a tale valore opera una scelta tra due diverse sequenze di istruzioni:

se  $A > 0$  esegue il salto ricercando la destinazione corrispondente  
 se  $A \leq 0$  non tiene conto dell'istruzione di salto e continua l'esame sequenziale delle istruzioni successive.

Le due istruzioni che definiscono un salto condizionato sono ovviamente legate tra loro, dato che a ogni istruzione di origine corrisponde un'unica istruzione di destinazione.

Le coppie di istruzioni di salto condizionato di cui si può disporre sono 16:

Istruzione di origine	Istruzione di destinazione
/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

Più semplicemente, esse possono essere rappresentate come segue:

Origine del salto	Destinazione del salto
/Δ	A /Δ
C /Δ	B /Δ
D /Δ	E /Δ
R /Δ	F /Δ

dove: Δ = V, W, Y, Z

Con le istruzioni di salto condizionato, il calcolatore acquista una capacità discriminante automatica, in quanto può scegliere tra due possibili sequenze del programma a seconda del contenuto del registro A.

Per chiarire meglio quanto asserito, valga ancora l'esempio sopra esposto, a illustrazione delle applicazioni relative ai salti incondizionati.

Qualora si fosse creato un contatore, aggiornato mediante l'aggiunta di una unità a ogni ciclo, la ripetizione del ciclo si sarebbe potuta automaticamente arrestare quando il contatore avesse raggiunto il valore 100.

La sequenza ciclica avrebbe dunque dovuto:

- *accumulare  $a_i$*
- *aggiornare il contatore*
- *sottrarre da 100 il valore del contatore.*

In tal modo, programmando alla fine di essa un salto condizionato, si sarebbe potuta interrompere la sua ripetizione quando il contatore avesse raggiunto il valore di 100.

(pag. 36)

Il programma risulterebbe quindi così modificato:

Istruzione	Descrizione
A V	Istruzione di destinazione iniziale
S	Impostazione di 1
D ↑	Memorizzazione di 1 in D
S	Impostazione di 100
C ↑	Memorizzazione di 100 in C
B *	Azzeramento di B, ove si forma l'accumulo
C /*	Azzeramento di c, ove si forma il contatore
B / V	Destinazione del salto condizionato
S	Impostazione di $a_i$
↓	Trasferimento di $a_i$ in A
B +	Accumulo di $a_i$ sul totale precedente
B ↓	Memorizzazione dell'accumulo
D ↓	Richiamo di 1 in A
C / +	Valore del contatore + 1
C / ↓	Accantonamento del nuovo valore
C ↓	Richiamo di 100 in A
C / -	100 - valore del contatore
C / V	Origine di salto condizionato: il salto si realizza se il risultato della sottrazione è positivo
B ↓	Richiamo del totale
C :	Divisione del totale per 100
A ◇	Stampa della media
V	Istruzione di salto incondizionato che chiude il programma

Come appare da questo esempio il calcolatore conta il numero di cicli da eseguire e, raggiuntolo, trascura l'istruzione di salto condizionato, portandosi invece sulle ulteriori istruzioni del programma.

Nel caso esaminato la condizione  $A > 0$  è stata generalizzata con quella di  $A > x$ . Infatti la fine della sequenza ciclica è stata determinata dal confronto del contatore con 100. Quindi per esaminare la condizione  $A > x$  (ove  $x$  è un dato contenuto in un registro numerico) basta sottrarre  $x$  dal contenuto di  $A$  prima di programmare l'istruzione di salto condizionato.

## DISCRIMINAZIONE FRA TRE CONDIZIONI

Un salto condizionato rende possibile la scelta tra due sequenze di programma, in funzione del contenuto del registro  $A$ .

Qualora si voglia distinguere se il contenuto di  $A$  è  $<$  oppure  $= 0$  occorrerà esaminare se sono soddisfatte non due ma tre condizioni scindendo il caso  $\leq 0$  in due casi distinti: minore ed eguale.

Ciò consente di stabilire se un certo risultato è: *maggiore di zero (positivo); eguale a zero (nullo); minore di zero (negativo)*.

È chiaro che programmando un salto condizionato dopo aver trasferito in  $A$  il dato che interessa (se già non vi si trova), il primo caso risulta facilmente individuato in quanto il salto ha luogo; nel secondo e nel terzo caso il programma continua normalmente. Per differenziarli occorre allora programmare, dopo l'istruzione di origine, quella di valore assoluto ( $A \uparrow$ ) che rende positivo il contenuto del registro  $A$ . Evidentemente se questo è zero non si ha alcuna modifica del contenuto stesso.

Programmando un nuovo salto condizionato, il contenuto di  $A$  viene riesaminato e, se risulta positivo, il salto ha luogo (indicando che il primitivo valore era negativo). Il programma continua invece alla successiva istruzione se il contenuto originario è nullo.

La sequenza di istruzioni necessarie per rendere possibile una discriminazione tra le tre condizioni indicate è dunque del tipo:

C/W (o altra istruzione di salto condizionato)

$A \uparrow$

R/W [o altra istruzione qualsiasi ( $\neq$  C/W) di salto condizionato.

(pag. 37)

L'esempio che segue varrà meglio a chiarire quanto esposto.

Sia  $x$  un risultato sul quale si devono svolgere determinate operazioni in base al confronto fra i due numeri «a» e «b».

Precisamente:

se  $a > b$  si deve calcolare  $x * z$

se  $a < b$  si deve calcolare  $x - z$

se  $a = b$  si deve calcolare  $x + z$

Le istruzioni da programmare saranno:

Istruzione	Descrizione
A W	Istruzione di destinazione iniziale
S	Impostazione di x
B ↑	Memorizzazione di x in B
S	Impostazione di a
↓	a nel registro A
S	Impostazione di b
-	(a-b)
/Y	1 <sup>a</sup> istruzione di salto condizionato che separa i casi $(a - b) > 0$ e $(a - b) \leq 0$
A ↑	Valore assoluto di (a-b)
/Z	2 <sup>a</sup> istruzione di salto condizionato che separa i casi $(a - b) < 0$ e $(a - b) = 0$
<u>caso <math>(a-b)=0</math></u>	
B ↓	$x \rightarrow A$
S	Impostazione di z
+	$x + z$ in A
A ◇	Stampa del risultato
W	Salto incondizionato di fine programma
A/Z	Destinazione del 2° salto condizionato
<u>caso <math>(a-b)&lt;0</math></u>	
B ↓	$x \rightarrow A$
S	Impostazione di z
-	$x - z$ in A
A ◇	Stampa del risultato
W	Salto incondizionato di fine programma
A/Y	Destinazione del 1° salto condizionato
<u>caso <math>(a-b)&gt;0</math></u>	
B ↓	$x \rightarrow A$
S	Impostazione di z
x	$x \cdot z$ in A
A ◇	Stampa del risultato
W	Salto incondizionato di fine programma

Dopo che l'operatore ha abbassato il tasto «W», la macchina esegue le istruzioni che trova dopo AW, le quali calcolano la differenza tra i due numeri «a» e «b» (analizzando non  $a <=> b$ , ma  $(a - b) <=> 0$ ).

L'istruzione /Y fa sì che il calcolatore esegua il confronto tra zero e quanto si trova in A. Di conseguenza:

- se  $a > b$ : in A c'è un valore positivo, si ricerca quindi la destinazione corrispondente A/Y e vengono eseguite le istruzioni che seguono
- se  $a < b$ : in A c'è un valore negativo, quindi si eseguono le istruzioni successive /Y. Il calcolatore determina il valore assoluto del contenuto di A e incontrando /Z esegue un nuovo confronto del contenuto di A con zero. Il valore questa volta è positivo poiché si è preso in considerazione un valore assoluto; il calcolatore ricerca quindi l'istruzione A/Z
- se  $a = b$ : in A c'è un valore nullo. In questo caso il calcolatore giunge, passo a passo, alla istruzione /Z, non esegue il salto, essendo  $a = b$ ,  $(a - b) = 0$ , e passa alle istruzioni che seguono /Z.

Alla fine di ciascuno dei sottoprogrammi alternativi sopra esaminati è necessario, per tenere separate le varie sequenze, programmare uno Stop o un'istruzione di origine di salto in condizionato per comandare l'arresto o il ritorno all'inizio del programma.

## Dati costanti

La registrazione di un programma su scheda magnetica comporta il trasferimento dell'intero contenuto dei registri 1, 2, F, E, D, sulla medesima; viceversa, la lettura di una scheda trasferisce in memoria, nei registri suddetti, quanto in essa si trova.

E' possibile quindi registrare su scheda anche dati numerici, in particolare costanti che intervengano più volte nel corso di un programma. La registrazione di costanti su scheda magnetica si può eseguire in due modi:

- costanti nei registri di deposito
- codificazione di costanti come istruzioni

### Costanti nei registri di deposito

I dati numerici introdotti nei registri D, d, E, e, F, f, possono essere direttamente registrati su scheda magnetica, insieme alle istruzioni di programma.

E' ovvio che si deve prevedere all'atto della stesura del programma di non impegnare per istruzioni i registri destinati alle costanti.

Ricordando che il flusso delle istruzioni procede da F verso D, e opportuno, al fine di evitare errori, prevedere di immagazzinare le costanti secondo un flusso di senso opposto, cioè partendo da d, D, passando successivamente, se necessario, a E ed F.

Le costanti, introdotte nei registri prestabiliti, sono registrate direttamente su scheda insieme con le istruzioni del programma.

