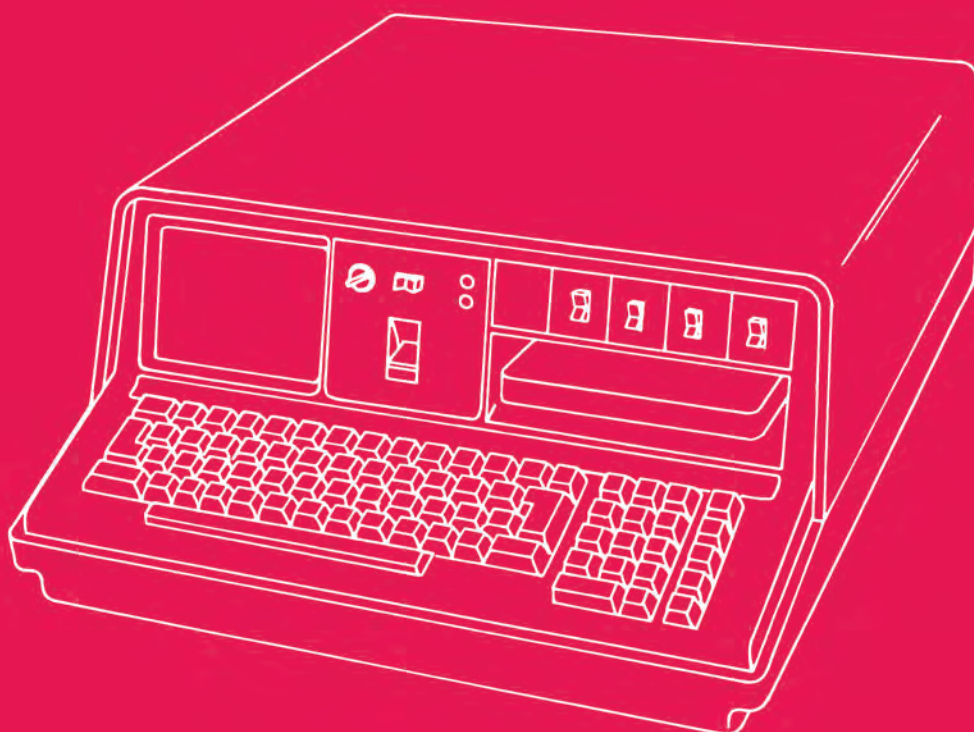




IBM 5100 APL Introduzione

5100



IBM 5100
Elaboratore Portatile
APL Introduzione

La presente pubblicazione è stata tradotta dalla corrispondente originale in lingua inglese "IBM 5100 APL Introduction" (form n. SA21 - 9212 - 1) reperibile presso la IBM Italia S.p.A.
Per eventuali aggiornamenti riferirsi alla fonte originale.

© International Business Machines Corporation, 1971, 1975

© IBM Italia S.p.A. - giugno 1976

All right reserved

SA13 - 0084 - 00

CAPITOLO 1. INTRODUZIONE ALL'IBM 5100	1
Il Manuale	1
Il Linguaggio APL	1
L'IBM 5100	1
Tasti Alfabetici	2
Tasti Numerici	2
Tasti Operativi	2
Parole chiavi di comando del Sistema	4
Tasti delle operazioni aritmetiche	4
Operazioni iniziali	4
Immissione e visualizzazione dei dati	5
Correzione degli errori	8
Sostituzione di un carattere	8
Cancellazione di un carattere	9
Inserimento di un carattere	10

CAPITOLO 2. INTRODUZIONE AL LINGUAGGIO APL	14
Tipi di funzioni in APL	14
Addizione, Sottrazione, Moltiplicazione e Divisione	14
Altra Funzione Aritmetica - Elevazione a potenza	16
Calcolo della radice di un numero	17
Memorizzazione dei dati nell'IBM 5100 per un successivo utilizzo	17
Svolgimento di alcune operazioni nel corso di una stessa istruzione	18
Specificazione dell'ordine di esecuzione - Uso delle parentesi	19
Uso di stringhe di numeri e tabelle	19
Uso del linguaggio APL con stringhe di numeri (vettori)	20
Uso del linguaggio con tabelle di numeri (Matrici)	22
Come riferirsi solo ad alcuni numeri in una stringa o tabella di numeri (Indirizzamento)	24
Non si è limitati ad usare solamente numeri	26

CAPITOLO 3. FUNZIONI APL CHE RICHIEDONO UN ARGOMENTO	28
Quanti argomenti sono richiesti da una funzione intrinseca APL	28
Simboli di funzioni APL che sono una combinazione di due caratteri	28
Determinazione del numero intero più vicino ad una data frazione	29
Arrotondamenti al numero intero più vicino	29
Disporre gli elementi di un vettore in ordine ascendente o discendente	30

Generazione di un numero casuale	31
Generazione di numeri consecutivi	31
Generazione di un vettore vuoto	32
Determinazione delle dimensioni di una variabile	32

CAPITOLO 4. FUNZIONI APL CON DUE ARGOMENTI	33
Funzioni Relazionali	34
Perchè due numeri apparentemente identici non sempre sono uguali	36
Un esempio di uso delle funzioni relazionali	36
Funzioni Logiche	37
And	38
Or	39
Determinazione del più grande di due numeri	41
Calcolo del più piccolo di due numeri	42
Determinazione dell'indice di un valore in un vettore	43
Generazione di una sequenza di numeri	43
Scelta di alcuni elementi di un vettore o di una matrice (Compressione)	44
Espansione di un vettore o di una matrice	45
Unire due vettori o due matrici	47
Costruzione di un vettore di risultati usando la concatenazione	49
Determinazione del logaritmo di un numero	51

CAPITOLO 5. APPLICAZIONE DELLE STESSE OPERAZIONI COLLETTIVAMENTE A TUTTI GLI ELEMENTI DI UN VETTORE (RIDUZIONE)	53
Riduzione	53
Uso della riduzione somma per calcolare il valore medio	53
Uso della riduzione somma per sommare i prodotti di due vettori	56
Riduzione sottrazione (somma a segno invertito)	56
Riduzione di massimo: Determinazione del valore più grande di un vettore	57
Riduzione di determinazione del valore più piccolo di un vettore	57
Riduzione Or: Ricerca di uno specifico valore in un vettore	58
Riduzione And: Confronto fra tutti gli elementi di due vettori	58

CAPITOLO 6. USO DELLE FUNZIONI INCORPORATE	59
Vediamo allora ciò che si può fare	59

CAPITOLO 7. DEFINIZIONE DI FUNZIONI	71
Cos'è la definizione di una funzione?	71
Come si definisce una funzione?	71
Controllo della funzione prima dell'uso	73
Come usare le funzioni di trace e di stop	75
Trace T Δ	75
Stop S Δ	75
Come fare quando si commette un errore nella definizione di una propria funzione	77
Visualizzazione dell'intera funzione	78
Aggiunta di una o più istruzioni alla fine della funzione	79
Sostituzione all'interno di una funzione	80
Inserimento di una o più istruzioni in una funzione	81
Cancellazione di una istruzione da una funzione	82
Visualizzazione di una determinata istruzione, oppure da una determinata istruzione fino alla fine della funzione	83
Modifica di una singola istruzione	83
Come variare l'intestazione della funzione	84
Una via più rapida, per raggiungere, sostituire o inserire una istruzione in una funzione	85
Tipi di intestazione di funzioni	87
Perché si usa una variabile risultato?	88
Nomi locali e globali	89
Salto, identificazione e iterazione (Looping)	91
Salto e identificazione	91
Iterazione (looping)	94
Come introdurre dati durante l'esecuzione di una funzione	95

CAPITOLO 8. COSA FARE QUANDO LA PROPRIA FUNZIONE SI INTERROMPE	98
Quando si preme il tasto ATTN	98
Quando si usa la funzione di stop	100
Quando viene trovato un errore nella funzione	101
Quando si immette un carattere Φ per una richiesta di dati $\square$	102
Determinazione di quali funzioni sono sospese	102
Uso del tasto di mantenimento Hold per arrestare temporaneamente l'esecuzione	103
CAPITOLO 9. USO DELLA CASSETTA A NASTRI (LIBRERIA)	104
Che cosa sono le variabili del sistema?	108

IL MANUALE

Il presente manuale insegna come operare con l'elaboratore IBM 5100 usando il linguaggio APL. Chi non conosce l'uso del linguaggio APL, deve effettuare le operazioni basilari proposte o gli esempi sul proprio 5100 seguendo il manuale dall'inizio alla fine. Chi conosce il linguaggio APL deve leggere i Capitoli 1 e 2 per imparare come usare il 5100 e poi può passare direttamente al Capitolo 7. In questo manuale non si trovano trattate tutte le caratteristiche o le funzioni del linguaggio APL. Per maggiori informazioni sul calcolatore 5100 o sul linguaggio APL si rimanda all'IBM 5100 APL, Manuale di riferimento.

Il presente manuale è stato scritto supponendo che il 5100 sia stato installato e collaudato. Se il 5100 non è stato ancora installato, è necessario seguire la procedura di installazione su "IBM 5100 APL Manuale di Riferimento prima di proseguire la presente lettura.

IL LINGUAGGIO APL

Nel linguaggio APL si trovano inserite un gran numero di funzioni che permettono di risolvere molti problemi. Tuttavia, se è necessaria una speciale funzione per la soluzione di un problema, l'APL permette anche la definizione di una propria funzione. Le funzioni così definite sono simili a programmi scritti in altri linguaggi.

L'APL è un ottimo linguaggio con cui esercitarsi; in nessun caso l'uso della tastiera può danneggiare il 5100, anzi più si fa esercizio, più si apprende l'uso dell'APL.

L'IBM 5100

L'IBM 5100 (Figura 1) è un elaboratore da tavolo progettato per aiutare la soluzione dei problemi. Lo schermo e le spie luminose trasmettono le informazioni, la tastiera e gli interruttori permettono il controllo e l'esecuzione delle operazioni del 5100.

Prima di usare il 5100, è necessario prendere confidenza con la tastiera ed il pannello di controllo (Figura 1). Gli interruttori del pannello di controllo saranno spiegati più avanti. Seguirà ora una breve spiegazione della tastiera; il suo uso verrà descritto più avanti.

Tasti alfanumerici

I tasti alfanumerici sono simili a quelli di una normale macchina da scrivere, fatta eccezione per l'assenza dei caratteri minuscoli. I caratteri alfanumerici sono tutti maiuscoli, anche se sui tasti si trovano in posizione minuscola. Pertanto il tasto  per le maiuscole (shift) non va usato per i caratteri alfabetici.

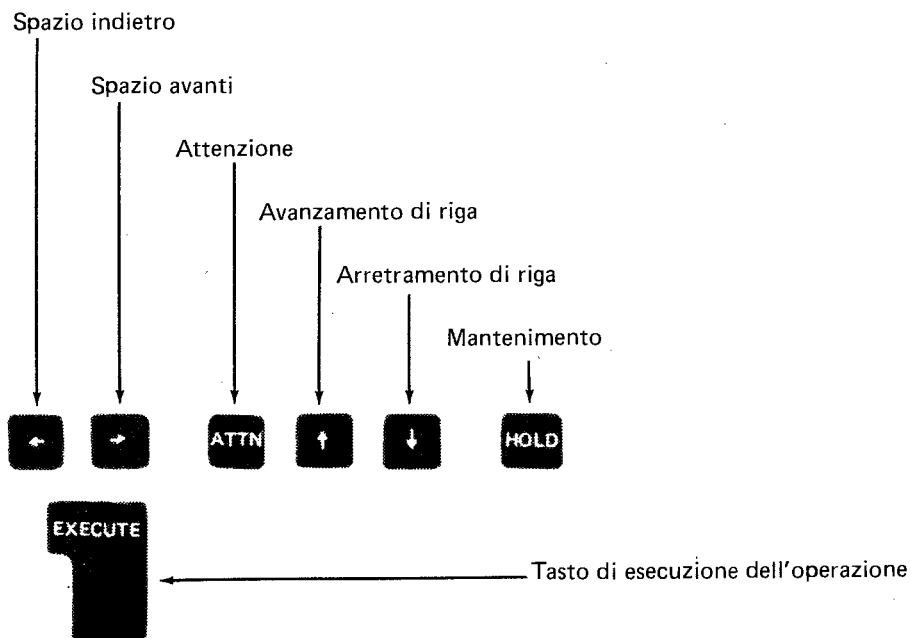
Per inserire il carattere superiore del tasto, basta tener premuto il proprio come si scrive un carattere maiuscolo in una ordinaria macchina da scrivere.

Tasti Numerici

Sia la riga alta dei tasti alfanumerici, sia lo speciale raggruppamento di tasti numerici, possono essere usati per l'inserimento dei numeri.

Tasti Operativi

Il tasto nero CMD, quelli grigi con le scritte EXECUTE, ATTN ed HOLD ed i tasti grigi con le frecce sono tutti tasti operativi speciali. I tasti con le frecce e la sbarra spaziatrice, usata per stampare i caratteri vuoti, automaticamente ripetono le operazioni, quando vengono tenuti premuti.



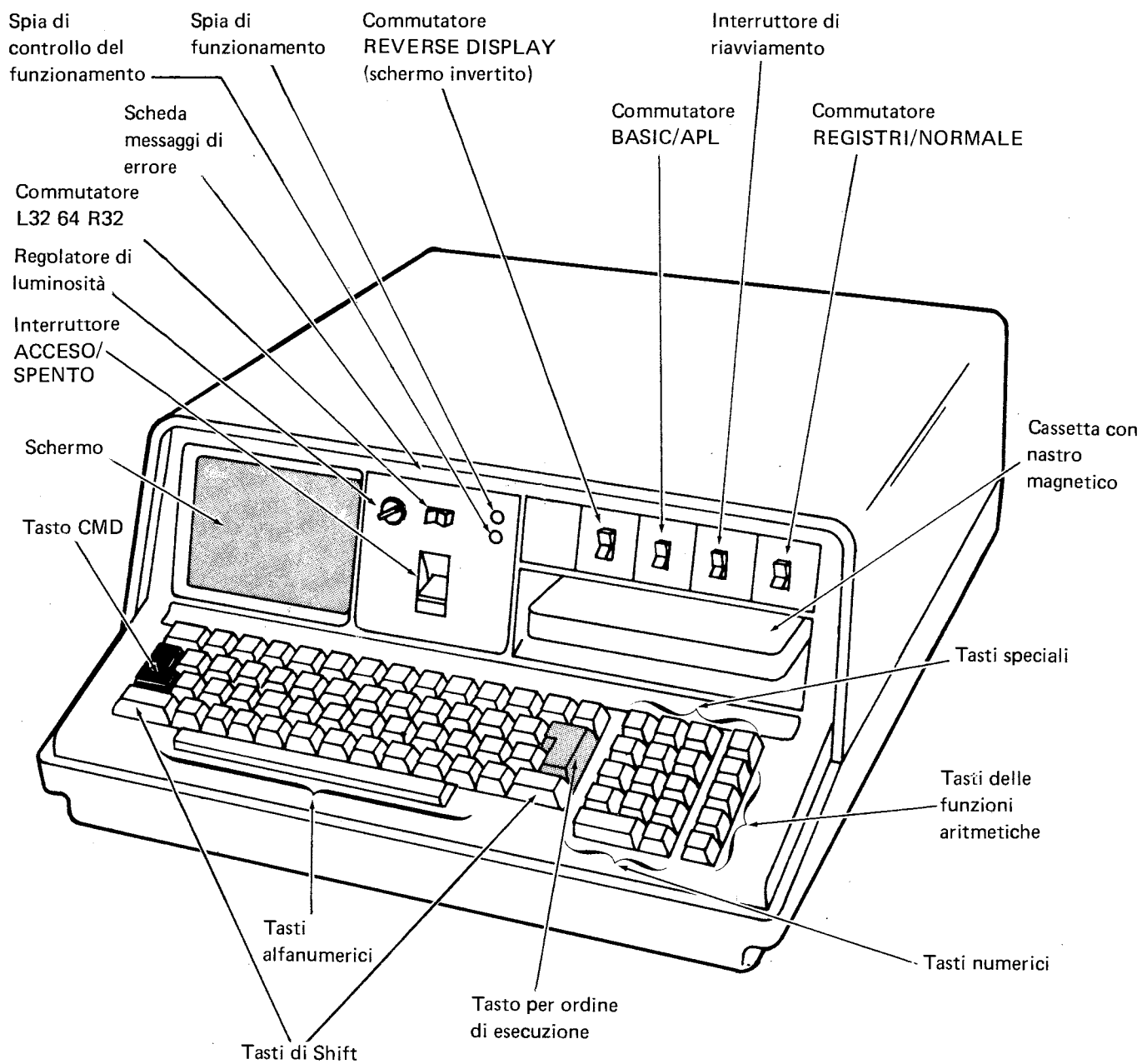


Figura 1. II 5100

Parole chiavi di comando del Sistema

Le parole che si trovano al di sopra della riga superiore dei tasti numerici sono le parole chiavi di comando del sistema, che si possono usare tenendo premuto il tasto CMD e battendo poi il tasto sotto la parola voluta. Ad esempio, per usare)LOAD, bisogna battere il tasto 1 tenendo premuto il tasto CMD. I comandi del sistema ed il loro uso sono trattati nel Capitolo 9.