Le operazioni da eseguire sono le seguenti:

1. premere il pulsante «registrazione programma» (Reg. Progr.)
2. registrare il programma nel calcolatore seguendo le modalità previste
3. premere nuovamente il pulsante «registrazione programma» per riportarlo in posizione di riposo
4. impostare la costante sulla tastiera
5. premere il tasto del registro in cui si vuole introdurre la costante e quindi il tasto «↑»
6. premere nuovamente il pulsante «registrazione programma»
7. inserire la scheda magnetica nel calcolatore per registrare le istruzioni e le costanti.

In fase di lettura della scheda gli stessi dati vengono nuovamente trasferiti nei registri originali, sicché nel corso del programma possano essere reperiti ove si era stabilito di posizionarli. E' quindi possibile registrare 3 dati di 22 cifre, 6 dati di 11 cifre, oppure ogni combinazione alternativa.

Questo metodo si usa generalmente quando i dati da registrare abbiano molte cifre e/o intervengano più volte nel corso del programma.

### Codificazione di costanti come istruzioni

Questo metodo consente di registrare dati numerici sotto forma di istruzioni di programma. Ogni cifra è rappresentata da un'istruzione. Queste istruzioni (compilate automaticamente nel modo descritto più avanti) sono precedute nel programma dall'istruzione A/↑ che serve a qualificarle come dati numerici, anziché come funzioni operative.

Quando il programma viene eseguito numerico e formato nel registro M il dato e può (pag. 39) quindi essere direttamente utilizzato per le successive operazioni previste dal programma.

Questo metodo è generalmente usata per la registrazione di dati numerici composti di poche cifre e/o da utilizzare una sola volta nel corso dello stesso programma.

Per la compilazione automatica delle istruzioni corrispondenti ai dati numerici da registrare, si opera come segue:

1. inserire il pulsante «stampa programma»
2. premere l'annullatore generale
3. impostare il dato nella tastiera numerica
4. premere nell'ordine i tasti «A », «↑».

Il calcolatore stampa il dato seguito dalla sequenza di istruzioni corrispondente.  
 Il segno e la virgola non richiedono ulteriori istruzioni in quanto vengono espressi nell'ambito delle cifre della costante.

#### Esempio

Impostando 3,14	A ↑
il calcolatore stampa	R +
	R ↓
	d ↑

La sequenza di istruzioni così compilata va inserita nel programma, alla posizione prevista, preceduta ogni volta dall'istruzione A /↑.  
 Per chiarire quanto esposto, si voglia calcolare la circonferenza di un cerchio di raggio  $r = 5$ , utilizzando la nota formula  $2\pi r$  e codificando sia il valore di  $\pi (3,14)$  sia 2.

Le istruzioni da programmare saranno:

decimali → 1

Istruzione	M	A	Descrizione
A V			Riferimento iniziale
S	5		Introduzione del raggio $r=5$
↓	5	5	$5 \rightarrow A$
A / ↑			Segnalazione di costante codificata
R +			Istruzioni corrispondenti
R ↓			alla costante codificata
D / ↑			3,14
x	3,14	15,7	$5 \times 3,14$
A / ↑			Segnalazione di costante codificata
D / ↑	2	15,7	Istruzione corrispondente alla costante 2
x	2	31,4	$5 \times 3,14 \times 2$
A ◇		31,4	Stampa del risultato
V			Salto di fine programma

Per la codifica manuale delle costanti da inserire in programma, senza ricorrere all'utilizzazione del calcolatore, vedi la tabella posta alla fine del volume.



## Tecniche di programmazione

### Inversione di segno di un dato

L'inversione di segno di un dato numerico si ottiene con la trasformazione del dato da positivo in negativo o da negativo in positivo.  
Se il dato è contenuto in A, l'inversione si realizza nel modo seguente:

Istruzione	Descrizione
A-	Azzeramento del contenuto di A
-	Esecuzione della sottrazione tra A e M (dove si trova il precedente valore di A) ottenendo come risultato l'inversione richiesta

### Reciproco di un dato

Supposto che il dato X ( $\neq 0$ ) sia contenuto nel registro A, il suo reciproco  $1/x$  si ottiene nel modo seguente:

Istruzione	Descrizione
A :	1 in A ed x in M
:	Esecuzione della divisione $1 : x$ , ottenendo così il reciproco

### CALCOLO DI $1 \pm x$

Si suppone, anche in questo caso, che il dato x ( $\neq 0$ ) sia contenuto nel registro A e si programma quanto segue:

Istruzione	Descrizione
A :	1 in A ed x in M
+ (o -)	A + M (o A - M)

### Generazione di costanti

Nel capitolo precedente sono stati descritti due metodi mediante i quali è possibile inserire in programma dati numerici costanti.

Alcune volte però si possono creare automaticamente delle costanti con opportuni accorgimenti, soprattutto quando il valore dei dati numerici non è elevato. Ad esempio:

- per ottenere 1, basta avere nel registro A un numero diverso da zero e programmare l'istruzione A :
- per avere 2 in A, si programma l'istruzione A : seguita dall'istruzione A +
- per avere  $1/2$  in A, si programma un'operazione di radice seguita dall'istruzione : (diviso).

Queste costanti, così generate, possono essere successivamente e variamente sommate al fine di ottenere altre costanti di valore non elevato.

## Arrotondamento all'unità più vicina

Accade spesso, in problemi di vario tipo, di dover eseguire arrotondamenti su determinate cifre in relazione alla precisione con cui si vuole operare.

Di solito, quando si stabilisce di eseguire un calcolo con un numero di cifre decimali prefissato, si esamina la prima cifra scartata; se questa è 5, o maggiore di 5, si aumenta di 1 il valore dell'ultima cifra non scartata.

Ad esempio, si abbia 51,3278 e si voglia operare con 3 decimali: in tal caso per effetto dell'arrotondamento il dato assume il valore 51,328 poiché la prima cifra scartata, 8, è maggiore di 5. Con il calcolatore Programma 101 gli arrotondamenti si possono ottenere assai semplicemente utilizzando il risultato completo che si trova nel registro R.

Le istruzioni con cui è possibile realizzare l'arrotondamento all'unità più vicina, nel risultato di un'addizione, sottrazione o moltiplicazione, sono le seguenti: .

Istruzione	Descrizione
R -	Sottrazione del risultato completo (contenuto in R) da quello troncato (contenuto in A). In A si ottiene zero; in R le cifre decimali scartate nella operazione precedente, con il segno meno; in M il risultato completo della precedente operazione
↓	Trasferimento in A del contenuto di M
R -	Sottrazione del risultato completo di partenza dal contenuto attuale di R per ottenere l'eventuale arrotondamento

Si voglia, ad esempio, arrotondare 5,1327 con l'indicatore dei decimali posizionato a 3. Le istruzioni da programmare saranno:

Istruzione	M	A	R
		5,132	5,1327
R-	5,1327	-0,000	-0,0007
↓	5,1327	5,1327	
R -	- 0,0007	5,133	5,1334

Nel risultato finale si ha l'arrotondamento all'unità superiore, perchè la prima cifra scartata è 7. Per arrotondare invece 5,1324 con il medesimo numero di decimali, le istruzioni saranno:

Istruzione	M	A	R
		5,132	5,1324
R -	5,1324	-0,000	-0,0004
↓	5,1324	5,1324	
R -	- 0,0004	5,132	5,1328

In questo caso nel risultato finale si ha l'arrotondamento all'unità inferiore perchè la prima cifra scartata è 4.

Nel caso di operazioni aritmetiche concatenate, se si vuole ottenere un maggior grado di precisione, è opportuno inserire la routine di arrotondamento dopo ogni operazione, in particolare dopo la moltiplicazione. (pag. 42)

Ovviamente, dopo la divisione e la radice quadrata, questa sequenza non può essere applicata poiché in R si forma il resto nel primo caso, un dato non significativo nel secondo.

L'arrotondamento dopo la divisione si ottiene perciò con la sequenza seguente (supponendo che il registro B contenga il dividendo e C il divisore):

Istruzione	Descrizione
R ↓	Trasferimento in A del resto aritmeticamente corretto della divisione
B +	Somma del resto al dividendo che si suppone contenuto nel registro B completo della precedente operazione
R ↓	Trasferimento in risultato completo della somma in A
C :	Divisione del risultato precedente (dividendo + resto) per il divisore, ottenendo il quoziente arrotondato.

Si voglia calcolare  $315 : 8$  con quoziente arrotondato e indicatore decimale posizionato a 2. Le istruzioni da programmare saranno:

Istruzione	M	A	R	B	C
				315	8
B ↓		315			
C :	8	39,37	0,04		
R ↓	8	0,04			
B +	315	315,04	315,04		
R ↓	315	315,04			
C :	8	39,38	0,00		

In tal caso si ha un arrotondamento per eccesso poiché la prima cifra scartata nel quoziente è 5 ( $315 : 8 = 39,375$ )

Volendo ora eseguire la stessa divisione con l'indicatore dei decimali posizionato a zero, si ottiene un arrotondamento per difetto del quoziente, perché la prima cifra scartata è 3.