Tasti delle operazioni aritmetiche

I quattro tasti posti a destra dei tasti numerici sulla tastiera del calcolatore sono i tasti delle funzioni aritmetiche. Essi vengono usati per divisioni, moltiplicazioni, sottrazioni e addizioni. Ci sono anche altri tasti tra quelli alfanumerici che eseguono le stesse operazioni. Tener presente che i simboli $\div$ e $\times$ indicano le operazioni di divisione e moltiplicazione.

OPERAZIONI INIZIALI

Assicurarsi che gli interruttori dell'IBM 5100 si trovino nella seguente posizione:

Interruttore	Posizione
L32 64 R32	64
BASIC/APL (solo nel modello con due linguaggi)	APL
REGISTRI/NORMALE	NORMALE

Nei calcolatori che hanno l'interruttore BASIC/APL, si possono usare sia istruzioni del linguaggio BASIC sia quelle dell'APL. Il linguaggio da usare viene scelto dal 5100 all'avviamento o alla ripartenza.

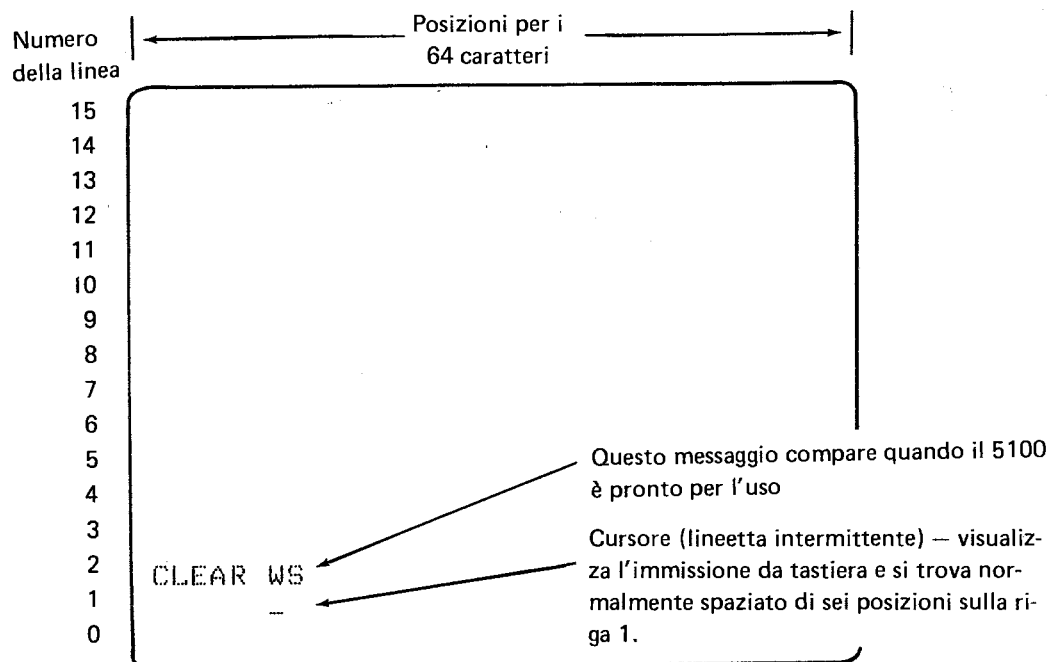
Assicurarsi che la spina del 5100 sia ben inserita nella presa di rete e quindi agire sull'interruttore di accensione. Se l'interruttore è già nella posizione di acceso, premere RIPARTENZA (RESTART) ed aspettare circa 20 secondi. Durante questo tempo il calcolatore effettua automaticamente alcuni controlli interni per assicurarsi che tutto sia in ordine.

Dopo 30 secondi, se non è comparso il messaggio CLEAR WS nell'angolo a sinistra in basso dello schermo, è stato individuato un errore durante il controllo interno. In questo caso bisogna premere nuovamente RIPARTENZA. Il calcolatore effettua di nuovo un controllo interno. Se dopo alcuni tentativi il messaggio CLEAR WS non è apparso, è opportuno chiamare il servizio di manutenzione.

IMMISSIONE E VISUALIZZAZIONE DEI DATI

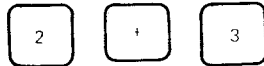
Per prima cosa si fissa l'attenzione allo schermo video. Normalmente le informazioni visualizzate dall'APL iniziano al margine dello schermo; i dati immessi da tastiera appaiono spaziati dal bordo quando vengono visualizzati. La lineetta orizzontale intermittente indica la posizione sulla riga dove ciò che sarà successivamente immesso attraverso la tastiera sarà visualizzato. Questa lineetta viene chiamata cursore. Il cursore si sposta quando i caratteri vengono visualizzati.

Lo schermo può contenere fino a 16 righe. Ciascuna riga possiede sullo schermo video, 64 posizioni. Le ultime due righe in basso vengono usate per visualizzare l'immissione, le rimanenti 14 righe contengono l'elenco delle operazioni precedentemente eseguite.



Vi sono 128 posizioni utilizzabili per l'immissione da tastiera; vi sono in particolare 64 posizioni utili sulla riga 1 e 64 sulla riga 0. Una volta utilizzata la sessantaquattresima posizione della riga 1, allorchè si vogliano inserire ulteriori dati della tastiera, il cursore riparte dal margine sinistro della riga 0. A questo punto il cursore si trova nella posizione 65 delle 128 posizioni utilizzabili per l'immissione.

Si provi ora ad immettere alcuni dati nel 5100 usando la tastiera numerica ed i tasti delle funzioni aritmetiche. Si premano nell'ordine i seguenti tasti:



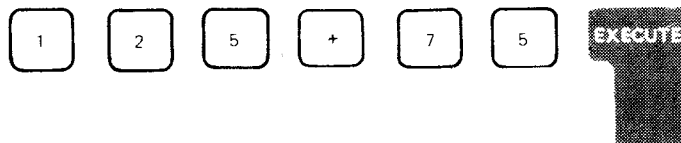
Si tenga presente che i caratteri vengono visualizzati non appena il corrispondente tasto viene premuto. Per elaborare i dati immessi bisogna premere il tasto EXECUTE. Si preme allora il tasto EXECUTE.

Lo schermo mostrerà:

```
CLEAR WS
      2+3
5
---
```

Si tenga presente che l'istruzione immessa, $2 + 3$, appare spaziata sullo schermo video; la risposta, 5, appare al margine sinistro sulla riga successiva; inoltre il cursore compare sulla riga ancora successiva. Le informazioni visualizzate si muovono verso l'alto ogni volta che viene battuto il tasto EXECUTE.

Si metta e si esegua l'istruzione $125 + 75$ battendo nell'ordine i seguenti tasti:



Lo schermo video mostrerà:

```
CLEAR WS
      2+3
5
      125+75
200
---
```

L'aspetto dello schermo può essere cambiato mediante i commutatori posti sul pannello di comando. L'interruttore REVERSE DISPLAY (SFONDO CHIARO/SFONDO SCURO) permette di cambiare da caratteri neri su fondo bianco a caratteri bianchi su fondo nero o viceversa. Pertanto si cambi la posizione dell'interruttore e si scelga il tipo di schermo con cui è più confortevole lavorare. E' opportuno, allorchè si passa da un tipo di schermo ad un altro, regolare la sua luminosità.

Ora, si controlli lo schermo mentre si pone sulla posizione S32 il commutatore S32 64-D32.

Con l'interruttore in questa posizione, i primi 32 caratteri a sinistra sulla riga vengono visualizzati con uno spazio in più fra un carattere e l'altro.

Gli ultimi 32 caratteri a destra sulla riga non saranno visualizzati. Pertanto con l'interruttore posto nella posizione S32, lo schermo video appare così:

```

C L E A R   W S
          2 + 3
5
          1 2 5 + 7 5
2 0 0

```

Nella posizione D32, 32 caratteri più a destra sono visualizzati con uno spazio tra un carattere e l'altro. Ora, se si pone l'interruttore nella posizione D32, si può notare che lo schermo è vuoto, in quanto non vi sono caratteri nelle ultime 32 posizioni a destra dello schermo video.

Si riporti l'interruttore nelle prime posizioni 64, e si osservi che tutti i caratteri sono visualizzati senza spazi tra loro. Per gli esercizi che vi sono nel resto del manuale, si mantenga l'interruttore nella posizione 64.

Vi sono due tasti, sopra il gruppo dei tasti numerici, che spostano la riga del video verso l'alto o verso il basso. La freccia verso l'alto



(avanzamento di riga) sposta verso l'alto lo schermo di una riga e la freccia verso il basso



(arretramento di riga) sposta verso il basso lo

schermo di una riga. Nel caso in cui le righe sono spostate verso l'alto o verso il basso, l'informazione visualizzata nella riga che viene spostata fuori dallo schermo video risulta persa. Sia l'uno sia l'altro dei tasti, se vengono tenuti premuti continuano a spostare le righe del video. Si provi ad usare la freccia verso il basso per spostare lo schermo di una riga verso il basso.

Lo schermo apparirà così:

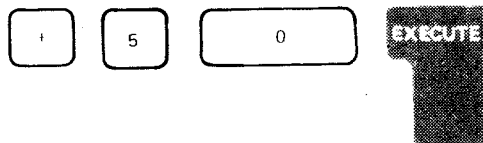
```

CLEAR WS
          2+3
5
          125+75
200

```

Il valore 200 è ora sulla linea dell'input per essere usato come input. Si noti che l'input può iniziare in qualunque posizione sulla linea.

Battiamo i seguenti tasti:



Lo schermo mostrerà:

```
CLEAR WS
      2+3
5
      125+75
200  +50
250
```

Ora che si è raggiunta una certa familiarità con lo schermo, saranno mostrate solo le righe di cui si parla.

CORREZIONE DEGLI ERRORI

L'IBM 5100 possiede alcuni dispositivi molto utili che permettono la correzione degli errori effettuata durante l'immissione dei dati. Linea per linea ed in qualsiasi momento si può:

- Sostituire un carattere
- Cancellare un carattere
- Inserire un carattere

Sostituzione di un carattere

Per sostituire un carattere si sposta il cursore con il tasto di spazio indietro  oppure il tasto di spazio avanti  finché il

cursore è posizionato in avanti.

Il cursore si sposta di uno spazio equivalente a un carattere nella direzione della freccia ogni volta che il relativo tasto viene battuto. Questi tasti continuano a muovere il cursore se sono tenuti premuti. Quando il cursore è posizionato al carattere sbagliato, lo si sostituisce semplicemente premendo il carattere giusto.

Supponiamo di dover risolvere il problema $22 + 12$ invece, vengono battuti i seguenti tasti

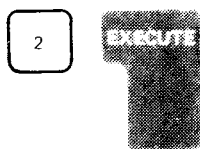


Lo schermo video mostrerà:

22+11...

Per correggere l'errore basta spostare il cursore indietro di una posizione (sotto il secondo 1) in modo da poter ribattere il carattere. Si batta pertanto il tasto di spazio indietro  una volta. Si può notare che

il cursore sarà sostituito da un carattere intermittente. Il carattere intermittente ha la stessa funzione del cursore; esso indica la posizione sulla riga dove può essere immesso e visualizzato il carattere successivo. Pertanto per correggere l'errore e risolvere il problema, si battano i tasti seguenti:



Cancellazione di un carattere

Per cancellare un carattere, si fa uso anche ora del tasto di spazio indietro  o di spazio avanti  per spostare il cursore.

Allorchè il cursore si troverà nella posizione in cui deve essere cancellato il carattere (il carattere sarà intermittente), si dovrà, tenendo premuto il tasto CMD, battere ancora il tasto di spazio indietro. In questo modo sarà cancellato il carattere ed inoltre ciascun carattere alla sua destra verrà spostato a sinistra di una posizione in modo da riempire lo spazio vuoto lasciato dalla cancellatura.

Supponiamo di voler risolvere il problema $13 + 45$. Viceversa vengono battuti i seguenti tasti:



Lo schermo video ci mostrerà:

123+45...

Bisognerà battere il tasto di spazio indietro e spostare il cursore (carattere intermittente) indietro fino al 2. Si osservino le etichette che sono al disopra dei tasti di spazio indietro, e spazio avanti: DELETE (CANCELLA) e INSERT (INSERISCE). Per cancellare il 2, bisogna, tenendo premuto il tasto CMD, battere una volta il tasto



Lo schermo video mostrerà:

13+45

— Questo carattere è intermittente

Bisognerà quindi, per eseguire il problema, battere il tasto EXECUTE.

Inserimento di un carattere

Per inserire un carattere, si posiziona il cursore usando i tasti di spazio indietro  e di spazio avanti  quindi tenendo premuto il tasto CMD si batte una volta il tasto di spazio avanti . Mediante

questa operazione si sposta il carattere intermittente (e con lui tutti gli altri caratteri alla sua destra) di una posizione a destra creando appunto lo spazio che serve per l'inserimento di un carattere. Il cursore con tale operazione, non viene spostato. Pertanto, per inserire il carattere basta semplicemente battere il tasto desiderato.

Supponiamo di voler risolvere il problema 123×6 . Invece vengono battuti i seguenti tasti:



Lo schermo video ci mostrerà:

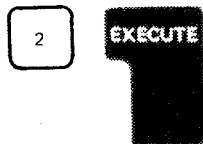
13x6_

Per correggere l'errore, bisogna battere il tasto di spazio indietro e spostare il cursore (carattere intermittente) fino al 3. Tener presente le scritte che compaiono sopra i tasti di spazio indietro, e spazio avanti: DELETE (CANCELLA) e INSERT (INSERISCE) Per svolgere l'operazione di inserimento con il cursore posizionato sul 3, battere una volta, tenendo premuto il tasto CMD, il tasto 

Lo schermo video mostrerà:

1_3x6

Per correggere l'errore di battuta ed eseguire il problema, basta battere nell'ordine i seguenti tasti:



C'è anche un'altra maniera per correggere gli errori di battuta. Se sono stati commessi alcuni errori sparsi lungo una riga, si può spostare indietro il cursore fino al carattere seguente l'ultimo carattere corretto e poi battere il tasto ATTN (attenzione). Tutto ciò che si trova tra la posizione del cursore e la fine della riga di immissione viene cancellato dallo schermo.

Poichè i tasti che sono sulla riga di immissione non vengono elaborati finchè è stato battuto il tasto EXECUTE, è possibile verificare visivamente ogni immissione prima di elaborarla. Tuttavia, se capita di notare un errore tipo aver battuto il tasto EXECUTE, si può semplicemente ribattere nuovamente i dati esatti in ingresso oppure, usando la freccia verso il basso  (arretramento di riga), spostare la riga che contiene l'errore

fino alla riga di immissione dati e quindi correggerla.

Per ogni caso si deve ribattere nuovamente il tasto EXECUTE.

Per esempio si voglia risolvere il problema $135 + 280$, ma si immette e si esegue l'operazione $134 + 280$. Lo schermo mostrerà:

134+280
414

Per eseguire la correzione, si batte tre volte il tasto freccia verso il basso  per cancellare il risultato dallo schermo. Questo ci mostrerà allora:

134+280

Battendo una volta il tasto  freccia verso l'alto si risposta la linea con gli originari tasti di ingresso sulla prima riga di ingresso e si esegue quindi la correzione.

Da questo punto in avanti saranno usati esempi per spiegare quanto si andrà discutendo, nel formato illustrato qui sotto. Saranno immesse le istruzioni che si trovano spaziate rispetto al tasto. I risultati visualizzati sul 5100 devono essere uguali a quelli illustrati nel presente manuale.

ESEMPIO:

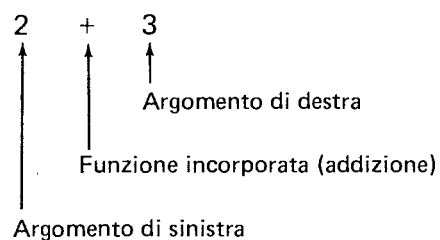
3+4 ← Istruzioni da immettere
7 ← Risultati

**Si ricordi che i dati immessi non sono elaborati
finchè il tasto EXECUTE non viene premuto.**

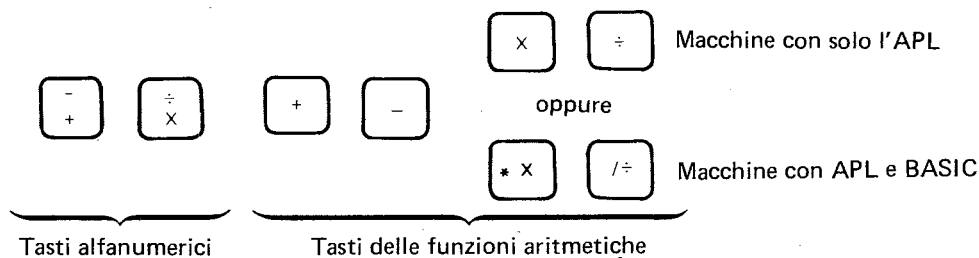
TIPI DI FUNZIONI IN APL

Esistono due tipi di funzioni nel linguaggio APL: le funzioni definite dall'utente (programmi) e quelle che si trovano incorporate nel linguaggio APL. Le funzioni incorporate nel linguaggio APL sono denotate con simboli speciali. Le funzioni definite dall'utente verranno discusse più oltre nel Capitolo 7.

Le funzioni incorporate operano sui dati forniti dall'utente, detti argomenti. Per esempio:

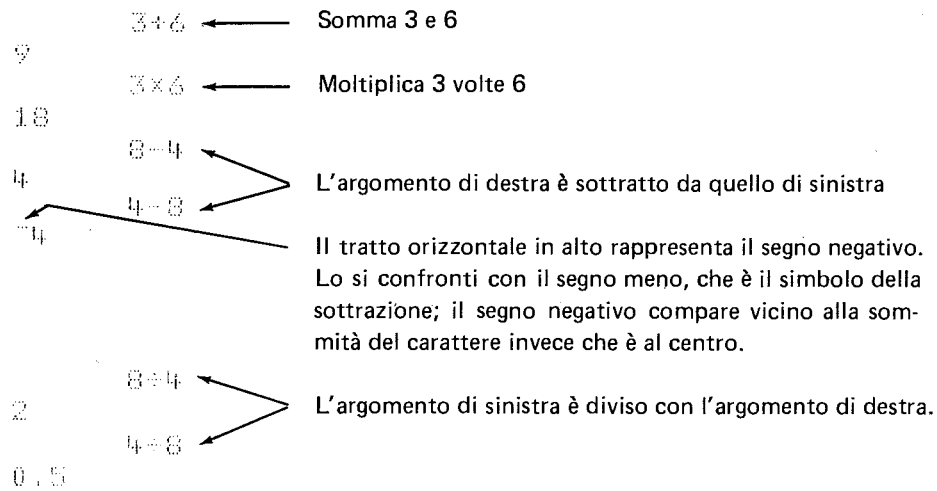


ADDIZIONE SOTTRAZIONE, MOLTIPLICAZIONE E DIVISIONE



Quattro funzioni incorporate, $(+ - \times \div)$ le normali operazioni aritmetiche. Questi simboli sono posti sulla fila più alta per testi alfanumerici oltre che a destra dei tasti numerici.

ESEMPI:



Come si è visto nell'esempio, il segno negativo è differente dal segno di sottrazione. Effettuando operazioni aritmetiche nel linguaggio APL è importante ricordarsi di non usare il segno meno per impostare i numeri negativi ovvero usare il segno negativo  per effettuare una sottrazione.

Problemi: L'uso delle operazioni di Addizione, Sottrazione, Moltiplicazione e Divisione.

1. Calcolare il numero totale di macchine che un negoziante ha venduto durante una settimana, se le vendite giornaliere sono state 3, 5, 2, 6, 7, 3 e 4.
2. Calcolare quante auto restano al venditore se 20 persone le hanno ritirate.
3. Calcolare l'incasso medio lordo per auto del negoziante se ha realizzato un incasso lordo totale di \$ 2700 per le vendite nel problema 1.
4. Calcolare il guadagno netto del negoziante se ha realizzato \$ 20 per ogni auto venduta.

Possibili soluzioni

Problema 1:

$$30 \quad 3+5+2+6+7+3+4$$

Problema 2:

$$10 \quad 30-20$$

Problema 3:

$$90 \quad 2700 \div 30$$

Problema 4:

$$600 \quad 20 \times 30$$

ALTRA FUNZIONE ARITMETICA - ELEVAZIONE A POTENZA



Un'ulteriore operazione aritmetica che probabilmente è già familiare al lettore è l'elevazione di un numero a potenza. Nel linguaggio APL si usa la funzione * per elevare l'argomento di sinistra alla potenza specificata dall'argomento di destra.

ESEMPI:

$$\begin{array}{lcl} 9 & 3*2 & \longleftarrow 3 \text{ elevato alla seconda potenza} \\ 8 & 2*3 & \longleftarrow 2 \text{ elevato alla terza potenza} \end{array}$$

Calcolo della radice di un numero

Si può utilizzare la funzione di potenza * per calcolare la radice di un numero. Per farlo basta semplicemente elevare il numero alla potenza $1:n$, dove n rappresenta la radice che si vuol calcolare.

ESEMPI:

$2 \quad 4 * (1 \div 2) \leftarrow$ Radice quadrata di 4

$2 \quad 4 * .5 \leftarrow$ Un altro modo di immettere l'istruzione per determinare la radice di un numero (.5 è lo stesso di $1 \div 2$).

$2 \quad 8 * (1 \div 3) \leftarrow$ Radice cubica di 8.

MEMORIZZAZIONE DEI DATI NELL'IBM 5100 PER UN SUCCESSIVO UTILIZZO

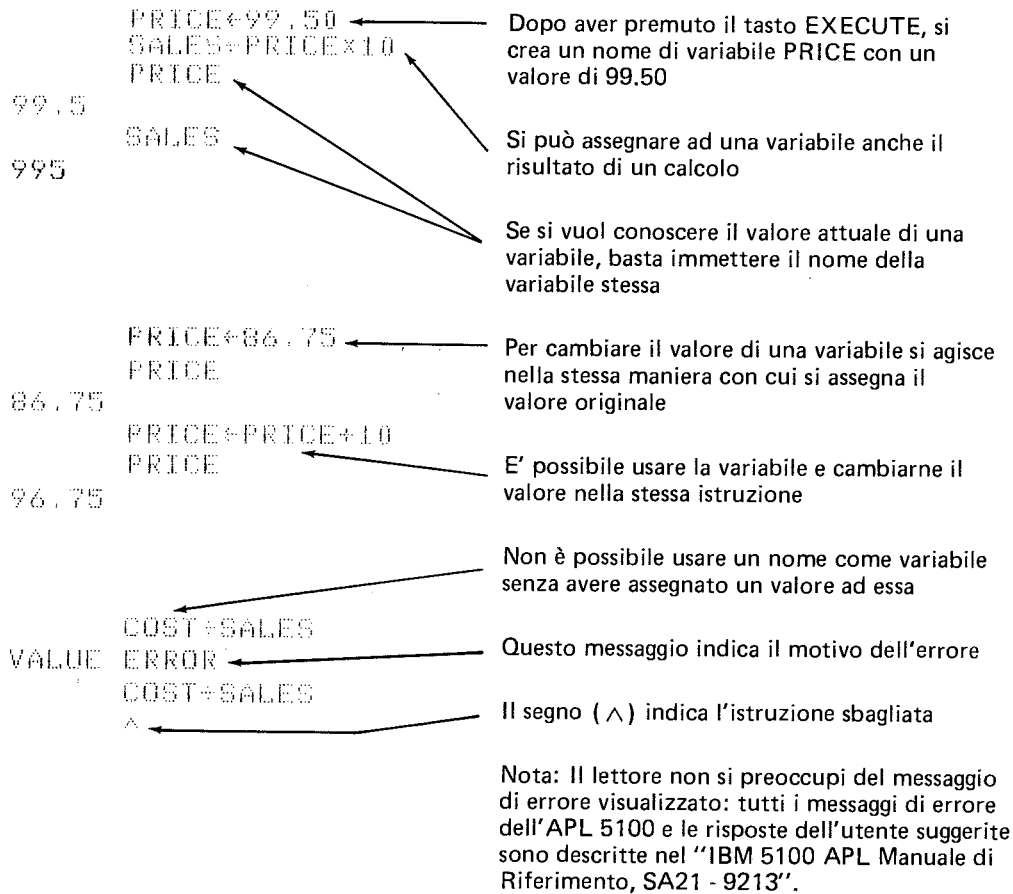


E' possibile memorizzare tutti i dati, sia quelli che vengono immessi direttamente da tastiera, sia i risultati di un calcolo. Questi dati memorizzati sono chiamati variabili. Ad ogni variabile viene associato un nome.

Ogni volta che viene utilizzato il nome della variabile, l'APL fornisce il valore collegato al nome. Il nome di una variabile può esser lungo fino a 77 caratteri (senza spazi fra i caratteri); il primo dei 77 caratteri deve però essere alfabetico; i rimanenti caratteri si possono raggruppare con una combinazione qualsivoglia di caratteri alfabetici e numerici. E' buona norma utilizzare nomi che ricordino i dati memorizzati. Per esempio, quando si memorizza un numero che rappresenta l'area di un rettangolo, è opportuno usare il nome AREA oppure se si vogliono memorizzare alcuni dati relativi alle vendite, è opportuno usare il nome VENDITE.

In questo modo si crea una variabile assegnando un tasto ad un nome. Per assegnare un valore ad un nome, bisogna utilizzare la freccia $\leftarrow$. Il valore posto a destra della freccia $\leftarrow$ è assegnato al nome posto a sinistra della freccia $\leftarrow$.

ESEMPI:



SVOLGIMENTO DI ALCUNE OPERAZIONI NEL CORSO DI UNA STESSA ISTRUZIONE

Negli esempi precedentemente illustrati è stata usata una sola funzione aritmetica per ciascun esempio. Tuttavia, non siamo limitati a scrivere delle istruzioni che contengano una sola funzione. Un numero qualunque di funzioni può trovarsi nella stessa istruzione. Dato che presto si useranno operazioni con più funzioni, è opportuno spiegare quale è la regola con cui esse vengono effettuate.

Nel linguaggio APL la funzione che si trova più a destra — nell'istruzione viene eseguita per prima, poi la successiva e così via.

ESEMPI:

L'ordine di esecuzione è da destra a sinistra

18 $3 \times 2 + 4$ ← 4 è sommato a 2 ed il risultato è moltiplicato per 3
10 $4 + 3 \times 2$ ← 3 è moltiplicato per 3 ed il risultato è sommato a 4

Si ricordi che una funzione APL usa come argomento il risultato dell'espressione alla sua destra.

SPECIFICA DELL'ORDINE DI ESECUZIONE - USO DELLE PARENTESI



Nel linguaggio APL, le parentesi vengono usate nella stessa maniera in cui si utilizzano nell'aritmetica convenzionale. Le operazioni all'interno delle parentesi vengono prima delle operazioni e immediatamente fuori di esse.

ESEMPI:

10 $(3 \times 2) + 4$ ← L'espressione 3×2 è eseguita per prima ed il risultato è sommato a 4
14 $(4 + 3) \times 2$ ← L'espressione $(4 + 3)$ è eseguita per prima ed il risultato è moltiplicato per 2.

Ricordarsi che: la regola dell'ordine di esecuzione è da destra verso sinistra dopo aver risolto per prima cosa le operazioni all'interno delle parentesi e ugualmente da destra a sinistra all'interno di esse.

USO DI STRINGHE DI NUMERI E TABELLE

Una efficace caratteristica del linguaggio APL è il sistema di trattare stringhe e tabelle di dati. Finora è stato utilizzato il linguaggio APL con l'uso di numeri singoli (chiamati scalari): l'APL invece può lavorare con stringhe di numeri (vettori) e tabelle (matrici). Le operazioni che sono state eseguite usando numeri, sono facilmente estendibili a ciascun numero in una stringa o una tabella. Se si ha per esempio una stringa di numeri assegnata alla variabile col nome SALES, si può sommare il valore 2 ad ogni numero della catena semplicemente immettendo $2 + \text{SALES}$.

Uso del linguaggio APL con stringhe di numeri (vettori)

Una stringa di numeri è chiamata vettore. Allorchè si immetta una stringa di numeri è necessario lasciare almeno uno spazio vuoto, tra due numeri; ciascun numero è chiamato elemento di vettore.

ESEMPIO:

```
1 44 2 9 35 ← E' stato immesso un vettore di 5
1 44 2 9 35      elementi (una stringa di 5 numeri)
STRING←144 16 39 2
STRING
144 16 39 2      Un vettore può essere assegnato
                  al nome di una variabile
SALES←125 220 316 90
SALES×10          Si può operare su ciascun elemen-
1250 2200 3160 900 to (numero) del vettore con un
                  unico numero.
```

```
SALES ← Si noti che il valore di SALES non
125 220 316 90 è cambiato
PRICE←.50 1.00 .75 1.10
TOTAL←SALES×PRICE
TOTAL ↑ Si può operare su ciascun elemen-
62.5 220 237 99 to di un vettore per mezzo del
corrispondente elemento in un
altro vettore avente lo stesso nu-
mero di elementi.
```

```
1 2 4+4 5 6 ←
5 7 10      Ci deve essere almeno uno spazio
124+456 ← fra ogni elemento del vettore, op-
580         pure il risultato sarà diverso.
```

```
1 2 3+4 5 ← Non è possibile usare due vettori
LENGTH ERROR che non abbiano lo stesso numero
1 2 3 + 4 5 di elementi, a meno che uno degli
              ^ argomenti non sia un numero
              singolo.
```


Problemi: Uso di stringhe di numeri

1. Calcolare il quadrato dei numeri da 1 a 5.
2. Calcolare il quadrato, il cubo e la quarta potenza dei numeri 2 e 3.
3. L'amministratore di un piccolo fondo mutuo suddivide l'attività in 5 fondi. Egli vuol sapere quanto ha realizzato per ciascun fondo alla fine della giornata. Alle 16,00 ha realizzato rispettivamente \$ 1500, \$ 3200, \$ 1200, \$ 2300 e \$ 2400 per i cinque fondi. Nella rimanente parte della giornata realizza rispettivamente per i cinque fondi riportati nell'ordine: \$ 100, \$ 500, \$ 300, \$ 200 e \$ 0. Scrivere un'unica istruzione APL per determinare le cifre realizzate in chiusura per ciascun fondo.
4. I cinque fondi del problema 3 realizzano rispettivamente \$ 7.30, \$ 11.58, \$ 3.45, \$ 2.17 e \$ 5.56 per azione. Quante azioni di ciascun fondo saranno state vendute?
5. L'amministratore ottiene le seguenti percentuali di commissione sui cinque fondi: 3.25, 2.5, 3.0, 3.75 e 3.5. Quanto viene a guadagnare per ciascun fondo nella giornata?
E quale è il totale dei guadagni per lo stesso giorno?