Istruzione	M	A	R	B	C
				315	8
B ↓		315			
C :	8	39	3		
R ↓	8	3			
B +	315	318	318		
R ↓	315	318			
C :	8	39	6		

## Arrotondamento per eccesso

Si è visto che l'arrotondamento all'unità più vicina si traduce in pratica in un arrotondamento per eccesso o per difetto a seconda del valore della prima cifra decimale che si scarta. Se invece, dopo un'operazione di addizione, sottrazione o moltiplicazione, si deve eseguire un arrotondamento per eccesso, è necessario programmare questa sequenza di istruzioni:

Istruzione	Descrizione
R +	Somma del risultato troncato a quello completo contenuto in R
↓	Scambio del contenuto di A con quello di M
-	Differenza A- M
+	Risultato della somma con l'arrotondamento desiderato

## Esempio

Istruzione	M	A	R
		12,00	12,001
R +	12,001	24,00	24,001
↓	24,00	12,001	24,001
-	24,00	-11,99	-11,999
+	24,00	12,01	12,010

## Arrotondamento a valori prestabiliti

Si richiede tal volta l'arrotondamento di un risultato a 5, 10, 100, ecc., o ad altri valori prestabiliti.

Per ottenere ciò dopo un'operazione di addizione, sottrazione o moltiplicazione, basta memorizzare in un registro di deposito (ad esempio d) un'opportuna costante e programmare questa sequenza di istruzioni:

D/x  
R -  
↓  
R -  
D/:

Contenuto di d	Arrotondamento
0,2	a 5
0,1	a 10
0,01	a 100
.....	.....

Si osserva che l'arrotondamento opera sulla cifra meno significativa per qualsiasi posizione dell' indicatore dei decimali.

Nella tabella seguente è illustrato un esempio di arrotondamento a 5, con l'indicatore dei decimali a zero.

Istruzione	M	A	R	d
		153		0,2
D/x	0,2	30	30,6	
R -	30,6	-0	- 0,6	
↓	30,6	30,6	- 0,6	
R -	- 0,6	31	31,2	
D/:	0,2	155	0,0	

## Programmazione di un contatore

Particolari istruzioni permettono al calcolatore di formare automaticamente un contatore, ossia una quantità il cui valore viene costantemente incrementato di una unità per contare quante volte un certo ciclo viene ripetuto.

Le istruzioni che permettono di aggiornare il valore di un contatore devono evidentemente essere inserite nella sequenza ciclica. Si deve invece provvedere prima di essa all'azzeramento del registro in cui il contatore si deve formare.

Supponiamo ad esempio che il registro diviso B/ sia utilizzato per il contatore e che nel registro A si trovi un dato diverso da zero; l'aggiornamento si ottiene con le seguenti istruzioni:

Istruzione	Descrizione
A :	Generazione di 1 che consente di incrementare il contatore di una unità per ogni ciclo
B / +	Incremento del contatore
B / ↓	Memorizzazione del nuovo valore del contatore

(pag. 44)

### Interruzione automatica di sequenza ciclica

Quando è noto a priori il numero di volte che deve essere eseguita una sequenza ciclica, si può memorizzare in un registro questo numero e sottrarre ad esso una unità ad ogni esecuzione del ciclo; quando si ottiene zero come risultato della sottrazione, la sequenza ciclica viene interrotta.

Il contatore può anche servire per interrompere automaticamente il ciclo, funzionando per successivi incrementi o decrementi di una unità ad ogni ciclo, fino a raggiungere il numero prestabilito (nel primo caso) o lo zero (nel secondo caso).

Supponendo che prima dell'inizio della sequenza ciclica il valore massimo del contatore sia già stato registrato in B, le istruzioni per l'aggiornamento sono:

Istruzione	M	A	B	Descrizione
			valore massimo contatore	
B ↓		contatore		Trasferimento del valore del contatore in A
A :	contatore	1		1 in A
↓	1	contatore		Scambio del contenuto di A con M
-	1	contatore - 1		Aggiornamento del valore del contatore
B ↓		contatore	contatore - 1	Memorizzazione in B del nuovo valore del contatore

Facendo seguire alla sequenza esaminata l'istruzione B↓ e un'istruzione di salto condizionato che riporti il calcolatore all'inizio del loop, si otterrà l'interruzione automatica del ciclo quando il valore di B sarà eguale a zero.

### Registrazione di più dati in un unico registro

Quando si hanno parecchi dati di poche cifre e si presenta la necessità di economizzare posizioni di memoria, si può sfruttare al massimo la capacità dei registri collocando più dati in uno stesso registro.

In tal caso però le cifre contenute nello stesso registro non potranno essere separate e costituiranno un unico numero.

Ad esempio, la registrazione in B dei dati seguenti:

234 329 150 215

si ottiene memorizzando il numero:

Per operare sui singoli dati occorrerà ovviamente separarli. Ciò è possibile dividendo il numero per 10, 100, 1000 o successive potenze di 10, a seconda del numero di cifre che si desidera via via isolare. Nell'esempio ora esposto, trattandosi di dati di 3 cifre ciascuno, occorrerà dividere il numero complessivo per 1000 ( $10^3$ ) per quattro volte consecutive. Dopo ogni divisione (con l'indicatore dei decimali a zero), il numero che si desidera isolare si troverà nel registro R come resto della divisione.

Perché questa tecnica sia conveniente, occorre che tutti i dati numerici che si desidera collocare nel medesimo registro abbiano eguale numero di cifre. In tal caso basterà utilizzare sempre lo stesso divisore per separarli. In caso contrario si dovrà rendere eguale il numero di cifre di ciascun dato aggiungendo degli zeri alla sinistra dei dati con meno cifre, fino a renderli tutti della stessa lunghezza.

(pag. 45)

### Uso promiscuo dei registri F, E, D, per dati e istruzioni

Quando nel corso di un programma che preveda di utilizzare come numerici i registri F, E, D, sia necessario disporre di oltre 48 istruzioni, è possibile impiegare una particolare tecnica di programmazione che consente l'uso promiscuo dei registri F, E, D, per dati e istruzioni, sfruttando le posizioni di ogni registro che non vengono occupate dai dati numerici.

Se si vuole utilizzare promiscuamente un registro intero o la parte destra di un registro diviso (F, E, D) occorre programmare, a partire dalla prima posizione del registro, tante S quante sono le cifre che compongono il dato, più una come segnale di fine numero.

Si dovrà poi programmare un'altra S, come segnale di inizio dato, alla 24<sup>a</sup> posizione del registro. Ciò perché il calcolatore legge circolarmente i registri e quindi la 24<sup>a</sup> posizione viene a precedere immediatamente la prima.

Se si vuole utilizzare promiscuamente la parte sinistra di un registro diviso (f, e, d) occorre programmare una S alla 12<sup>a</sup> posizione del registro (cioè alla prima del registro diviso), come segnale di inizio dato, e tante S quante sono le cifre del numero, più una come segnale di fine dato.

Le sequenze di Stop, programmate secondo la tecnica indicata, devono essere precedute da un'istruzione di origine di salto incondizionato e terminare con l'istruzione di destinazione corrispondente.

Ciò è necessario per collegare fra loro le istruzioni e per isolare quelle posizioni di memoria che dovranno conservare dati numerici.

Se per questo scopo vengono utilizzate le istruzioni di salto incondizionato V, W, Y, Z, non occorre programmare lo Stop di fine numero.

Nel caso in cui venga utilizzata promiscuamente sia la prima sia la seconda parte di registro diviso, occorrerà tener presente entrambe le tecniche esposte.

Si debbano ad esempio calcolare i parametri della retta di regressione e del coefficiente di correlazione lineare secondo le formule seguenti:

$$r = \frac{N \cdot \Sigma xy - (\Sigma x) \cdot (\Sigma y)}{\sqrt{[N \cdot \Sigma x^2 - (\Sigma x)^2] \cdot [N \cdot \Sigma y^2 - (\Sigma y)^2]}}$$

$$b = \frac{N \cdot \Sigma xy - (\Sigma x) \cdot (\Sigma y)}{N \cdot \Sigma x^2 - (\Sigma x)^2}$$

$$a = \frac{\Sigma y - b \cdot \Sigma x}{N}$$

dove:

- r = coefficiente di correlazione
- b = coefficiente angolare
- a = ordinata all'origine
- x, y = dati variabili
- N = numero delle coppie di osservazioni.

Le istruzioni da programmare saranno quelle indicate nella tabella della pagina seguente; la procedura operativa sarà:

1. abbassare il tasto «V»
2. impostare la variabile  $x_1$  (sul 1° S)
3. impostare la variabile  $y_1$  (sul 2° S)
4. ripetere le operazioni 2 e 3 per tutte le coppie di variabili
5. abbassare il tasto «Z»
6. il calcolatore stampa nell'ordine i risultati «r», «b», «a».

I registri F ed E sono usati in modo promiscuo, per la memorizzazione di dati (di capacità massima di 15 cifre) e istruzioni.

Il registro d è usato per la memorizzazione di un dato di 3 cifre (N).

Si noti come C debba essere usato come registro intero per conservare la  $\Sigma xy$  che può superare le 11 cifre.