Soluzioni possibili

Problema 1:

```
      1 2 3 4 5*2
1 4 9 16 25
```

Problema 2:

```
      2 3*2
4 9
      2 3*3
8 27
      2 3*4
16 81
```

```
      2*2 3 4
4 8 16
      3*2 3 4
9 27 81
```

Problema 3:

```
1500 3200 1200 2300 2400+100 500 300 200 0
1600 3700 1500 2500 2400
```

Problem 4:

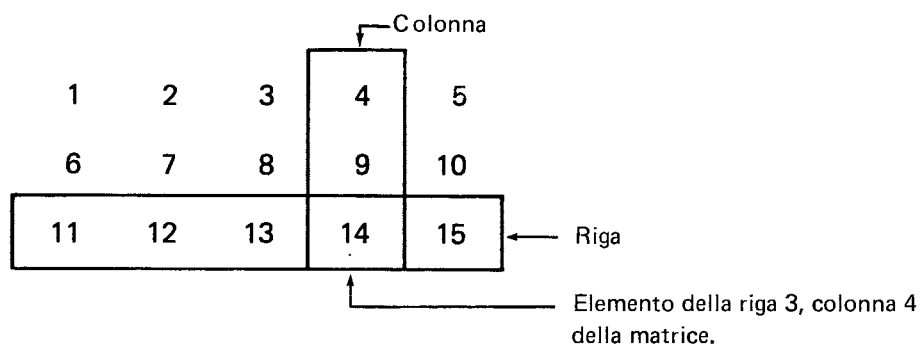
```
1600 3700 1500 2500 2400÷7.30 11.58 3.45 2.17 5.56
219.18 319.52 434.78 1152.1 431.65
```

Problema 5:

```
1600 3700 1500 2500 2400×.0325 .0250 .0300 .0375
52 92.5 45 93.75 84
52.00+92.50+45.00+93.75+84.00
367.25
```

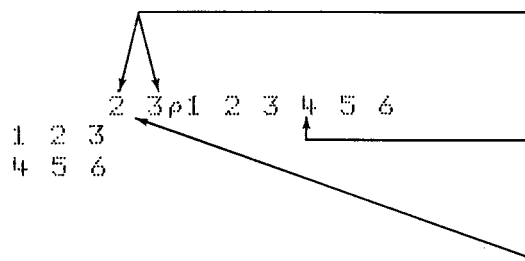
Uso del linguaggio APL con tabelle di numeri (Matrici)

Una tabella di numeri viene talvolta chiamata matrice. I numeri nella matrice sono ordinati in righe e colonne; ogni numero è chiamato elemento della matrice.



Per creare le matrici si fa uso della funzione di forma ℓ . L'argomento di sinistra specifica il numero delle righe e delle colonne, e l'argomento di destra specifica i dati o il nome di variabile per i dati da inserire nella matrice.

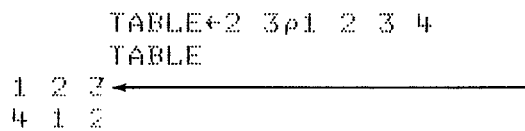
ESEMPI:



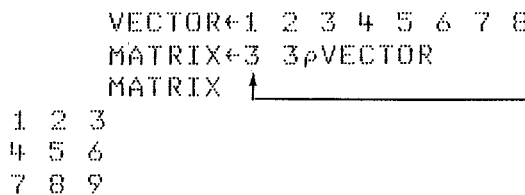
Il primo numero nell'argomento di sinistra specifica il numero di righe il secondo specifica il numero di colonne.

L'argomento di destra specifica i valori dei singoli elementi della matrice.

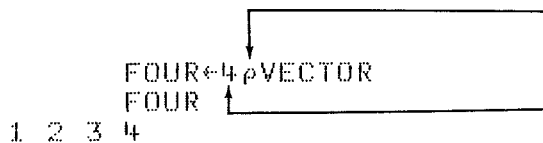
Ci deve essere uno spazio fra i numeri che specificano righe e colonne.



Se non ci sono abbastanza elementi nell'argomento di destra per riempire la matrice, gli elementi sono ripetuti.

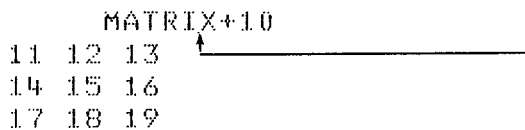


Se nell'argomento di destra ci sono più elementi di quelli richiesti per riempire la matrice; sono usati solo i primi (quelli più a sinistra).

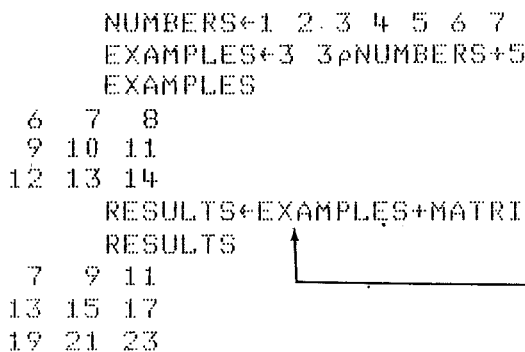


La funzione può essere usata anche per creare un vettore.

Numero degli elementi del vettore.



Si può variare ciascun elemento della matrice con un valore costante (si ricordi che il valore di MATRIX non è cambiato).



Si ricordi che l'APL esegue un'istruzione da destra a sinistra - il risultato di **NUMBERS + 5** è usato come argomento di destra per la funzione.

Si può operare su ciascun elemento della matrice per mezzo del corrispondente elemento in una matrice della stessa forma.

ESEMPI — continua

```

LESS+2 2*NUMBERS
LESS
1 2
3 4

MATRIX*NUMBERS ← Una matrice e un vettore.
RANK ERROR
MATRIX*NUMBERS oppure
^
MATRIX*LESS ← due matrici che non hanno le stesse
LENGTH ERROR dimensioni (numero di righe e co-
MATRIX*LESS lonne) non possono essere usate, a
^ meno che uno degli argomenti sia
una costante.

```

COME RIFERIRSI SOLO AD ALCUNI NUMERI IN UNA STRINGA O TABELLA DI NUMERI (INDIRIZZAMENTO)



E' possibile far riferimento solo a determinati elementi in una stringa o tabella specificando la posizione dell'elemento desiderato. I numeri usati per specificare la posizione degli elementi sono chiamati numeri indice. Questi numeri indice sono chiusi in parentesi quadre [] che seguono il vettore oppure la matrice alla quale si riferiscono

ESEMPI:

```

TEMP+68 74 78 65 80 85 72
TEMP[2] ← Si vuole far riferimento ad un sin-
74 golo elemento.
TEMP[3 1 2]
78 68 74 ← Si vuole far riferimento a più ele-
menti. Si tenga presente che gli ele-
menti sono visualizzati nell'ordine
con cui sono stati indicati.

TEMP[2]+TEMP[3 1 2]
152 142 148 ← Si vuole far riferimento agli elemen-
ti voluti e compiere altre operazio-
ni nella stessa istruzione.

TEMP[7]+88 ← Si vuole cambiare un singolo ele-
TEMP mento di un vettore.
68 74 78 65 80 85 88
TEMP[3 6]+70 ← Si vuole cambiare diversi elementi.
TEMP[3 6]
70 70

```

ESEMPI — continua

```

      TEMP[5 11+32 56
      TEMP
56 74 70 65 32 70 88

```

Si noti che i nuovi valori vengono attribuiti nello stesso ordine con cui sono stati indicati gli elementi voluti.

Per una matrice è necessario fornire fra parentesi un numero per la riga ed uno per la colonna — questi numeri sono separati da un punto e virgola.

```

      TIMES+3 3*NUMBERS
      TIMES
1 2 3
4 5 6
7 8 9

```

Si noti che era stato preventivamente assegnato un valore a NUMBERS.

La parte a sinistra del ; specifica la riga (o righe).

La parte a destra del ; specifica la colonna (o colonne).

Si può far riferimento ad un singolo elemento.

oppure

si vuole far riferimento a diversi elementi. In questo caso, si è fatto riferimento al secondo ed al terzo elemento della terza riga.

Si noti che quando si fa riferimento a più di una colonna e più di una riga, il risultato è una matrice.

```

      TIMES[2;2]
5
      TIMES[3;2 3]
8 9
      TIMES[1;1 2]
1 2
      TIMES[2 3;2 3]
5 6
8 9

```

ESEMPI — continua

```

      TIMESL2;J
4 5 6
      TIMESL;3J
3 6 9
      TIMESL;1 3J
1 3
4 6
7 9

```

Se non si specifica una colonna, si ottiene l'intera riga

Se non si specifica una riga, si ottiene l'intera colonna

Questi valori (la terza colonna) sono visualizzati orizzontalmente perchè essi costituiscono una stringa di numeri (vettori).

Nota: Anche quando si sceglie un'intera riga o colonna il punto e virgola è necessario per chiarire se il numero indica la riga o la colonna.

```

      TIMESL2;1J×TIMESL3;3J
36
      TIMESL2;2J+0
      TIMES
1 2 3
4 0 6
7 8 9
      TIMESL1;1J+2+3
      TIMESL1;1J
5

```

Si vuole far riferimento agli elementi voluti e compiere altre operazioni nella stessa istruzione.

Si vuole cambiare il valore degli elementi della matrice.

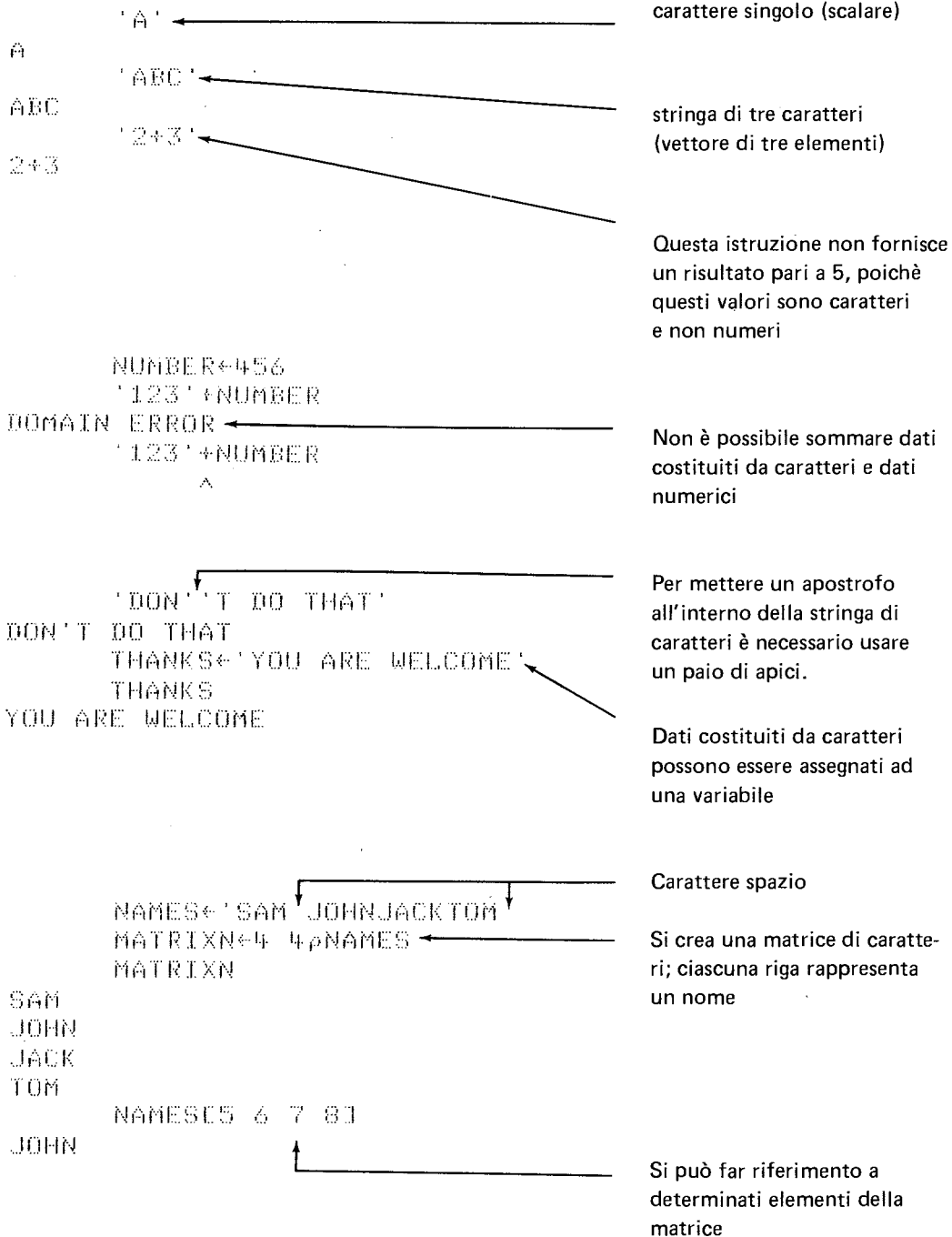
NON SI E' LIMITATI AD USARE SOLAMENTE NUMERI



Benchè gli esempi fino a questo punto abbiano usato solamente dati numerici, l'APL lavora anche con dati costituiti da caratteri. I dati costituiti da caratteri, ad esempio, possono essere usati per intestazioni su una tabella o per creare una lista di nomi. Allorchè si immettano dati costituiti da caratteri, bisogna includerli tra apici. Questi apici specificano che i dati sono dei dati caratteri e non nomi di variabile o numeri o funzioni. Allorchè un dato costituito da caratteri è visualizzato, gli apici non compaiono.

I dati costituiti da caratteri così come i dati numerici possono essere singoli caratteri (scalari), stringhe di caratteri (vettori), o tabelle di caratteri (matrici). Diversamente dai tasti numerici, allorchè si ha un vettore oppure una matrice di caratteri, ciascun carattere è un elemento a se stante e non va separato dagli altri elementi con uno spazio. In realtà uno spazio nei dati costituiti da caratteri rappresentante anch'esso un carattere (cioè un carattere vuoto).

ESEMPI:



Finora si è usato l'APL con alcune comuni operazioni aritmetiche. Si è visto inoltre come l'APL lavora con gli scalari (dati singoli), con i vettori (di dati) e con le matrici (tabelle di dati). Tuttavia non si è limitati solo alle funzioni descritte fino ad ora. Nei seguenti capitoli saranno presentate altre possibilità dell'APL.

Capitolo 3 - Funzioni APL che richiedono un argomento.

In questo capitolo si farà uso di alcune funzioni APL per:

- Determinare il numero intero più vicino ad una data frazione.
- Disporre gli elementi di un vettore in ordine crescente o decrescente.
- Generare un numero casuale.
- Determinare le dimensioni di una variabile.

Vi sono altre funzioni APL che richiedono un solo argomento; queste altre funzioni saranno discusse più oltre, nel Capitolo 6.

QUANTI ARGOMENTI SONO RICHIESTI DA UNA FUNZIONE INTRINSECA APL

In questo capitolo, si fa uso di funzioni APL con un solo argomento. Nel capitolo successivo si farà uso di alcuni degli stessi simboli di funzioni APL con due argomenti. Come si potrà notare, questi simboli eseguono funzioni APL allorchè vengono usati con uno oppure due argomenti. Allorchè si fa uso di una funzione APL con un solo argomento deve essere a destra del simbolo della funzione.

SIMBOLI DI FUNZIONI APL CHE SONO UNA COMBINAZIONE DI DUE CARATTERI

Alcuni simboli di funzione APL sono una combinazione di due caratteri. Si è visto che, quando si correggono degli errori di battuta, se si posiziona il cursore ad un dato carattere e se si preme un altro tasto, un carattere nuovo sostituisce quello precedente. Tuttavia alcuni simboli APL richiedono due caratteri, battuti, uno sull'altro. Per questi simboli si batte il primo carattere, si preme il tasto di spazio indietro e poi si batte il secondo carattere. Non ha importanza in quale ordine i due caratteri vengono battuti. I simboli che sono una combinazione di due caratteri sono chiamati caratteri sovrapposti. In Appendice A sono elencati i caratteri sovrapposti e quali tasti si usano per immetterli.

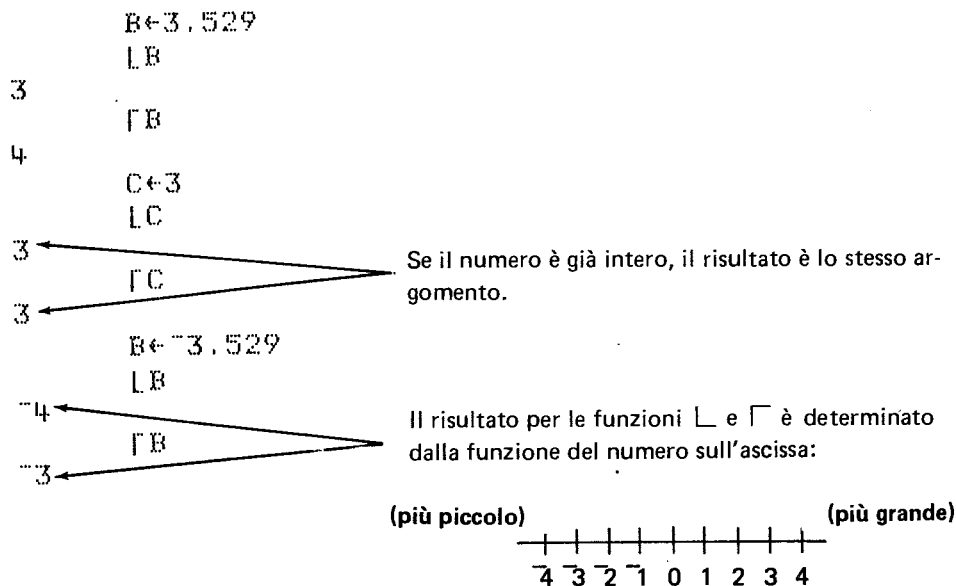
Nota: Se, dopo aver battuto un carattere sovrapposto, lo si vuole cambiare, basta posizionare il cursore sul carattere e battere un altro carattere. Il nuovo carattere sostituirà il carattere sovrapposto.

DETERMINAZIONE DEL NUMERO INTERO PIU' VICINO AD UNA DATA FRAZIONE



Quando si vuole trascurare la parte frazionaria di un numero e considerare solo la sua parte intera più vicina, si può far uso dei simboli funzionali $\lfloor$ e $\lceil$. Il simbolo funzionale $\lfloor$ arrotonda il numero per difetto, mentre il simbolo funzionale $\lceil$ arrotonda il numero per eccesso.

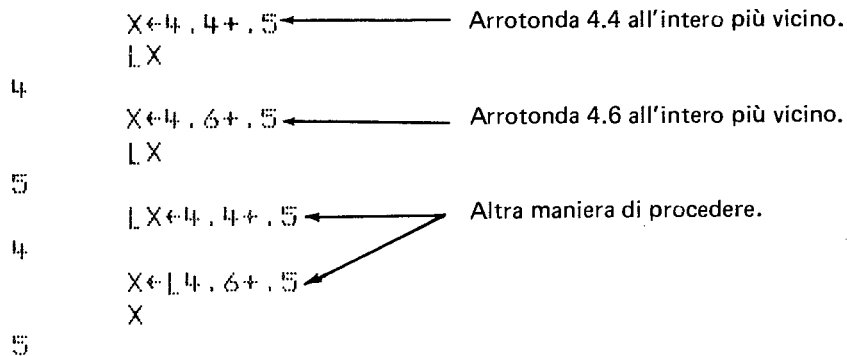
ESEMPLI:



Arrotondamenti al numero intero più vicino

E' pratica comune arrotondare i numeri all'intero più vicino. Si può fare ciò sommando al numero il valore 0.5 e poi usando la funzione

ESEMPLI:



DISPORRE GLI ELEMENTI DI UN VETTORE IN ORDINE ASCENDENTE O DISCENDENTE



I simboli funzionali Δ e ∇ vengono usati per disporre un vettore, in sequenza ascendente o discendente, in quanto queste funzioni forniscono gli indici di posizione degli elementi del vettore in ordine ascendente o discendente.

ESEMPI:

A ← 80 45 62 37 29 74 58 15 96
 ΔA
 8 5 4 2 7 3 6 1 9

← L'elemento maggiore è il nono.
 ← L'elemento minore è l'ottavo.

B ← A[ΔA]
 B
 15 29 37 45 58 62 74 80 96

← In questa maniera si dispongono gli elementi di A in ordine ascendente.

Si ricordi che quando si fa riferimento ad uno o più elementi di un vettore, i numeri indici o l'espressione indice deve essere posta fra parentesi.

∇A
 9 1 6 3 7 2 4 5 8
 C ← A[∇A]
 C
 96 80 74 62 58 45 37 29 15

← Gli elementi di A sono disposti in ordine discendente.

GENERAZIONE DI UN NUMERO CASUALE



Per generare un numero casuale si usa il simbolo funzionale `?`, il quale genera un numero casuale compreso tra 1 e il valore dell'argomento.

ESEMPI:

```

X←?6 ← Genera un numero fra 1 e 6.
X
5 ← Il risultato può essere qualunque numero fra 1 e 6.
X←?6 6 6 6 ← Quando questa funzione è usata con un vettore, viene
X           generato un unico casuale per ciascun elemento
1 3 1 3     del vettore.
  
```

GENERAZIONE DI NUMERI CONSECUTIVI



Vi sono casi in cui si vuol creare un vettore costituito da numeri consecutivi da un valore ad un altro valore. Ciò può essere fatto immettendo una istruzione come la seguente:

```

VECTOR←1 2 3 4 5 6 7 8
VECTOR
1 2 3 4 5 6 7 8
  
```

Tuttavia, è possibile usare il simbolo funzionale `↑`, che genera numeri consecutivi da 1 fino al valore specificato dell'argomento.

ESEMPI:

```

      ↑8
1 2 3 4 5 6 7 8 ← otto numeri consecutivi.
      VECTOR←5+↑5 ← si aggiunge 5 ad ogni numero consecutivo.
      VECTOR
6 7 8 9 10
      2*↑6 ← le prime 6 potenze di due.
2 4 8 16 32 64
  
```

Generazione di un vettore vuoto

Un settore vuoto è un vettore che non ha elementi. A cosa serve un vettore di questo tipo? Come si vedrà più avanti, in particolari circostanze (per esempio quando si adopera una funzione definita dall'utente) talvolta, può essere necessario generare un vettore vuoto. Una maniera di fare ciò è la seguente (1 0).

ESEMPL:

```

NAME
VALUE ERROR ← C'è un errore se si usa un nome di variabile a cui non è
NAME          stato assegnato precedentemente un valore
^

NAME+1 0 ← Generazione di un vettore vuoto
NAME
          Il risultato è una riga vuota (nessun valore)

```

DETERMINAZIONE DELLE DIMENSIONI DI UNA VARIABILE



Come si è appreso nel capitolo 2, l'argomento di sinistra della funzione ρ determina il numero di elementi di un vettore oppure il numero di righe e di colonne di una matrice. In questo modo il numero di elementi di un vettore o di una matrice è chiamato la *dimensione* del vettore o della matrice. Ad esempio, la forma della matrice M che ha 2 righe e 3 colonne è: 2 3. Per determinare le dimensioni di una variabile, si deve usare la funzione ρ .

ESEMPL:

```

SCALAR+4
VECTOR+2 4 6 8
MATRIX+2 3 p 6
pSCALAR
  ← riga priva di caratteri (la dimensione di uno
pVECTOR      scalare è un vettore vuoto).
4 ← numero di elementi sul vettore.
pMATRIX
2 3 ← numero di righe e colonne nella matrice.

EMPTY+1 0 ← generazione di un vettore vuoto.
EMPTY
  ← riga priva di caratteri.
pEMPTY
0 ← numero di elementi in un vettore vuoto.

```

Nel presente capitolo si farà uno di alcune funzioni APL che richiedono due argomenti. Si potranno usare queste funzioni per risolvere i seguenti problemi:

- Confrontare gli argomenti per determinare se uno è uguale, maggiore o minore dell'altro
- Elaborare variabili logiche vere (1°) e false (0°)
- Determinare il più grande di due numeri
- Determinare il più piccolo di due numeri.
- Determinare l'indice di un valore in un vettore.
- Generare una sequenza di numeri casuali
- Estrarre alcuni elementi da un vettore o da una matrice.
- Espandere un vettore od una matrice inserendo degli zero o degli spazi.
- Unire due vettori o matrici.
- Determinare il logaritmo di un numero

Esistono altre funzioni APL con due argomenti. Esse saranno discusse più avanti, nel capitolo 6.

FUNZIONI RELAZIONALI

Nel risolvere problemi con APL, può essere necessario determinare le relazioni fra due valori. Per esempio, può essere necessario controllare se un contatore ha raggiunto un certo valore; oppure operare in maniera differente a seconda che una certa condizione sia vera o falsa. Le seguenti funzioni APL sono usate per determinare le relazioni fra due valori:

Funzione	Simbolo	Tasto
Maggiore di	$>$	
Minore di	$<$	
Maggiore o uguale di	$\geq$	
Minore o uguale a	$\leq$	
Uguale a	$=$	
Non uguale a	$\neq$	

Quando queste funzioni vengono usate il 5100 valuta le relazioni fra i due valori e fornisce come risultato il valore 1 se la relazione è vera, il valore 0 se è falsa.

ESEMPI:

```

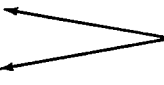
      A←10
      B←20
      A=B
0
      A=B-10
1
      A≤B
1
      A≠B
1
      A≥B
0

```

```

      A←'ABC'
      B←'DEF'
      A=B
0 0 0
      A≠B
1 1 1

```


 Gli operatori = e ≠ funzionano anche con dati costituiti da caratteri. Si noti che ogni elemento viene confrontato con l'elemento corrispondente all'altro argomento.

Perchè due numeri apparentemente identici non sempre sono uguali

L'APL memorizza tutti i valori numerici con una precisione interna di 15 cifre decimali: tuttavia i valori decimali con più di cinque cifre significative vengono normalmente arrotondati a cinque cifre prima di essere visualizzati. Pertanto talvolta numeri diversi possono apparire uguali quando vengono visualizzati.

ESEMPI:

```
A←1÷3
B←.33333
A
0.33333 ← Vengono visualizzate solo 5 cifre.
B
0.33333
A=B ← I valori sono diversi.
0
□PP ← □PP è una variabile del sistema che determina
5      quante cifre significative devono essere visualiz-
      zate. Questa variabile è posta a 5 automatica-
      mente quando l'elaboratore viene acceso, o vie-
      ne dato RESTART. (Le variabili del sistema
      sono trattate nel capitolo 9).
```



```
□PP←15 ← In questa maniera vengono visualizzate cifre si-
A      gnificative.
0.3333333333333333
B
0.33333 ← Si noti la differenza fra i due valori.
□PP←5 ← La variabile □5 viene nuovamente impostata a 5.
```

Si ricordi che il valore visualizzato può non essere il valore esatto memorizzato dal 5100.

Un esempio di uso delle funzioni relazionali

Si supponga che la risposta giusta di un problema è stata memorizzata come una variabile chiamata GIUSTA e la risposta fornita da uno studente è stata memorizzata come una variabile chiamata RISPOSTA. Si vuole aggiungere 1 al punteggio dello studente se la risposta è uguale alla risposta giusta; altrimenti, si vuole lasciare invariato il suo punteggio.

Se lo studente ha risolto il problema in maniera giusta, allora è vero che $RISPOSTA = GIUSTA$. Per sommare 1 al suo punteggio solamente se la sua risposta è uguale alla giusta risposta, si può immettere questa istruzione:

$$PUNTEGGIO \leftarrow PUNTEGGIO + RISPOSTA = GIUSTA$$

Allora il totale sommato a PUNTEGGIO è 1 quando RISPOSTA e GIUSTA sono uguali, è 0 quando sono diversi.

Supponiamo che invece di aggiungere 1 quando lo studente risponde esattamente, si voglia dare un peso maggiore ad alcuni problemi rispetto ad altri. Il peso del problema corrente è memorizzato sulla variabile PESO. Se lo studente risolve esattamente il problema, si aggiunge PESO al suo punteggio; viceversa si lascia il suo punteggio inalterato. Si può immettere la seguente istruzione:

$$PUNTAGGIO \leftarrow PUNTEGGIO + PESO \times RISPOSTA = GIUSTA$$

Se la risposta dello studente è uguale all'esatta risposta, allora $RISPOSTA = GIUSTA$ ha il valore 1, cosicchè il valore aggiunto è $PESO \times 1$. Ma se la risposta dello studente non è giusta, allora la quantità sommata è $PESO \times 0$, cioè 0.

FUNZIONI LOGICHE

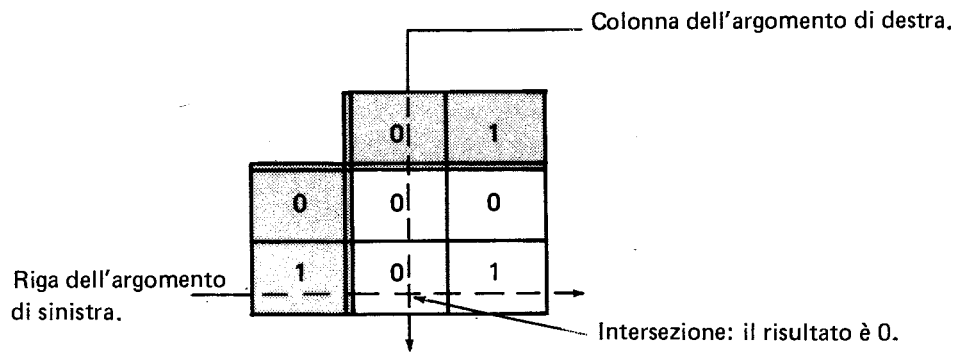
Le funzioni logiche usano come argomento solo i numeri 1 e 0 e vengono utilizzate per verificare determinate condizioni. (Normalmente verificano i risultati delle funzioni relazionali). Le funzioni logiche fondamentali sono:

Funzione	Simbolo	Tasto
And	$\wedge$	
Or	$\vee$	

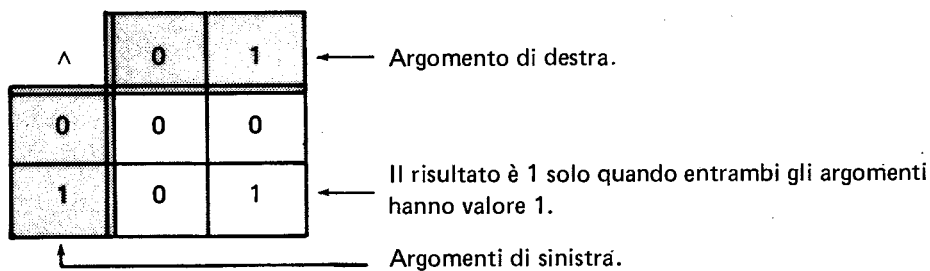
Trattando delle funzioni logiche, saranno utilizzate tabelle come quelle sotto riportate per mostrare i risultati possibili:

Funzione logica	$\wedge$	0	1	Valori dell'argomento di destra
	0	0	0	
	1	0	1	Risultati
		Valori dell'argomento di sinistra		

Per usare questa tabella basta prendere il valore dell'argomento di destra, posto in alto nella tabella, ed il valore dell'argomento di sinistra, posto nel lato sinistro. Poi si segue la colonna rappresentata dall'argomento di destra e la riga dell'argomento di sinistra. La loro intersezione è il risultato della funzione logica con quei valori, come argomenti. Per esempio, così si determina il risultato di $1 \wedge 0$:



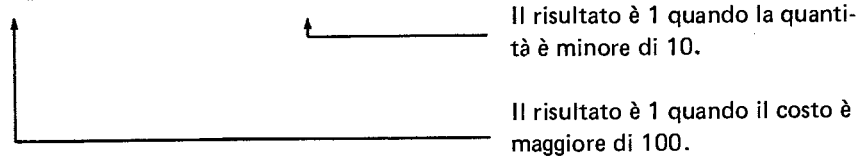
And



La funzione And si usa per controllare se due condizioni sono vere contemporaneamente.