ISTRUZIONI DI PROGRAMMA															SCHEDA N°.....				
REG. 1			REG. 2			REG. F			REG. E			REG. D							
1	A	v	25	A	x	49		s	73		s	97	F	w					
2	F	*	26		↓	50		s	74		s	98	E	↓					
3	E	*	27	E	↓	51		s	75		s	99	A	√					
4	D	/	28		+	52		s	76		s	100		↓					
5	C	*	29	E	↓	53		s	77		s	101	C	↓					
6	B	/	30	B	+	54		s	78		s	102		:					
7	B	*	31	B	↓	55		s	79		s	103	E	:					
8	A	w	32	A	/	↑	56		s	80		s	104	A	◊				
9		s	33	D	/	↓	57		s	81		s	105	C	↓				
10	B	/	↓	34	.	↓	58		s	82		s	106	F	:				
11		+	35	D	/	+	59		s	83		s	107	R	Y				
12	B	/	↓	36	D	/	↓	60		s	84		s	108		s			
13		↓	37		w	61		s	85		s	109			s				
14		x	38	A	Z	62		s	86		s	110			s				
15		↓	39	B	/	↓	63		s	87		s	111		s				
16	F	↓	40	B	x	64		s	88		s	112			s				
17		+	41	C	↓	65	F	v	89	F	Z	113	F	Y					
18	F	↓	42	D	/	x	66	D	/	x	90	E	↓	114	A	◊			
19		s	43	C	-	67	F	-	91	D	/	x	115	B	/	x			
20		x	44	C	↓	68	F	↓	92	E	-	116	B	↓					
21		↓	45	B	/	↓	69	B	↓	93	E	↓	117	B	-				
22	C	↓	46	A	x	70	A	x	94	F	√	118	D	/	:				
23		+	47	F	↓	71	R	Z	95	R	w	119	A		◊				
24	C	↓	48	R	v	72		s	96		s	120			v				
REG. 1			REG. 2			REG. f/			REG. e/			REG. d/							

CONTENUTO REGISTRI	
M	
A	
R	
b/	$\Sigma x$
B	$\Sigma y$
c/	
C	$\Sigma xy$
d/	N + ISTRUZIONI
D	ISTRUZIONI
e/	
E	$\Sigma y^2 + ISTRUZIONI$
f/	
F	$\Sigma x^2 + ISTRUZIONI$

## Lunghezza dei dati e decimali nei risultati

Poiché, come è noto, la capacità massima dei registri è di 22 cifre, occorre tener presente che il risultato di qualsiasi operazione non potrà superare tale numero di cifre. In addizione e moltiplicazione si può avere un risultato di 23 cifre che però non è ulteriormente operabile. È opportuno quindi esaminare la compatibilità tra la lunghezza dei dati e il numero di decimali richiesti nelle varie operazioni.

### ADDIZIONE E SOTTRAZIONE

Queste due operazioni non presentano problemi dal punto di vista della capacità, in quanto basta ricordare che gli operandi subiscono un incolonnamento preliminare prima che l'operazione venga eseguita e che l'eventuale scarto di cifre, predisposto con l'indicatore dei decimali, avviene soltanto dopo che il risultato si è formato nel registro A.

*È quindi necessario che il numero di cifre intere dell'operando avente un maggior numero di cifre intere, sommato al numero di cifre decimali dell'operando avente un maggior numero di cifre decimali, non superi 22.*

#### Esempi

$$\begin{array}{r} \text{a)} \quad 432100118,467 \\ \quad \quad 325,123456789123456 \end{array} \quad +$$

il primo operando ha 12 cifre di cui 9 intere e 3 decimali, il secondo operando 18 cifre di cui 3 intere e 15 decimali, per cui

$$9 + 15 = 24$$

L'operazione non si può dunque eseguire poiché il risultato supererebbe la capacità dell'accumulatore, pur avendo i singoli operandi una lunghezza accettabile.

La segnalazione di supero di capacità si ha indipendentemente dal posizionamento dell'indicatore dei decimali per il motivo prima enunciato.

$$\begin{array}{r} \text{b)} \quad 432100118,467 \\ \quad \quad 3251234,56789123456 \end{array} \quad +$$

L'operazione è invece, in tal caso, effettuabile poiché il primo operando ha 9 interi e 3 decimali, il secondo 7 interi e 11 decimali. Si ha infatti:

$$9 + 11 = 20$$

rispettando così la condizione richiesta.

### MOLTIPLICAZIONE

Il risultato di una moltiplicazione può avere al massimo un numero di cifre pari alla somma delle cifre dei due fattori.

È possibile perciò determinare se la lunghezza del risultato è compatibile con la capacità dell'accumulatore, tenendo presente che la somma del numero di cifre del moltiplicando e del moltiplicatore non deve superare 22.

Quanto sopra è valido indipendentemente dal posizionamento dell'indicatore dei decimali poiché, come già per l'addizione e la sottrazione, anche nel caso di moltiplicazione il risultato viene troncato solo a operazione avvenuta.



Così l'operazione

$$\begin{array}{r} 345678,7876589459 \times \\ 765,7438 \end{array}$$

non può essere eseguita dal calcolatore pur posizionando a zero l'indicatore dei decimali. Analogamente a quanto avviene per l'addizione, il prodotto di due fattori può avere eccezionalmente (pag. 48) 23 cifre. In questo caso però il risultato non è più operabile.

Qualora invece il numero di decimali richiesto sia superiore alla somma dei decimali dei singoli operandi, occorre tener presente che la somma del numero di cifre intere del moltiplicando e del moltiplicatore, nonché dei decimali richiesti, deve essere  $\leq 22$ .

Quindi, generalizzando, si può dire che:

*la somma del numero di cifre intere dei fattori e del maggiore tra il numero di decimali richiesti e la somma dei decimali degli operandi non deve superare 22.*

## DIVISIONE

*Nel caso della divisione il numero dei decimali che si può richiedere è data dalla differenza tra 22 e la somma delle cifre intere del dividendo e dei decimali del divisore.*

Questa regola è valida con dividendo o divisore sia interi che decimali e qualunque sia la relazione dell'uno rispetto all'altro (dividendo  $\leq$  divisore), purché il divisore abbia almeno una cifra intera significativa.

### Esempi

a) dividendo di 10 cifre intere e divisore di 5. In tal caso basta calcolare:  $22 - 10 = 12$ . L'indicatore dei decimali potrà dunque essere posizionato al massimo a 12

b) dividendo di 8 interi e 5 decimali, divisore di 10 interi:  $22 - 8 = 14$

c) dividendo di 6 interi e 3 decimali, divisore di 4 interi e 4 decimali:  $22 - (6 + 4) = 12$

d) dividendo di 6 interi e 3 decimali, divisore di 8 interi e 2 decimali:  $22 - (6 + 2) = 14$ .

Qualora il divisore non abbia invece cifre intere significative (sia cioè 0, . . . . .), il massimo numero di decimali che può essere richiesto per il quoziente è eguale a:

*$22 - (\text{numero delle cifre intere del dividendo} + \text{numero delle cifre decimali del divisore} + \text{numero degli zeri che precedono la prima cifra significativa del divisore})$ .*

Occorre dunque sottrarre al numero di decimali, determinato con la regola precedentemente enunciata, tante unità quanti sono gli zeri (compreso quello che precede la virgola) che stanno a sinistra della prima cifra significativa del divisore.

### Esempi

a)  $1234,32 : 0,003$   
 $22 - (4 + 3 + 3) = 12$

L'indicatore dei decimali potrà dunque essere posizionato a 12

b)  $123456,32 : 0,35$   
 $22 - (6 + 2 + 1) = 13$

Se il numero di decimali che si può richiedere risulta superiore a 15, si considera ovviamente 15 come massimo numero di decimali ottenibile poiché questa rappresenta il limite imposto dal calcolatore.

#### RADICE QUADRATA

Occorre tener presente che il calcolatore modifica la composizione decimale del radicando, analogamente a quanta avviene per il dividendo nel caso della divisione.

Infatti, prima che sia eseguita l'operazione di radice, il radicando viene modificato in modo che il numero dei suoi decimali risulti sempre eguale al numero di decimali richiesti per il risultato.

Il numero dei decimali che si può richiedere è dato da:

$22 - (\text{numero delle cifre intere del radicando} - 1).$
--

## Programmi particolari

### Programma su più schede

Per risolvere problemi complessi, che richiedono l'esecuzione di complicate o lunghe sequenze di calcoli con oltre 120 istruzioni, occorre spezzare il programma in parti, ognuna delle quali, preposta all'esecuzione di una particolare sequenza, viene registrata in memoria dopo aver ultimato la parte precedente. Ciascuna parte viene registrata su una scheda magnetica. I risultati intermedi che si desidera man tenere nel passaggio tra una scheda e la successiva, possono essere memorizzati nei registri M, A, R, b, B, c, C, che non sono interessati nella lettura della scheda magnetica.

Qualora oltre al contenuto dei registri elencati interessi conservare anche il contenuto dei registri d e D, occorre programmare prima del passaggio alla scheda successiva l'istruzione speciale RS.

### Istruzione RS

Questa istruzione comanda il trasferimento del contenuto del registro D in R e viceversa; essa ha caratteristiche speciali e può essere utilizzata nei seguenti casi:

*Programmi registrati su più schede*, per conservare temporaneamente nel registro R dati o risultati parziali contenuti nel registro D (D, d). Tale registro infatti, unitamente ai registri E ed F, viene automaticamente azzerato non appena ha inizio la lettura di una scheda. Con l'istruzione RS viene scambiato il contenuto dell'intero registro D, anche se diviso in D e d, con quello del registro R, rendendo quindi possibile, nel passaggio da una scheda all'altra, la conservazione di dati e/o risultati parziali nel registro R, oltre che in M, A, B, C.

L'istruzione RS è posta alla fine della prima parte del programma e, per ripristinare la situazione originale, all'inizio della successiva.

Tra la prima e la seconda istruzione RS non si può eseguire alcuna funzione aritmetica, di stampa o di trasferimento, che interessi i registri R e D.