Ad esempio, si voglia sapere quando gli articoli che hanno un costo maggiore di \$ 100.00 sono in quantità minore di 10. Si può usare la istruzione seguente:

$(COST > 100) \wedge (QUANTITY < 10)$



Si noti che le parentesi in questa istruzione specificano l'ordine di esecuzione.

ESEMPLI:

```

      QUANTITY←8
      COST←120
      (COST>100)^(QUANTITY<10)
1 ←----- Ambedue le condizioni sono vere.
      QUANTITY←25
      (COST>100)^(QUANTITY<10)
0 ←----- Almeno una condizione non è vera.

```

Or



	v	0	1	← Argomento di destra.
	0	0	1	← Il risultato è 1 se almeno uno degli argomenti (o tutti e due) vale 1.
	1	1	1	
				← Argomento di sinistra.

La funzione Or si usa per verificare se almeno una delle condizioni è vera.

Per esempio, si voglia sapere quando succede che la scorta per un certo articolo sia minore di dieci unità, oppure che le ordinazioni per lo stesso articolo superino la scorta. Si può usare la seguente istruzione:

```

      (ORDINAZIONI>SCORTA)∨(SCORTA<10)

```

Il risultato è 1 quando le ordinazioni sono maggiori della scorta.

Il risultato è 1 quando la scorta è minore di 10.

ESEMPI:

0	$INVENTORY \leftarrow 15$ $ORDERS \leftarrow 5$ $(INVENTORY < 10) \vee (ORDERS > INVENTORY)$	Ambedue le condizioni sono false.
1	$ORDERS \leftarrow 25$ $(INVENTORY < 10) \vee (ORDERS > INVENTORY)$	Almeno una condizione è vera.

Problemi: Uso delle funzioni relazionali e logiche

1. E' di vitale importanza inserire strumenti di controllo in ogni sistema spaziale per prevenire errori catastrofici. Ad esempio due strumenti che controllano un parametro si trovano nella stessa posizione. Se uno dei due, od ambedue gli indicatori avvertono di un pericolo, è necessario agire in maniera opportuna.

Si assuma che lo strumento A sia oltre il suo limite a 1.3725 ampères e che lo strumento B sia oltre il suo limite a 1.5365 ampères. Si costruisca una espressione che abbia come risultato 1 allorchè uno od ambedue gli strumenti abbiano oltrepassato il proprio limite: gli strumenti indicano rispettivamente 1.3732 e 1.5362.

2. Viene condotta una inchiesta dell'autorità scolastica, in cui l'insegnante ed i genitori degli alunni valutano dieci caratteristiche del bambino.

Un insegnante risponde con 1, 0, 1, 1, 0, 1, 0, 0, 1, 0 alle domande. I genitori dello stesso bambino rispondono 1, 0, 0, 1, 0, 1, 1, 0, 0, 0.

Mostrare a quali domande sia l'insegnante sia i genitori hanno risposto con 1.

Soluzioni possibili

Problema 1:

$$(1.5365 \leq 1.5362) \vee 1.3725 \leq 1.3732$$

1

Problema 2:

$$((10) \times 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \wedge 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0)$$

1 0 0 4 0 6 0 0 0 0

DETERMINAZIONE DEL PIU' GRANDE DI DUE NUMERI



Il risultato della funzione di massimo Γ è il più grande fra i 2 argomenti.

ESEMPI:

```

A←5
B←6
AΓB
6
36
(B×B)ΓA×A

```

Per vedere come usare la funzione di massimo Γ , si supponga che, in un deposito, ogni mese si debba calcolare l'ammontare addebitato e quello pagato per ogni cliente. Bisogna calcolare la differenza fra l'addebito totale ed il pagamento totale per ogni cliente: questa differenza viene memorizzata in una variabile chiamata BALDUE. Si addebita inoltre un interesse di 1,5% delle differenze stesse. E' possibile calcolare questo addebito nella seguente maniera:

```
CHARGE←BALDUE×.015
```

Tuttavia alcuni clienti pagano più del dovuto. Per essi, BALDUE è un numero negativo e deve essere interpretato come debito sui loro estratti conto mensili. Se si calcola l'interesse con l'istruzione precedente, la società che, gestisce il deposito dovrebbe pagare al cliente un interesse di 1,5%. Invece, si preferisce calcolare l'interesse come 1,5% su BALDUE o su 0, a seconda di quale dei due sia maggiore. Per far ciò, si può usare la seguente istruzione:

```
CHARGE←.015×0ΓBALDUE
```

CALCOLO DEL PIU' PICCOLO DI DUE NUMERI



Il risultato della funzione di minimo è il più piccolo tra due argomenti.

ESEMPI:

```

      A+5
      B+6
      A/B
5
      (B*B) L A*A
25
      C+1 4 7 9
      D+3 2 8 7
      C/D
1 2 7 7
    
```

Problemi: Uso delle funzioni di massimo e di minimo

1. Calcolare la maggior spesa in dollari per i seguenti acquisti di benzina:
 - a. 16.8 galloni a 52.9 cents per gallone
 - b. 13.5 galloni a 55.9 cents per gallone
 - c. 15.6 galloni a 57.9 cents per gallone
2. Calcolare la quantità minore di frutta secca acquistata ai seguenti prezzi:
 - a. 71 cents di nocciole a 33 cents per libbra
 - b. 53 cents di arachidi a 27 cents per libbra
 - c. 64 cents di mandorle a 29 cents per libbra

Soluzioni suggerite

Problema 1:

```

      (16.8*52.9) L (13.5*55.9) L 15.6*57.9
9.0324
    
```

Problema 2:

```

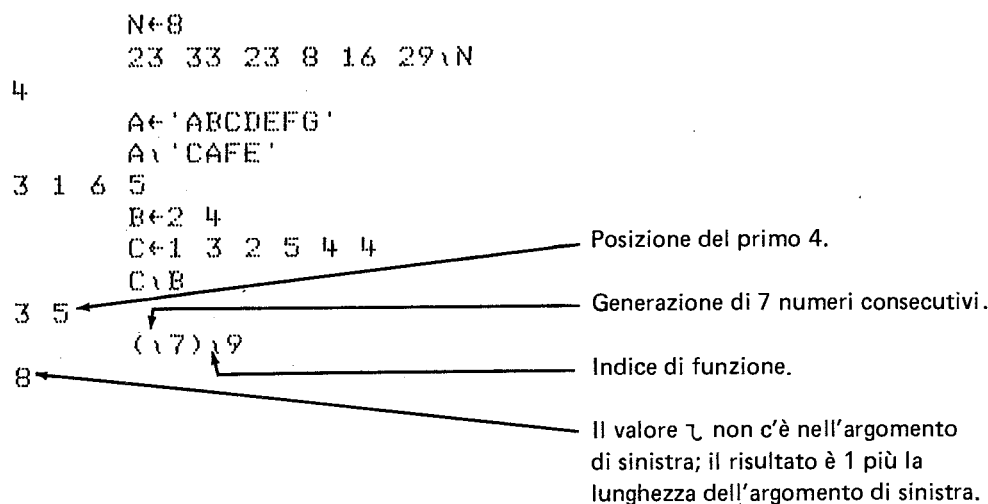
      (71÷33) L (53÷27) L 64÷29
1.963
    
```

DETERMINAZIONE DELL'INDICE DI UN VALORE IN UN VETTORE



Quando si vuol determinare se un numero è un elemento di un vettore, e se lo è, quale elemento è, si fa uso della funzione d'indice ι . L'indice della funzione fornisce la posizione (l'indice) della prima volta che, nell'argomento di sinistra, appare il valore dell'argomento di destra. Se il valore dell'argomento di destra non si trova nell'argomento di sinistra allora il risultato sarà la dimensione dell'argomento di sinistra più 1.

ESEMPLI:



GENERAZIONE DI UNA SEQUENZA DI NUMERI



Nel capitolo 3 si è usata la funzione $?$, con un solo argomento, per generare un numero casuale. Invece, usando la stessa funzione $?$, con due argomenti, si può creare una sequenza di numeri casuali, senza generare due volte lo stesso numero. Cioè la funzione $?$ genera tanti numeri casuali quanti sono specificati dall'argomento di sinistra, presi fra 1 ed il valore specificato dall'argomento di destra. I numeri casuali sono presi in modo che non vi siano mai numeri uguali. Perciò l'argomento di sinistra non deve mai avere un valore superiore a quello dell'argomento di destra. Se l'argomento di sinistra è uguale a quello di destra, si ottengono tutti i numeri da 1 fino a quello specificato dall'argomento di destra ma in ordine casuale.

ESEMPI

5 4 ← 2 7 5
 1 0 7 1 0
 5 3 8 1 0 6 9 2 4 1 7

Qualunque coppia di numeri differenti compresi fra 1 e 5.

Questi numeri possono essere in qualunque ordine, come può essere visto se questa istruzione viene immessa più di una volta.

SCELTA DI ALCUNI ELEMENTI di UN VETTORE O di UNA MATRICE (COMPRESSIONE)



E' possibile utilizzare la funzione di compressione per scegliere elementi da un vettore ad una matrice. L'argomento di sinistra deve essere un vettore composto da tutti 1 e 0, oppure una espressione che abbia come risultato un vettore di questo tipo. Quando si scelgono elementi di un vettore, il numero degli elementi in ciascun argomento deve essere lo stesso; i corrispondenti elementi dell'argomento di destra vengono estratti per ciascun 1 esistente nell'argomento di sinistra.

ESEMPI:

V ← 1 2 3 4 5 6 7 8
 1 0 1 1 0 1 1 1/V
 1 3 4 6 7 8
 0 0 0 0 0 0 0 0/V

Il risultato è un vettore vuoto.

A ← 1 0 0 0 0 0 0 1
 A/V
 1 8
 1 0 1 0/'ABCD'
 AC

Quando nello scegliere elementi da una matrice si debbano scegliere ed omettere intere righe e colonne, è necessario specificare la coordinata (righe o colonne) che deve essere attivata usando un valore d'indice [I]. Il valore d'indice è 1 per le righe e 2 per le colonne.

ESEMPI:

					B←3 4p\12	
					B	
1	2	3	4			
5	6	7	8			
9	10	11	12			

					0 1 0/E1]B	
5	6	7	8			
					1 0 1 0/E2]B	
1	3					
5	7					
9	11					

					1 0 1 0/B	
1	3					
5	7					
9	11					

Si ricordi che l'argomento di sinistra deve contenere un 1 per ogni elemento che deve essere scelto e 0 per ogni elemento che deve essere omissso.

E' specificata la prima coordinata (righe).

E' specificata la seconda coordinata (colonna).

Se non è specificato nessun indice, viene attivata la seconda coordinata (colonne).

ESPANSIONE DI UN VETTORE O DI UNA MATRICE



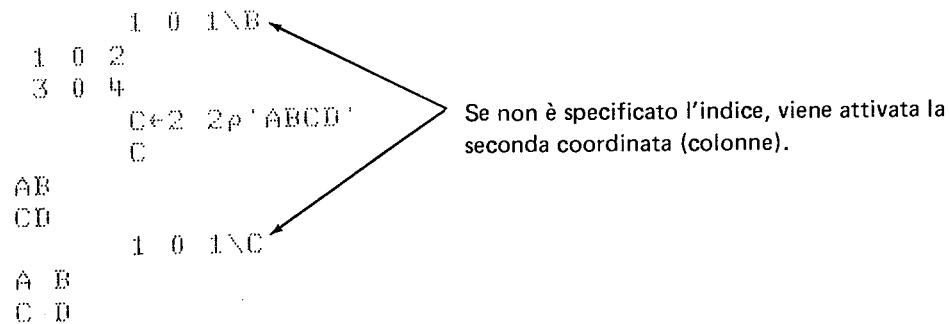
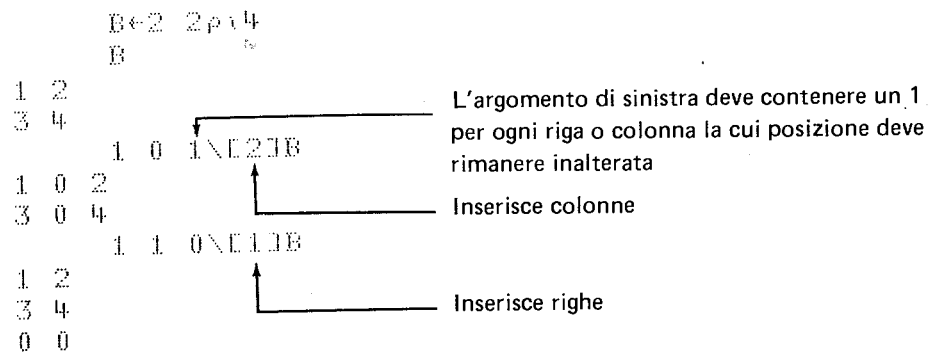
Si usa la funzione di espansione \ per inserire spazi o zeri in un vettore od in una matrice. L'argomento di sinistra deve essere un vettore composto da tutti 1 e 0, oppure una espressione che abbia come risultato, un vettore di questo tipo. Il numero degli 1 nell'argomento di sinistra deve essere uguale al numero di elementi dell'argomento di destra. Gli zeri nell'argomento di sinistra indicano dove sono inseriti gli spazi o gli zeri; gli spazi sono inseriti in un vettore o matrice di caratteri, mentre gli zeri sono inseriti in un vettore o matrice numerica.

ESEMPI:

1	2	0	3	0	4								
1	1	0	1	0	1	\	'ABCD'						
AB	C	D											

Nell'espandere una matrice vengono inserite intere righe o colonne di spazi o di zeri. Come capita allorchè si usa la funzione di compressione, bisogna specificare la coordinata (righe o colonne) utilizzando un indice [1]. Il valore di indice è 1 per le righe e 2 per le colonne.

ESEMPLI:



Problemi: Uso delle funzioni di compressione od espansione

1. Definire un vettore chiamato ACCTS che contenga cinque conti bancari: 56 103 100 13 0; estrarre quelli con saldi di \$ 100 o più.
2. Definire la matrice DATA ← 3 3p19. Poi inserire una riga in DATA, con i valori 20, 21 e 22, dopo la prima riga.

Soluzioni possibili

Problema 1:

```
ACCTS←56 103 100 13 0
(ACCTS≥100)/ACCTS
103 100
```

Problema 2:

```

DATA←3 3⍲9
DATA
1 2 3
4 5 6
7 8 9
DATA←1 0 1 1⍲1⍲DATA
DATA
1 2 3
0 0 0
4 5 6
7 8 9
DATA←2;1←20 21 22
DATA
1 2 3
20 21 22
4 5 6
7 8 9

```

UNIRE DUE VETTORI O DUE MATRICI



Si usa la funzione di concatenazione per unire insieme due vettori e per formarne uno solo; ciò si fa inserendo una virgola tra l'argomento di sinistra a quello di destra. Il numero di elementi nel vettore risultante è la somma del numero di elementi dei due vettori che sono stati uniti (concatenati).

ESEMPI:

```

A←1 2 3
B←4 5 6
A,B
1 2 3 4 5 6
B,A
4 5 6 1 2 3
C←'CAT'
D←'EN'
E←'ATION'
C,D,E
CATENATION

```

```

A,C
DOMAIN ERROR
A,C
A

```

Un vettore può essere formato da tutti numeri o da tutti caratteri, ma non è possibile unire dati numerici con dati di caratteri.