Si debba ad esempio calcolare in base a certi dati di partenza il valore di 8 funzioni  $x_1, x_2, \dots, x_8$ , che devono poi essere moltiplicate rispettivamente per  $y_1, y_2, \dots, y_8$ , ove le  $y$  rappresentano i risultati di calcoli successivi. Si supponga di aver organizzato il programma su due schede: la sequenza registrata sulla prima fornisce i valori delle  $x$ , mentre con la seconda scheda si calcolano le  $y$  e si eseguono i prodotti finali  $x_i \cdot y_i$  ( $i = 1, 2, 3, \dots, 8$ ).

Si organizzerà il calcolo della  $x$  in modo che la situazione dei registri alla fine del programma della prima scheda sia quella qui sotto indicata e si utilizzerà quindi l'istruzione RS:

Istruzione	M	A	R	b	B	c	C	d	D
AV									
.									
.									
.	$x_1$	$x_2$	.....	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
RS	$x_1$	$x_2$	$x_7 x_8$	$x_3$	$x_4$	$x_5$	$x_6$		

L'introduzione della seconda scheda non altera M, A, R, b, B, c, C, per cui all'inizio del programma corrispondente si avrà:

Istruzione	M	A	R	b	B	c	C	d	D
AV	$x_1$	$x_2$	$x_7 x_8$	$x_3$	$x_4$	$x_5$	$x_6$		
RS	$x_1$	$x_2$	.....	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
.									
.									
V									

*Programmi che usano il registro D per istruzioni nella parte iniziale e, successivamente, per contenere dati numerici.*

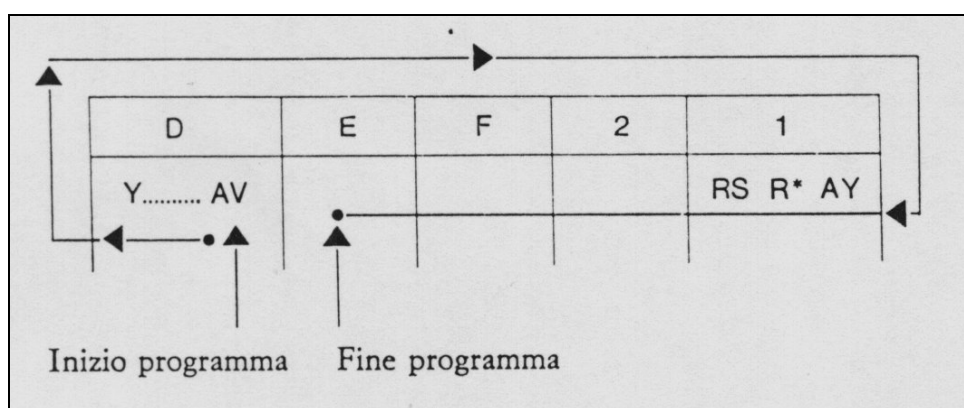
Senza la programmazione di opportune istruzioni che consentono l'uso promiscuo dei registri, non è possibile introdurre dati nei registri che ospitano istruzioni di programma. Mediante le istruzioni R\* ed RS è possibile liberare il registro D dalla parte di programma in esso contenuta, consentendo così, nel corso successivo del programma, una riutilizzazione del registro per dati numerici.

Le istruzioni R\* ed RS vanno situate nel programma in posizione tale da essere eseguite dopo le istruzioni contenute in D.

Si può in tal modo ottenere un programma di 120 istruzioni e al tempo stesso fare uso del registro D come registro numerico.

Per ripetere il programma è necessaria ogni volta la lettura della scheda magnetica.

Il programma deve essere organizzato in modo che le prime istruzioni che il calcolatore esegue siano quelle contenute in D.



*Organizzazione del programma per la duplice utilizzazione del registro D.*

## Utilizzazione del tasto S per l'avvio di un programma

Dopo la lettura di una scheda magnetica il calcolatore si porta sempre su una posizione che precede la prima istruzione del programma, dove si ferma.

L'esecuzione del programma può essere avviata sia dalla selezione dei tasti V, W, Y, Z, che indirizzano il calcolatore all'istruzione di destinazione corrispondente con cui si inizia il programma, sia dall'abbassamento del tasto S che, come già visto, ha la funzione di avviare il programma partendo dal punta in cui è fermo.

## Utilizzazione dei sottoprogrammi

Per sottoprogramma, o subroutine, si intende un vero e proprio programma autosufficiente e indipendente da altre eventuali sequenze di calcolo, che può costituire una parte del programma principale.

Si considerano sottoprogrammi sequenze di calcolo di utilità generale che possono essere inserite in programmi differenti o richiamate più volte nel corso di uno stesso programma. Per utilizzare un sottoprogramma è necessario registrarlo in memoria e richiamarlo quando occorre elaborare la sequenza di calcolo a cui si riferisce. Il programma principale deve quindi contenere istruzioni per l'accesso al sottoprogramma e quest'ultimo deve a sua volta terminare con istruzioni che permettano di rientrare nel programma principale.

L'inserimento di più sottoprogrammi in un programma principale può far superare le 120 istruzioni. Per risolvere tale difficoltà si registrano sottoprogrammi su altrettante schede magnetiche parziali che vengono utilizzate in punti prestabiliti del programma principale. Su que-

ste schede è possibile registrare soltanto i contenuti dei registri D ed E (fino a un massimo di 48 istruzioni) senza alterare il contenuto del registro F né quello dei due registri di programma. In fase di lettura esse inviano il loro contenuto soltanto nei registri D ed E. (pag. 51)

Con tale procedura si possono richiamare, nel corso di un programma principale, più sottoprogrammi differenti, introducendo semplicemente al momento opportuno le schede corrispondenti.

Per inserire un sottoprogramma è consigliabile programmare le istruzioni:

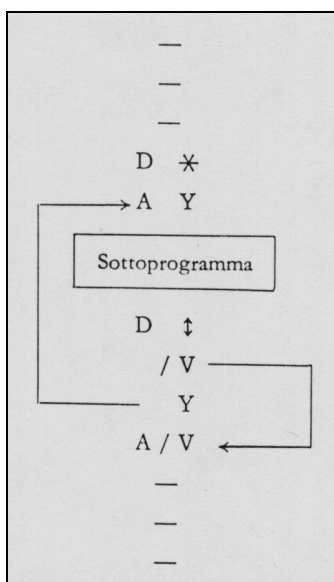
- /◇ interlinea, che indica il punto dove inserire la scheda del sottoprogramma
- S arresto del programma principale.

L'operatore dovrà poi introdurre la scheda parziale e premere manualmente il tasto di selezione del sottoprogramma.

Nella stesura del programma principale occorre non utilizzare i salti condizionati o incondizionati previsti per il sottoprogramma.

Se si deve utilizzare ripetutamente un sottoprogramma si può ricorrere, come si è visto, a un contatore, oppure, quando è sufficiente ripetere poche volte il ciclo, a tecniche particolari.

La più comune è quella di programmare dopo il sottoprogramma un'istruzione di scambio con un registro precedentemente azzerato e una istruzione di origine di salto condizionato.



Dopo la prima esecuzione del sottoprogramma, con l'istruzione  $D \uparrow$  si trasferisce zero in A e un valore (che si suppone maggiore di zero) in D. Alla lettura dell'istruzione  $/ V$ , il calcolatore non esegue il salto e proseguendo incontra l'istruzione Y che lo riporta al punto di inizio del programma.

Alla seconda ripetizione del sottoprogramma, il valore in A prima dell'istruzione  $/ V$  è  $> 0$  e il calcolatore esegue il salto portandosi all'istruzione A/V per rientrare nel programma principale. Quanto precedentemente descritto costituisce una traccia della via da seguire per problemi di tal genere. Particolari organizzazioni logiche potranno essere di volta in volta studiate per risolvere casi specifici.

#### UTILIZZAZIONE DELLE SCHEDE PARZIALI PER L'INTRODUZIONE DEI DATI

Le schede parziali non sono utilizzate solo per registrare sottoprogrammi da inserire in un programma principale, ma anche per memorizzare dati numerici.

In tal modo si possono introdurre da scheda magnetica, anziché da tastiera, dati numerici da elaborare, diminuendo le possibilità di errori derivanti dall'impostazione manuale. Ciò è particolarmente conveniente quando i dati da introdurre siano molti o debbano essere reimpostati più volte, non essendo possibile conservarli in memoria.

In quest'ultimo caso basta introdurre la scheda su cui sono registrati i dati che interessano nei punti prestabiliti del programma.

I dati numerici introdotti da scheda si posizionano nei registri E e D.

## Il calcolo manuale

Il calcolatore Olivetti Programma 101 può essere utilizzato anche manualmente in modo non dissimile dalle normali calcolatrici.

Tutte le funzioni che possono essere programmate vengono eseguite anche con comando manuale.

Nel calcolo manuale, a differenza di quello programmato, quando l'operatore imposta i dati, deve impartire al calcolatore anche le relative istruzioni impostandole sulla tastiera

## Differenza tra calcolo manuale e calcolo programmato

Quando si opera manualmente il calcolatore stampa:

- tutti i dati impostati in tastiera, i simboli dei registri interessati al calcolo e quelli delle operazioni eseguite
- il risultato delle operazioni di moltiplicazione, divisione e radice quadrata
- il contenuto di qualsiasi registro mediante una istruzione di stampa  $\diamond$  o di azzeramento  $*$ . Non è possibile azzerare i registri R ed M, per cui  $R^* = R\diamond$ ;  $* = \diamond$ .