Si può altresì usare la funzione di concatenazione per unire due matrici tra loro. Per far ciò, bisogna usare un valore d'indice [I] per specificare quale coordinata deve essere estesa (cioè se bisogna aumentare il numero delle righe oppure, il numero delle colonne). Il valore indice è 1 se è la prima coordinata (numero delle righe) che deve essere estesa e 2 se è la seconda coordinata (numero di colonne) che deve essere estesa.

ESEMPI:

```

      A+2 2p14
      B+2 2p4+14
      A
1 2
3 4
      B
5 6
7 8
      A,[2]B ← Sono state aggiunte due colonne alle due già esistenti (aumento del numero di colonne).
1 2 5 6
3 4 7 8
      A,B ← Quando non viene specificata la coordinata, si intende l'ultima coordinata (colonne).
1 2 5 6
3 4 7 8
      A,[1]B ← In questo caso, sono state aggiunte due righe alle due già esistenti (aumento del numero di righe).
1 2
3 4
5 6
7 8

```

Quando si inseriscono due matrici, gli argomenti devono essere congrui. Cioè la lunghezza delle colonne deve essere la stessa, se sono le colonne ad essere concatenate, e la lunghezza delle righe deve essere la stessa se sono le righe ad essere concatenate.

ESEMPLI:

```

      A←2 2p4
      A
4 4
4 4
      C←2 3p6
      C
6 6 6
6 6 6

```

```

      A,C1J C
LENGTH ERROR ←
      A,C1J C
      ^

```

L'errore di lunghezza è stato causato dal fatto che A e C hanno righe di differente lunghezza.

A	4	4	
	4	4	
C	6	6	6
	6	6	6

```

      A,C2J C
4 4 6 6 6
4 4 6 6 6

```

Si noti che le matrici possono essere unite per colonne, poichè la lunghezza delle colonne è la stessa.

Costruzione di un vettore di risultati usando la concatenazione

Supponiamo di lavorare ad una serie di problemi e di voler accumulare le risposte. Una maniera per far ciò è di unire ciascun nuovo risultato ad un vettore di risultati ottenuti precedentemente. Se il risultato più recente è in una variabile chiamata LATEST e tutti i precedenti risultati sono in un vettore chiamato RESULT, si può usare la seguente istruzione:

```
RESULT ← RESULT, LATEST
```


Soluzione Possibile:

```

A←'I'
B←'T'
C←'D'
D←'R'
E←'GH'
F←'YO'
G←' '
H←'U'
F,H,G,C,A,C,G,A,B,G,D,A,E,B
YOU DID IT RIGHT

```

DETERMINAZIONE DEL LOGARITMO DI UN NUMERO



Si usa la funzione logaritmo $\circ$ per calcolare il logaritmo dell'argomento di destra nella base specificata dall'argomento di sinistra. Il logaritmo di un numero B in base A è quel numero che usato come potenza della base A, fornisce il numero B.

ESEMPI:

```

A←2
B←A*3
B
8
3
A∘B ←———— Logaritmo di B in base A.

```

Problemi: Uso della funzione logaritmo

1. Qual'è il logaritmo di 256 in base 2?
2. A che potenza bisogna elevare 10 per avere come risultato 100000?

Soluzioni Possibili:

Problema 1:

$$2^8 = 256$$

Problema 2:

$$10^5 = 100000$$

Capitolo 5. Applicazione delle stesse operazioni collettivamente a tutti gli elementi di un vettore (Riduzione)

E' sovente utile avere la somma (o il prodotto, o il massimo per esempio) di tutti gli elementi di un vettore. L'APL possiede una procedura semplice per applicare la stessa operazione collettivamente a tutti gli elementi di un vettore. Questa operazione è chiamata *riduzione*, dato che riduce un vettore numerico ad un singolo numero che rappresenta ad esempio la somma, il prodotto od il massimo. L'operatore di riduzione è `/`. L'argomento di sinistra è la funzione che viene applicata a tutti gli elementi di un vettore; il vettore è l'argomento di destra.

E' da tener presente che l'operatore di riduzione e la funzione di compressione hanno lo stesso simbolo. Tuttavia, si può comprendere la differenza tra la funzione di compressione e l'operatore di riduzione mediante l'argomento di sinistra. Per la funzione di compressione l'argomento di sinistra è un vettore formato da zeri e 1, invece per l'operatore di riduzione l'argomento di sinistra è una funzione APL incorporata.

RIDUZIONE

ESEMPI:

```
A←1 2 3 4 5
+/A
15
1+2+3+4+5 ← Sommare tutti gli elementi di A è lo stesso di
15          fare +/A.
```

Uso della riduzione somma per calcolare il valore medio.

L'operatore di riduzione è molto utile per calcolare il valore medio degli elementi di un vettore. Supposto che X sia il seguente vettore:

```
X←2 4 3 3 2.5 2
```

Si può usare la seguente istruzione per calcolare la media degli elementi di X:

```
AVG(+ / X) ÷ pX
AVG
2.75
```

Si analizzi la precedente istruzione:

1. Si calcola il numero di elementi di X (la lunghezza di X):

```
pX
6
```

2. Poi si calcola la somma degli elementi di X:

```
+ / X
16.5
```

3. Ora si può calcolare la media, dividendo 16.5 per 6:

```
AVG+16.5÷6
AVG
2.75
```

Problemi: Uso della riduzione somma

1. Utilizzando la riduzione calcolare l'importo medio che una certa famiglia spende ogni settimana in alimentari. Le note settimanali del fornitore, per il mese di Novembre sono state: \$ 31.05, \$ 29.78, \$ 25.44 e \$ 35.98.
2. In un laboratorio è stata misurata la temperatura di una soluzione ogni ora per dodici ore consecutive:

6	AM	:	75.8°	C
7	AM	:	71.9°	C
8	AM	:	77.0°	C
9	AM	:	80.3°	C
10	AM	:	85.1°	C
11	AM	:	82.2°	C
12	AM	:	83.2°	C
1	PM	:	84.9°	C
2	PM	:	85.3°	C
3	PM	:	85.0°	C
4	PM	:	82.5°	C
5	PM	:	80.9°	C
6	PM	:	78.4°	C

Calcolare la temperatura media.

Soluzioni possibili

Problema 1:

BILLS+31.05 29.78 25.44 35.98

AVG+(+/BILLS)÷pBILLS

AVG

30.563

Problema 2:

TEMP+75.8 71.9 77.0 80.3 85.1 82.2 83.2 84.9 85.3 85.0 82.
5 80.9 78.4

AVG+(+/TEMP)÷pTEMP

AVG

80.962

Uso della riduzione somma per sommare i prodotti di due vettori

Supponiamo che PRICE sia una variabile che contege la lista dei prezzi di alcuni articoli venduti da un magazzino e che Q1 e Q2 siano due vettori che indicano la quantità di questi articoli che due clienti hanno ordinato. Evidentemente il conto totale per il cliente 1 sarà la somma dei prodotti di PRICE per Q1 ed il conto totale per il cliente 2 sarà la somma dei prodotti di PRICE per Q2.

ESEMPI:

```
PRICE+.66 1.40 27.10 2.39 14.00 7.60 8.45 2.80
Q1+0 0 2 1 0 0 0 0
Q2+12 7 0 5 0 0 0 10
+ /Q1×PRICE
56.59
+ /Q2×PRICE
57.67
```

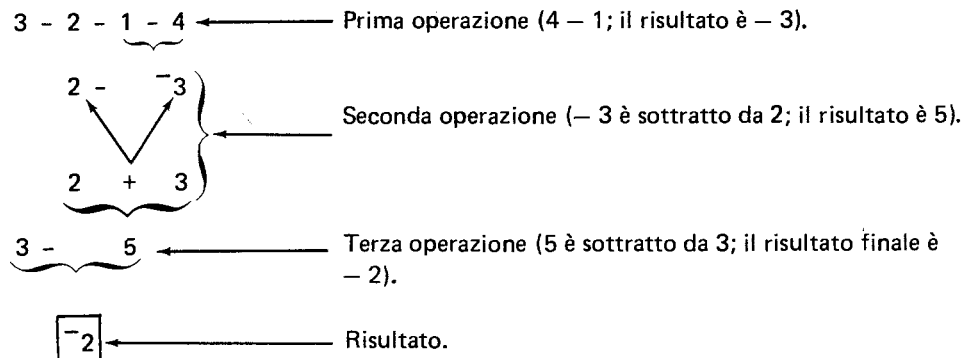
RIDUZIONE SOTTRAZIONE (SOMMA A SEGNO INVERTITO)

ESEMPIO:

```
A+3 2 1 4
-/A
-2
3-2-1-4 ← -/A equivale a questa istruzione
-2
```

Il seguente schema mostra perchè il risultato è - 2.

Il calcolo va da destra a sinistra.



RIDUZIONE DI MASSIMO: DETERMINAZIONE DEL VALORE PIU' GRANDE IN UN VETTORE

Per estrarre l'elemento più grande in un vettore, si riduce il vettore mediante la funzione Γ di massimo.

ESEMPI:

BALDUE+62.15 127 4.42 18.65 ← Importo dovuto da
Γ/BALDUE tutti i clienti di un
127 ← negozio
Maggior importo
dovuto

RIDUZIONE DI DETERMINAZIONE DEL VALORE PIU' PICCOLO DI UN VETTORE

Per estrarre l'elemento più piccolo di un vettore, si riduce il vettore mediante la funzione L di minimo.

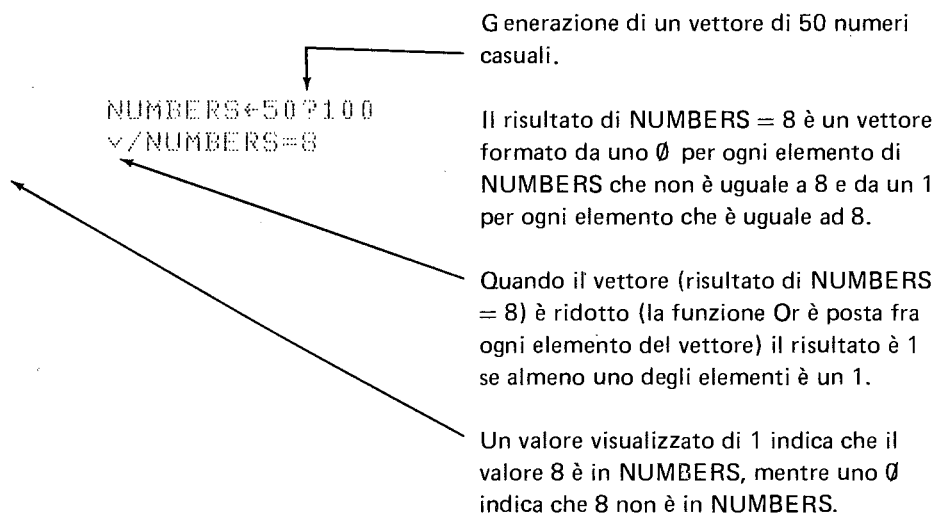
ESEMPI:

NUMBER+1 16 4 7 -9
L/NUMBER
-9

RIDUZIONE OR: RICERCA DI UNO SPECIFICO VALORE IN UN VETTORE

Supponiamo di voler sapere se esiste un determinato valore in un vettore. A tale scopo si usa la riduzione Or (v) per trovare la risposta.

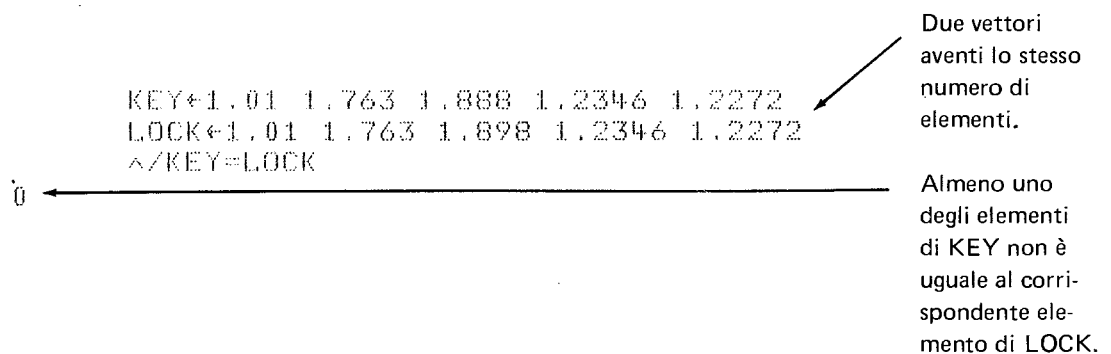
ESEMPLI:



RIDUZIONE AND: CONFRONTO FRA TUTTI GLI ELEMENTI DI DUE VETTORI

Per determinare se gli elementi corrispondenti di due vettori sono uguali, si usa la riduzione And $\wedge$.

ESEMPLI:



Questo capitolo contiene un sommario di ciò che si può fare mediante l'uso delle funzioni APL incorporate. Alcune delle funzioni sono già state discusse in capitoli precedenti mentre tutte le funzioni sono descritte nell'IBM 5100 APL Manuale di Riferimento, SA21 - 9213. In questo sommario si trova incluso un esempio per ciascuna funzione; si possono pertanto immettere questi esempi nel proprio 5100 per controllare come funzionano queste funzioni.

Nota: Gran parte di queste funzioni richiedono conoscenze matematiche particolari.

VEDIAMO ALLORA CIO' CHE SI PUO' FARE

Ciò che si può fare

Nome della funzione

Tasti

Funzioni APL che richiedono un argomento (consulta per maggiori informazioni il Capitolo 3)

- Estrazione per eccesso della parte intera di un numero

Massimo intero



5 6 ⌈4.68 6 ← Se il numero è già un numero intero, il numero stesso rappresenta il risultato

- Estrazione per difetto della parte intera di un numero

Minimo intero



4 2 ⌊4.68 2 ← Se il numero è già un numero intero, il numero stesso rappresenta il risultato

- Distribuzione di una serie di numeri in ordine crescente

Ordine crescente



5 3 1 2 4 ⍋A←3 7 2 9 1 ← Indici di A in ordine crescente
1 2 3 7 9 A[⍋A] ← Distribuzione di A usando gli indici.

Ciò che si può fare	Nome della funzione	Tasti
Distribuzione di una serie di numeri in ordine decrescente $\Psi A \leftarrow 3 \ 7 \ 2 \ 9 \ 1$ $A[\Psi A]$ $9 \ 7 \ 3 \ 2 \ 1$	Ordine decrescente Indici di A in ordine decrescente. Distribuzione di A usando gli indici.	 
Creazione di un numero a caso $?6$ 3	Trottola Il risultato può essere un qualsiasi numero tra 1 e 6.	
Creazione di una serie di numeri consecutivi $1 \ 2 \ 3 \ 4 \ 5$	Generatrice d'indice Creazione di una serie di cinque numeri consecutivi.	
Determinazione della lunghezza di una serie di numeri oppure del numero di righe e colonne in una tabella ρA 5 $\rho \text{MATRIX} \leftarrow 2 \ 3 \rho 16$ MATRIX $1 \ 2 \ 3$ $4 \ 5 \ 6$	Forma Lunghezza della stringa chiamata A. Crea una tabella e calcola la sua forma nella stessa istruzione (il numero di righe e di colonne). Funzione di formazione (discussa nel Capitolo 2). Funzione di forma.	

Ciò che si può fare

Nome della funzione

Tasti

Funzioni APL che richiedono due argomenti (consulta per maggiori informazioni il Capitolo 4).