Nel calcolo programmato il calcolatore stampa:

- i dati impostati in tastiera con accanto il simbolo S
- il contenuto di qualsiasi registro solo se è stata programmata un'istruzione di stampa  $\diamond$ . L'istruzione  $*$ , come già visto, azzerava e non stampa.

Nel calcolo manuale non è mai richiesto l'uso del tasto «S» (Start) in quanto la tastiera, alla fine di ogni singola operazione, è libera per ricevere l'impostazione di nuovi dati. Così pure non è mai richiesto l'uso dei tasti di selezione programmi V, W, Y, Z, data l'assenza di un programma registrato.

## Utilizzazione del calcolo manuale

La possibilità di far eseguire al calcolatore alcune sequenze comandate manualmente amplia evidentemente le possibilità di colloquio tra l'uomo e la macchina.

L'operatore può intervenire direttamente anche nel corso di un programma. A questo scopo occorre programmare in punti predeterminati istruzioni di Stop. Dopo l'arresto l'operatore può utilizzare il calcolatore in modo manuale, avendo però l'avvertenza di non alterare il contenuto di quei registri che ospitano dati da conservare per la prosecuzione del programma. Per calcolo programmato tornare al calcolo programmato basta premere il tasto «S».

Il calcolo manuale è particolarmente utile per verificare la correttezza di un programma. Si possono infatti eseguire manualmente tutte o parte delle istruzioni che lo compongono, controllando dopo ogni operazione il contenuto dei registri.

L'utilizzazione manuale del calcolatore può essere vantaggiosa per economizzare istruzioni, posizionando, con operazioni eseguite manualmente, i dati richiesti nei registri prestabiliti prima di avviare il programma.

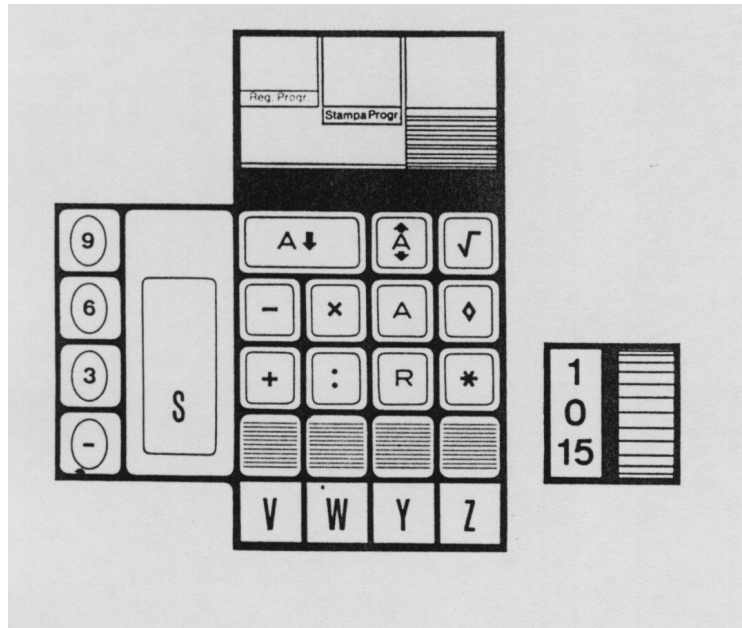
Il calcolo manuale viene inoltre utilizzato ogni volta che si debba eseguire qualche calcolo di natura non ripetitiva che non giustifichi la preparazione del corrispondente programma.

## Norme operative

### Registrazione dei programmi

Un programma è registrato in memoria direttamente da tastiera, abbassando i tasti corrispondenti ai simboli delle istruzioni. L'operazione viene eseguita nel modo seguente:

1. mettere in funzione il calcolatore
2. premere l'annullatore generale
3. premere il pulsante «registrazione programma»



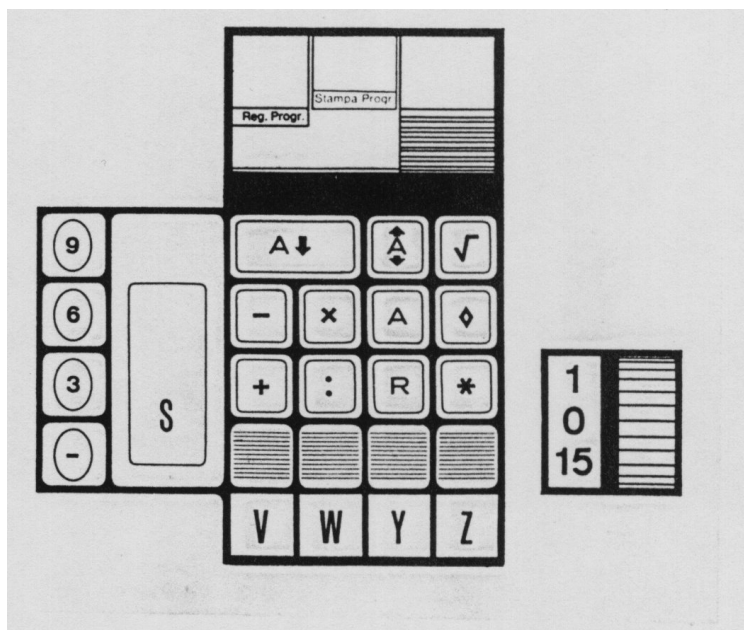
4. impostare le istruzioni secondo la sequenza programmata; ogni istruzione impostata viene stampata. Quando è costituita da un solo simbolo è stampata preceduta da M. Per correggere eventuali errori d'impostazione, premere il tasto annullatore che cancella l'ultima istruzione completamente introdotta, oppure la parte di istruzione già impostata, anche se non ancora completa. In entrambi i casi la striscia avanza di un'interlinea per consentire l'identificazione dell'istruzione erroneamente impostata.
5. disinserire il pulsante «registrazione programma» dopo l'introduzione delle istruzioni. Se il programma prevede costanti nei registri D, E, F, impostare i dati in tastiera e trasferirli nei relativi registri.

Dalla memoria il programma può essere registrato su scheda nel modo seguente:

1. inserire il pulsante «registrazione programma»
2. introdurre la scheda nell'apposita fessura dell'unità di lettura e registrazione (il dispositivo per il trascinamento automatico provvede a far passare la scheda sotto la testina di lettura e registrazione che registra nell'apposita pista magnetica le istruzioni contenute in memoria)
3. disinserire il pulsante «registrazione programma».

### Stampa dei programmi

Quando un programma è registrato in memoria, l'intera sequenza di istruzioni che 10 compone può essere stampata nel modo seguente:



(pag. 54)

1. premere il pulsante « stampa programma»
2. abbassare il tasto di selezione programma. La stampa si arresta dopo ogni istruzione di Stop; per riprenderla abbassare il tasto «◇»
3. disinserire il pulsante «stampa programma» dopo la stampa di tutte le istruzioni.

Se un programma è letto da scheda non occorre abbassare il tasto che ne seleziona l'inizio. In tal caso, dopo l'inserimento di «stampa programma», abbassare il tasto «◇» inizialmente e dopo ogni istruzione di Stop.

Alla fine del programma, abbassando il tasto «◇» viene stampato soltanto S.

### **Letture ed esecuzione dei programmi**

Per trasferire in memoria un programma registrato su scheda, si opera come segue:

1. mettere in funzione il calcolatore
2. premere l'annullatore generale
3. disinserire i pulsanti «registrazione programma» e «stampa programma»
4. inserire nella fessura dell'unità di lettura e registrazione la scheda che contiene il programma.

Per eseguire un programma già registrato in memoria, si opera nel seguente modo:

1. disinserire i pulsanti «registrazione programma» e «stampa programma»
2. predisporre l'indicatore dei decimali
3. abbassare il tasto di selezione per avviare la sequenza di programma desiderata.

Lo svolgimento del programma avviene secondo quanto previsto dalle relative norme operative. Durante l'elaborazione la tastiera è automaticamente bloccata e si libera a ogni istruzione di Stop per consentire l'introduzione dei dati.

### **Registrazione e lettura di scheda parziale**

Per introdurre in memoria istruzioni destinate a una scheda parziale si eseguono queste operazioni:

1. premere i pulsanti «registrazione programma» e «stampa programma»



2. introdurre da tastiera le istruzioni (che vanno a occupare esclusivamente i registri E e D).

Per registrare dati numerici occorre trasferirli nei registri E e D, con le normali procedure.

Per registrare su scheda parziale il contenuto di E e D, si opera nel seguente modo:

1. premere i pulsanti «registrazione programma » e «stampa programma»
2. introdurre la scheda magnetica.

Per leggere infine una scheda parziale si opera come segue:

1. premere il pulsante «stampa programma»
2. introdurre la scheda magnetica, il cui contenuto è trasferito nei registri E e D, senza alterare il contenuto degli altri registri.

## Tempi macchina

L'esecuzione di un'istruzione richiede sempre un tempo di lettura e un tempo operativo. Il tempo di lettura è costante mentre il tempo operativo varia a seconda dell'istruzione. La tabella seguente riporta i tempi macchina, espressi in millisecondi.

Istruzioni	Tempo operativo	Tempo lettura
<u>Aritmetiche</u>		
Addizione	Tempo allineamento + 8,8	2,2
Sottrazione	Tempo allineamento + 8,8	2,2
Moltiplicazione	Tempo allineamento + 8,8 (C + N)	2,2
Divisione	Tempo allineamento + 8,8 (C + N)	2,2
Radice quadrata	Tempo allineamento + 17,6 (C + N/2)	2,2
Valore assoluto	Tempo allineamento	2,2
Trasferimenti	Tempo allineamento	2,2
<u>Servizio</u>		
Stampa	500 : 1200	2,2
Interlinea	~ 500	2,2
Azzeramento	Tempo allineamento	2,2
Salti	2,2	2,2
Scambio D ed R (RS)	Tempo allineamento	2,2
Segnale di costante codificata (A/↑)	Tempo allineamento	2,2

I simboli C ed N utilizzati nella tabella significano:

### Moltiplicazione

C = somma delle cifre del fattore contenuto in A  
N = numero delle cifre del fattore contenuto in A

### Divisione

C = somma delle cifre del quoziente  
N = numero delle cifre del quoziente

### Radice quadrata

C = somma delle cifre della radice  
N = numero delle cifre della radice

Il tempo di allineamento, che deve essere considerato per le operazioni aritmetiche come un tempo di incolonnamento legato al posizionamento della virgola, varia tra 24 x 2,2 m/sec e 47 x 2,2 m/sec.

Per ragioni tecniche il tempo di allineamento interviene anche nei trasferimenti di qualsiasi tipo, nell'azzeramento e nelle istruzioni RS e A/↑.

Il tempo di stampa varia tra 500 e 1200 m/sec (per la stampa da 0 a 23 caratteri).

Il tempo varia quindi in funzione dei caratteri stampati. Considerando N il numero dei caratteri, si ha circa:

$$t = 500 + 31 N$$

## Analisi di un problema e preparazione del programma

Qualsiasi problema, di tipo scientifico o contabile, deve essere opportunamente studiato ed elaborato al fine di presentarlo al calcolatore in una forma da esso accettabile.

Si possono distinguere tre fasi principali:

- *analisi numerica*
- *analisi logica*
- *programmazione*

Ogni programma dovrà però essere successivamente controllato per verificarne l'esattezza e individuare eventuali errori.

### Analisi numerica

In questa fase si determinano i metodi di calcolo numerico più adatti al problema proposto. E' necessario infatti scegliere metodi che traducano calcoli complessi in una serie di operazioni elementari o di semplici decisioni logiche, le uniche, in sostanza, che ogni calcolatore può eseguire.

### Analisi logica

#### DIAGRAMMA DI FLUSSO

La fase precedente serve a fissare un certo numero di formule che occorre collegare opportunamente tra di loro per risolvere il problema. L'analisi logica stabilisce i collegamenti tra le varie sequenze di calcolo, prevedendo tutte le eventualità possibili e determinando per ciascuna il lavoro del calcolatore.

E' molto utile, per non dire indispensabile, rappresentare graficamente la loro concatenazione logica, sia per materializzare il ragionamento di chi ha condotto l'analisi, sia, soprattutto nei problemi più complessi, per ricordare tutte le eventualità che possono presentarsi e che devono essere programmate.

Tali rappresentazioni grafiche sono comunemente chiamate diagrammi di flusso o «flow-charts» e si realizzano mediante simboli prestabiliti, in modo che anche chi non abbia direttamente analizzato il problema li possa interpretare.

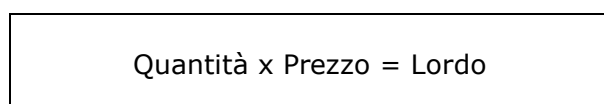
E' opportuno indicare i simboli più comunemente usati per la stesura di un diagramma di flusso.

#### Operazione generica



Ogni operazione per la quale non esistano simboli appositi viene descritta esplicitamente, o in forma simbolica, all'interno di un rettangolo. In tal caso occorre integrare il diagramma con opportune indicazioni.

Ad esempio si può avere:



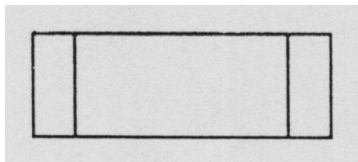
oppure

$$Q \times P = L$$

dove: Q = Quantità P = Prezzo L = Lordo

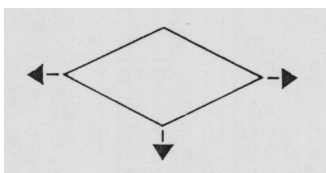
(pag. 57)

### Sottoprogramma



Questo simbolo indica un sottoprogramma. Per distinguere i vari sottoprogrammi tra loro se ne scrive, all'interno del simbolo, il nome.

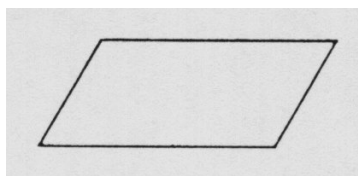
### Alternativa



Questo simbolo indica il punto del programma in cui vengono operate scelte logiche tra differenti alternative.

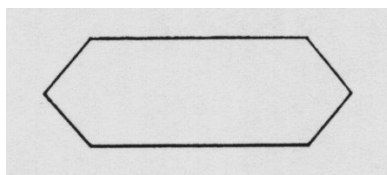
Per convenzione, se il risultato del confronto interno al simbolo è maggiore di zero si prosegue verticalmente; se è minore o eguale si continua a destra. Nel caso di triplice condizione, se il risultato è eguale a zero si prosegue a sinistra.

### Entrata o uscita dati



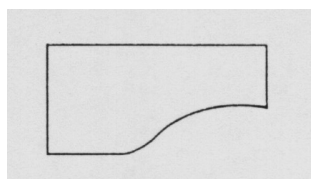
Questo simbolo indica le impostazioni da tastiera o introduzione da scheda magnetica, nonché l'uscita di dati con registrazione su scheda magnetica.

### Preparazione



Questo simbolo indica l'esistenza di un contatore o di un deviatore.

### Stampa



Si utilizza questo simbolo per rappresentare l'uscita dei risultati stampati su striscia.

### Connettori



Questi simboli sono utilizzati per collegare punti distanti di un diagramma di flusso senza bisogno di lunghe righe di collegamento.

Qualora una o più vie terminino con un circolo entro il quale è racchiuso un numero o una lettera di riferimento, è necessario ricercare il circolo che contenga lo stesso numero o lettera di riferimento e considerarlo come punto di partenza per il proseguimento del programma.

I segnali di inizio e fine di un programma sono due connettori particolari entro i quali si scrive rispettivamente Start e Stop.

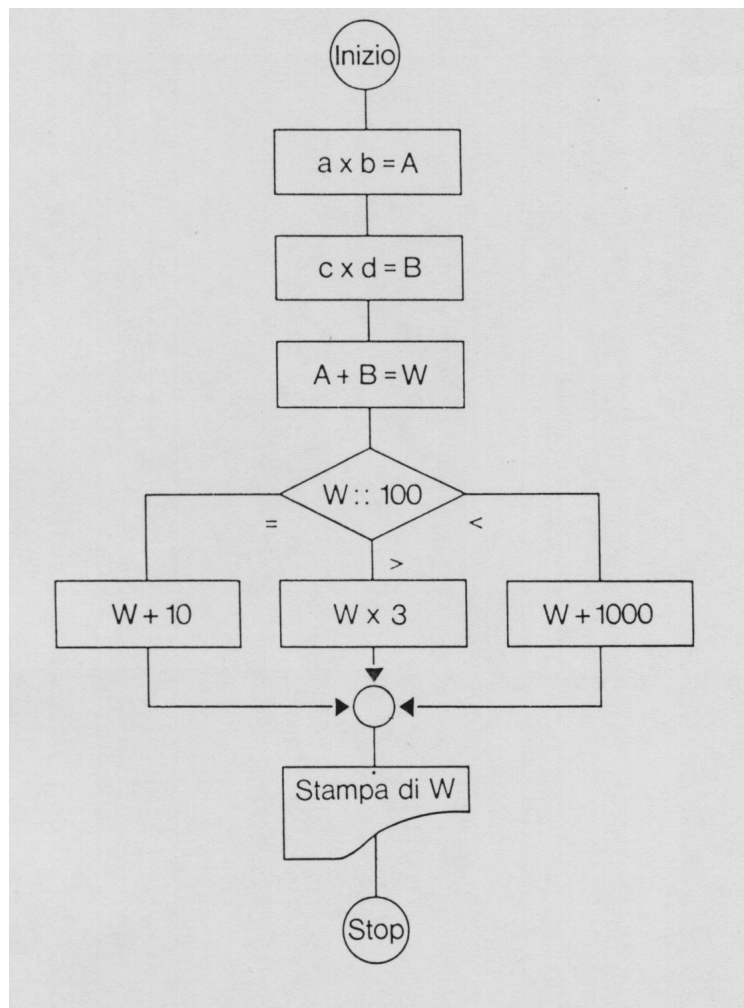
Si consideri ad esempio di dover effettuare il seguente calcolo:

$$a \times b + c \times d = W$$

con le seguenti alternative:

- $W > 100$  si calcola  $W \times 3$
- $W = 100$  si calcola  $W + 10$
- $W < 100$  si calcola  $W + 1000$

il diagramma di flusso relativo evidenzia la successione e le connessioni logiche tra le singole operazioni.



Un diagramma di flusso si disegna indipendentemente dal tipo di calcolatore su cui si realizzerà il calcolo.

Soltanto un diagramma molto dettagliato può fare riferimento al calcolatore utilizzato, ponendo in luce tutti i movimenti elementari delle singole informazioni, in relazione all'organizzazione della sua memoria e alle sue istruzioni. Un diagramma di tal genere è pressoché equivalente al programma stesso e di solito non è necessario.

## **Programmazione**

Dopo che un problema è stato esaminato in modo analitico fissando sia il metodo di calcolo sia lo sviluppo logico, occorre scriverlo in un linguaggio comprensibile per il calcolatore, codificando singole operazioni secondo regole prestabilite. Questo lavoro costituisce la stesura del programma, ovvero la programmazione.

## Messa a punta e controllo di un programma

Per *messa a punto di un programma* si intende il lavoro di controllo che un programmatore deve compiere quando si accorge che un suo programma non si svolge correttamente.

In molti casi una segnalazione di errore si ottiene dallo stesso calcolatore attraverso la luce rossa.

La luce verde indica che tutto si svolge normalmente, mentre quella rossa si accende quando si verifica qualcosa di anormale e conseguentemente il calcolatore si arresta.

La segnalazione di errore si ha quando:

- si esegue un'operazione aritmetica il cui risultato supera la capacità dei registri (overflow); in questo caso si perde il contenuto originario dei registri A ed R
- si introduce da tastiera un dato eccedente la capacità del registro M
- si trasferisce un dato formato da più di 11 cifre in un registro diviso
- si accede alla parte sinistra di un registro diviso, mentre il registro utilizzata prima nella sua interezza contiene ancora un dato avente più di 11 cifre
- si fa una divisione per zero; anche in questo caso si perde il contenuto originario dei registri A ed R
- si esegue una qualsiasi operazione che interessa un registro occupato da istruzioni.

Inoltre si accende la luce rossa quando una scheda magnetica non è stata letta in modo corretto o non è stata letta affatto perché danneggiata o introdotta irregolarmente.

Si può individuare facilmente l'istruzione che ha provocato il verificarsi di una situazione anormale facendo stampare il programma registrato a partire dall'istruzione immediatamente successiva a quella che ha causata l'errore. Ciò si ottiene abbassando il pulsante «stampa programma» e quindi il tasto «◇» (inizialmente e a ogni eventuale successiva istruzione di Stop). Nei casi in cui il calcolatore non possa segnalare l'errore (perché il programma, benché errato, non è incompatibile con la sua logica operativa), occorre ricercarne le cause controllando ad una ad una le singole istruzioni o le varie sequenze.

Questo controllo può essere eseguito usando manualmente il calcolatore Programma 101.

E' opportuno ricordare che quando si incontra in programma una costante codificata nella quale non è stata indicata la posizione della virgola, il calcolatore si arresta.

## Codificazione manuale delle costanti

Per eseguire manualmente la codificazione delle costanti, si deve procedere come segue:

1. si inizia la codificazione partendo dalla cifra meno significativa
2. si deve sempre porre l'indicazione di virgola (anche se la costante è intera) quando si codifica la cifra delle unità
3. la cifra più significativa deve contenere il segnale di fine numero
4. per ogni cifra l'istruzione si compone di due parti: la parte di destra specifica, con opportuno codice, il valore numerico della cifra; quella di sinistra, invece, è espressa sempre con codice alfabetico e specifica il segno, l'eventuale virgola e la posizione della cifra stessa nel numero.

Parte sinistra dell'istruzione			Parte destra dell'istruzione (valore numerico)
Segno	Virgola	Codice	
			0 = S
-----			
per la cifra più significativa della costante	+	SI D /	1 = ↓
		NO D	2 = ↑
	-	SI E /	3 = ↓
		NO E	4 = +
-----			
per tutte le altre cifre	+	SI R /	6 = x
		NO R	7 = :
	-	SI F /	8 = ◇
		NO F	9 = *

Tabella per la codificazione manuale delle costanti

### Esempi

#### Codificazione di 250,34

4 R +  
3 R ↓  
0 R / S  
5 R -  
2 D ↑

#### Codificazione di 1,55

5 R -  
5 R -  
1 D / ↓

#### Codificazione di -144,325

5 F -  
2 F ↑  
3 F ↓  
4 F / +  
4 F +  
1 E ↓

#### Codificazione di -0,4891

1 F ↓  
9 F \*  
8 F ◇  
4 F +  
0 E / S

#### Codificazione di 10

0 R / S  
1 D ↓

#### Codificazione di -10

0 F / S  
1 E ↓



## Tabella riassuntiva delle funzioni

Tipo delle istruzioni	Funzioni	Codici	Contenuto registri a fine operazione			
			M	A	R	X
aritmetiche	Addizione	X +	2° addendo x	Rt	Rc	x
	Sottrazione	X -	sottraendo x	Rt	Rc	x
	Moltiplicazione	X x	2° fattore x	Rt	Rc	x
	Divisione	X :	divisore x	Rt	Resto	x
	Radice quadrata Valore assoluto	X $\sqrt{\quad}$ A $\uparrow$	$2 \sqrt{x}$ inalterato	$\sqrt{x}$  a	non signific. inalterato	x x
di introduzione e trasferimento dati	Introduzione dato M (1)	S	m			
	Trasferimento da M	X $\uparrow$	m			m
	Trasferimento in A	X $\downarrow$		x		x
	Scambio con A	X $\updownarrow$		x		a
	Trasfer. parte decimale	/ $\updownarrow$	parte dec. di a		a	
di servizio	Stampa (2)	X $\diamond$				
	Azzeramento	X *				0
	Interlinea	/ $\diamond$				
logiche	Salti incondizionati (3)	$\Delta \rightarrow A\Delta$ $C\Delta \rightarrow B\Delta$ $D\Delta \rightarrow E\Delta$ $R\Delta \rightarrow F\Delta$	s'interrompe sempre la sequenza all'istruzione di origine e si riprende a quella di destinazione			
	Salti condizionati (3)	$/\Delta \rightarrow A/\Delta$ $C/\Delta \rightarrow B/\Delta$ $D/\Delta \rightarrow E/\Delta$ $R/\Delta \rightarrow F/\Delta$	s'interrompe la sequenza all'istruzione di origine per saltare a quella di destinazione solo se il contenuto di A > 0			
speciali	Segnale di costante codificata	A / t	deve precedere una costante codificata			
	Scambio tra D ed R	R S	a) scambia il contenuto di D con. quello di R b) consente di utilizzare il registro D come numerico dopo che sono state eseguite le istruzioni in esso registrate. In tal caso RS deve essere preceduto dall'istruzione R *			

Nella tabella si indica con:

X il nome di un generico registro

x il contenuto del registro X

a il contenuto del registro A

m il contenuto del registro M

Rt il risultato troncato (secondo il posizionamento dell'indicatore dei decimali)

Rc il risultato completo

(1) Può servire per selezionare una sequenza, per introdurre una scheda o per l'uso manuale del calcolatore

(2) Stampa  $x_t$

(3)  $\Delta = V, W, Y, Z$

## Indice analitico

<b>A</b>			
Arrotondamento a valori prestabiliti	43	Operazioni con dati contenuti nei registri di deposito	17
Arrotondamento all'unità più vicina	41	Operazioni con dati contenuti nei registri operativi	20
Arrotondamento per eccesso	42	Operazioni aritmetiche (sequenze di)	21
<b>C</b>		Organi di entrata	9, 12
Calcolo manuale	52	Organi di uscita	9, 12
Capacità dei registri operativi	17	<b>P</b>	
Contatore (programmazione di un)	43	Posizionamento dei dati nei registri	23
Costanti (codificazioni di)	38	Programma	10
Costanti (generazione di)	40	Programma (lettura ed esecuzione dei)	54
Costanti nei registri di deposito	38	Programma (messa a punto di un)	59
<b>D</b>		Programma (registrazione dei)	53
Decimali (operazioni con)	47	Programmazione	14
Decimali (trasferimento dei)	28	Programmazione (tecniche di)	40
Discriminazione fra tre condizioni	36	<b>R</b>	
<b>E</b>		Radice quadrata	19
Esecuzione ripetuta di un programma	32	Reciproco di un dato	40
<b>F</b>		Registrazione dei programmi	53
Funzioni aritmetiche	16	Registrazione di più dati in un unico registro	44
Funzioni di servizio	29	Registri di deposito	17
Funzioni di trasferimento	24	Registri divisi	13,18
Funzioni logiche	31	Registri operativi (capacità dei)	17
<b>G</b>		Registri operativi (operazioni con dati contenuti nei)	20
Generazione di costanti	40	Registro	13
<b>I</b>		<b>S</b>	
Interlinea	30	Salti condizionati	34
Interruzione automatica di una sequenza ciclica	44	Salti incondizionati	31
Introduzione dei dati	23	Salti incondizionati (applicazioni relative ai)	33
Inversione di segno di un dato	40	Scambio con A	26
Istruzione di azzeramento	29	Scheda magnetica	14
Istruzione di stampa	29	Schede parziali	51
<b>L</b>		Sequenze di operazioni aritmetiche	21
Lunghezza dei dati e decimali nei risultati	47	Sottoprogrammi (utilizzo dei)	50
<b>M</b>		<b>T</b>	
Memoria	10, 12	Tecniche di programmazione	40
Messa a punto e controllo di un programma	59	Tempi macchina	55
<b>N</b>		Trasferimento da M	24
Norme operative	53	Trasferimento della parte decimale	28
<b>O</b>		Trasferimento in A	25
Operazioni aritmetiche elementari	16	<b>U</b>	
		Unità aritmetico-logica	11, 14
		Unità di governo	11, 14
		Uso promiscuo dei registri F, E, D, per dati e istruzioni	45

<b>V</b>	
Valore assoluto (determinazione del)	20