Il risultato delle seguenti sei funzioni è 1 se la relazione espressa dalla funzione APL è vera; viceversa il risultato è 0

- Determinazione dell'uguaglianza di due valori

Uguale a



33=33

1

- Determinazione se l'argomento di sinistra è più grande dell'argomento di destra

Maggiore di



16>7

1

- Determinazione se l'argomento di sinistra è più piccolo dell'argomento di destra

Minore di



3<4

1

- Determinazione se l'argomento di sinistra è maggiore od uguale all'argomento di destra

Maggiore od uguale a



12≥11 12

1 1

- Determinazione se l'argomento di sinistra è minore od uguale all'argomento di destra

Minore od uguale a



6≤6 9

1 1

- Determinazione se due valori sono diversi

Diverso da



7≠77 7

1 0

Ciò che si può fare

Nome della funzione

Tasti

Le due funzioni logiche seguenti sono normalmente usate per verificare i risultati fra operazioni relazionali. I loro argomenti possono essere solo 1 e 0. Il risultato è 1 se la condizione che si controlla è verificata; se ciò non avviene, il risultato è zero.

- Determinare se tutte e due le condizioni sono vere

And



1 1 0
1 0

- Determinare se almeno una delle condizioni è vera

Or



1 1 0
1 1

- Calcolare il più grande tra due numeri

Massimo



5 4
5

- Calcolare il più piccolo tra due numeri

Minimo



5 4
4

- Calcolare l'indice di un determinato valore in un vettore

Indice di



9 8 7 6 5 4
3 ←

L'argomento giusto si trova al terzo posto dell'argomento di sinistra, il quale è un vettore.

- Creare un determinato numero di diversi numeri scelti a caso



3 7 6
2 3 1 ←

Possono essere tra numeri qualsiasi e diversi tra loro, compresi tra 1 e 6.

Ciò che si può fare	Nome della funzione	Tasti
Comprimere (estrarre alcuni determinati elementi da) un vettore od una matrice <pre> 2 5 1 0 0 1/2 3 4 5 </pre>	Compressione Vengono selezionati gli argomenti a cui corrisponde 1 nell'argomento di sinistra	
Espandere un vettore od una matrice <pre> 2 0 3 0 4 0 5 0 1 0 1 0 1 0 1 0 2 3 4 5 </pre>	Espansione Vengono inseriti degli elementi in accordo con gli zeri nell'argomento di sinistra	
Unire due argomenti assieme <pre> 'CAT', 'EN', 'ATION' CATENATION </pre>	Incatenamento	
Calcolare il log di un numero <pre> 3 2 8 </pre>	Logaritmo Log di 8 in base 2	
<i>Funzioni APL in aggiunta a quelle già discusse in capitoli precedenti (consulta l'IBM 5100 APL Manuale di Riferimento, SA21 - 9213 per maggiori informazioni).</i>		
Cambiare il segno di un numero <pre> -3 -4 3 4 </pre>	Negazione	
Determinare il segno di un numero <pre> -1 0 1 x 2 0 2 </pre>	Segno Il risultato è 1 per un numero negativo, 0 per 0 e 1 per un numero positivo.	
Calcolare il reciproco di un numero <pre> +3 0.33333 </pre>	Reciproco	

Ciò che si può fare	Nome della funzione	Tasti
Elevare e (2.71828) a una potenza	Esponenziale	 
$2.7183^{1/3} = 20.086$		
Calcolare il log di un numero di base e	log naturale	
$\ln(2.7183) = 20.086$		
Moltiplicare un numero per pigreco π 3.14159	Pigreco	 
$\pi \times 3 = 9.4248$		
Calcolare il prodotto tra loro di tutti i numeri compresi tra 1 ed il valore specificato (numero intero)	Fattoriale	
$4! = 24$	E' lo stesso risultato di $1 \times 2 \times 3 \times 4$.	
Cambiare in 0 l'1 ed in 1 lo 0	Negazione logica	
$\sim 1 = 0$		
Determinare se almeno una delle due condizioni è falsa	Nand	 
$1 \wedge 1 = 0$	Il risultato è 1 se almeno un argomento è 0 o viceversa il risultato è 0.	
$0 \wedge 1 = 1$		
Determinare se due condizioni sono false	Nor	 
$1 \vee 1 = 0$	Il risultato è 1 quando ambedue gli argomenti sono 0; viceversa il risultato è 0.	
$0 \vee 0 = 1$		

Ciò che si può fare

Nome della funzione

Tasti

- Trasformare una scalare od una matrice in un vettore

Intertrasformazione



```
MATRIX←2 3ρ16
MATRIX
1 2 3
4 5 6
,MATRIX
1 2 3 4 5 6
```

Matrice trasformata in un vettore

- Eseguire una serie di caratteri come espressione APL

Esecuzione



```
⊥'2x4'
8
```

- Convertire dati numerici in dati di caratteri

Formato



```
A←⊥24
A
24
ρA
2
```

L'uso di questa funzione con due argomenti è discussa nell'IBM 5100 APL Reference Manual, SA21 - 9213.

Questo è un valore di carattere

A è un vettore a 2 elementi (caratteri)

- Calcolare il valore di un numero senza riguardo al segno del numero

Valore assoluto



```
153 -46
53 46
```

- Invertire una matrice quadrata o calcolare la pseudo-inversa di una matrice rettangolare

Matrice inversa



```
⊞A←2 2ρ1 3 5 7
-0.875 0.375
0.625 -0.125
```

- Invertire gli elementi in un vettore od una matrice

Inversione



```
⊖'LIVE'
EVIL
```

Ciò che si può fare	Nome della funzione	Tasti
Calcolare il resto di una divisione	Residuo	
$\begin{array}{r} 318 \\ 2 \overline{) 8} \end{array}$ 2 ←	2 è il resto della divisione: 8 diviso 3.	
Calcolare i valori delle funzioni trigonometriche di un angolo	Circolare	
$\begin{array}{r} 308 \div 0 \div 4 \\ 1 \leftarrow \end{array}$	L'argomento di sinistra specifica la funzione trigonometrica (in questo caso tangente). Il risultato è la tangente di 45° ($\pi/4$ radianti).	
Calcolare il numero di combinazioni di un certo numero di elementi precedendone una certa quantità fissa ogni volta	Binomiale (combinazione)	 
$\begin{array}{r} 214 \\ 6 \overline{) 6} \end{array}$ 6 ←	Quattro elementi presi due alla volta possono fornire sei differenti combinazioni.	
Trovare se un determinato valore (l'argomento di sinistra) si trova (appartiene) in un vettore od una matrice	Appartenenza	
$\begin{array}{r} 1 \ 1 \ 0 \\ 'ABC' \in 'BANANA' \end{array}$ 1 ←	Il risultato è 1 se il valore dell'argomento di sinistra appartiene anche all'argomento di destra. Viceversa il risultato è 0.	
Esprimere un valore in un altro sistema numerico	Decodifica (Valore di base)	
$\begin{array}{r} 24 \ 60 \ 60 \ 11 \ 30 \ 15 \\ 5415 \leftarrow \end{array}$	Esprime 1 ora (3600 sec.) 30 minuti (1800 sec.) e 15 secondi tutto in secondo.	
Rappresentare (trasformare) un valore in un sistema numerico specificato	Codifica (rappresentazione).	
$\begin{array}{r} 24 \ 60 \ 60 \ 15415 \\ 1 \ 30 \ 15 \leftarrow \end{array}$	Trasforma 5415 secondi in ore, minuti, secondi.	

Ciò che si può fare	Nome della funzione	Tasti
Risolvere uno o più insiemi di equazioni lineari con il sistema delle matrici dei coefficienti	Divisione di matrici	 
$\begin{matrix} & & 26 & 9 & 8 & 2 & 2 & 3 & 5 & 1 & 2 \\ 7 & 1 & & & & & & & & & \end{matrix}$		
Prelevare un certo numero di elementi fra un vettore od una matrice	Prelevamento	
$\begin{matrix} & & 3 & \uparrow & A & + & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & \leftarrow & & & & & & & \end{matrix}$	Questi tre elementi sono prelevati dal vettore.	
Escludere un certo numero di elementi di un vettore od una matrice	Esclusione	
$\begin{matrix} & & 3 & \downarrow & A \\ 4 & 5 & \leftarrow & & & \end{matrix}$	Il risultato è espresso dagli elementi che restano dopo l'esclusione degli elementi specificati.	
Unire due argomenti tra loro formando un insieme con una dimensione in più	Laminare	
$\begin{matrix} & & 1 & 2 & 3 & 4 & 5 & , & [& 5 &] & 6 & 7 & 8 & 9 & 0 \\ 1 & 2 & 3 & 4 & 5 & \leftarrow & & & & & & & & & & \\ 6 & 7 & 8 & 9 & 0 & & & & & & & & & & & \end{matrix}$	Due vettori sono uniti per formare una matrice.	
Ruotare gli elementi in un vettore o matrice così come specificato dall'argomento di sinistra	Rotazione	 
$\begin{matrix} & & 2 & \Phi & 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 1 & 2 & \leftarrow & & & & & \end{matrix}$	Viene ruotato il vettore di due posizioni.	

Ciò che si può fare	Nome della funzione	Tasti
● Creare delle disposizioni di dati con almeno una dimensione (una disposizione con due dimensioni sia righe che colonne)	Formazione	P R

- Interscambiare le coordinate (come le righe e le colonne di una matrice od un insieme)

2	11
5	14
8	17

3	12
6	15
9	18

Trasposizione oppure
trasposizione generaliz-
zata

1

Quando viene usata con un solo argomento, questa funzione inverte le coordinate.

Operatori APL

Un operatore APL applica determinate funzioni incorporate ad un vettore o una matrice. L'operatore di riduzione è stato precedentemente discusso nel capitolo 5.

- Applicare la stessa operazione collettivamente a tutti gli elementi di un vettore.

Riduzione



	$+ / 1 \ 2 \ 3 \ 4 \ 5$	
15 ←	_____	Somma degli elementi.
	$\uparrow / 32 \ 6 \ 77 \ 19 \ 2$	
77 ←	_____	L'elemento più grande.
	$\downarrow / 32 \ 6 \ 77 \ 19 \ 2$	
2 ←	_____	L'elemento più piccolo.

- Applicare cumulativamente la stessa operazione ad ogni elemento di un vettore (il risultato di ciascuna operazione è usato nella successiva operazione)

Scansione



	$+ \backslash 1 \ 2 \ 3 \ 4$	
1	3	6
1		1
3		1+2
6		1+2+3
10		1+2+3+4

La funzione lavora nello stesso modo come se si immettessero queste istruzioni.

- Generare tabelle operative per varie funzioni APL e dati

Prodotto esterno



	$A \leftarrow 1 \ 2 \ 3 \ 4$	
	$A \circ . \times A$	
1	2	3
2	4	6
3	6	9
4	8	12

Tabella di moltiplicazione per numeri da 1 a 4.

Ciò che si può fare

Nome della funzione

Tasti

- Calcolare la matrice prodotto di due matrici

Prodotto interno



```
A←2 2p1 2 3 4
B←2 2p 5 6 7 8
A+.xB
```

```
19 22
43 50
```

Matrice prodotto delle matrici A e B.

COS'E' LA DEFINIZIONE DI UNA FUNZIONE?

Benchè il sistema APL contenga un gran numero di funzioni incorporate, può essere necessaria, per la soluzione di un problema, una funzione particolare. L'APL permette di definire una propria funzione (chiamata anche funzione definita dall'utente) e di memorizzarla per riusarla successivamente in caso di necessità.

COME SI DEFINISCE UNA FUNZIONE?



Si utilizzano le funzioni esistenti nel sistema APL per creare le nuove funzioni definite dall'utente.

Le nuove funzioni constano di:

Una intestazione, che contiene il nome della funzione ed altre informazioni (i diversi tipi di intestazioni sono trattati più avanti in questo capitolo).

Una istruzione oppure una serie di istruzioni che definisce la, o le, operazione (i) da svolgere.

Allorchè esegue una istruzione APL, l'IBM 5100 si trova nello stato di esecuzione; però, prima di definire una nuova funzione, bisogna cambiarne lo stato in definizione di funzione. Il simbolo ∇ si usa per far passare il 5100 da uno stato all'altro. Per esempio, per passare da stato di esecuzione a stato di definizione di funzione, si inserisce un ∇ come primo carattere nella intestazione; poi, dopo aver definito la funzione, si inserisce un altro ∇ per chiudere la definizione della funzione e ritornare nello stato di esecuzione.

Una volta riportato il 5100 nello stato di esecuzione, è possibile utilizzare la funzione definita dall'Utente.

Per spiegare ora con un esempio come si definisce una funzione, creiamo una funzione per il calcolo dell'ipotenusa di un triangolo rettangolo. L'istruzione, è a tale scopo, $((A^2) + (B^2))^*.5$, con cui si calcolano i quadrati dei lati A e B e la radice quadrata della loro somma, che rappresenta appunto il valore dell'ipotenusa. A questa funzione va assegnato un nome mediante il quale identificarla; questo nome sia HYP. Si immetta allora il carattere di apertura ∇ (per porre il 5100 nello stato di definizione di funzione) e la intestazione detta funzione:

∇ HYP←A HYP B ← Intestazione della funzione.

[1.] ← APL risponde con il numero della prima funzione da immettere.

Quando viene immessa la prima istruzione, compare sullo schermo il numero di quella successiva. Si inserisce allora il resto della funzione:

HP←((A*2)+(B*2))* .5 ← Istruzione.
 ▽ - riporta il 5100 nello stato di esecuzione.

E' da tener presente che i nomi nelle intestazioni (oltre quello della stessa funzione) sono tutti usati nel corpo della funzione. In particolare si deve notare come il nome della variabile risultato HP, è assegnato al risultato finale con una istruzione nella funzione.

Lo schermo mostrerà:

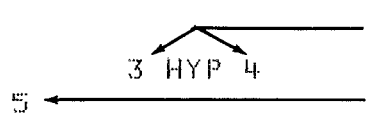
```
VHP←A HYP B
[1] HP←((A*2)+(B*2))* .5
[2] ▽
```

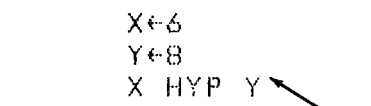
Nota: Se si commette un errore nell'inserimento di questa funzione, si deve consultare il paragrafo "Cosa fare dopo aver commesso un errore nella definizione di una propria funzione", che si trova più oltre in questo stesso capitolo.

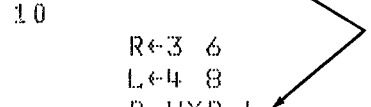
I tasti di arretramento di riga  (freccia verso l'alto) e di avanzamento di riga  (freccia verso il basso) non funzionano durante la stesura delle funzioni definite dall'utente.

Allorchè si immette il carattere di chiusura ▽, la funzione HYP viene memorizzata in modo da potersi usare come qualsiasi altra funzione APL con due argomenti.

ESEMPI:


 Lunghezza dei due lati.
 Lunghezza dell'ipotenusa.


 Come altre funzioni APL, gli argomenti possono essere fatti in forme differenti.


 5 10

Ogni volta che si vuole usare la funzione HYP, basta immettere il suo nome assieme agli argomenti scelti. Il simbolo per il calcolo dell'ipotenusa di un triangolo diventa così come il simbolo per l'addizione +.

Una funzione può contenere una sola istruzione, come HYP, oppure ne può contenere diverse.

ESEMPIO:

```

      VHP←A HYPL B
[1] A2←A*2
[2] B2←B*2
[3] S←A2+B2
[4] HP←S*.5V
      3 HYPL 4
5

```

La funzione HYP può anche essere definita in questo modo.

Si noti che il simbolo ∇ di chiusura può essere sulla stessa linea dell'ultima istruzione.

Stesso risultato di HYP.

Problemi: impiego della definizione di funzione

1. Definire la funzione che visualizza la somma di due numeri qualsiasi. Usare poi la funzione.
2. Definire la funzione che visualizza l'area di un rettangolo qualunque. Usare poi la funzione.

Soluzioni possibili

Problema 1:

```

      VS←M SUM N
[1] S←M+N∇
      6 SUM 3
9

```

Problema 2:

```

      VA←LENGTH AREA WIDTH
[1] A←LENGTH×WIDTH
[2] ∇
      4 AREA 5
20

```

CONTROLLO DELLA FUNZIONE PRIMA DELL'USO

Dopo aver definito una funzione, si dovrebbe sempre controllarla mediante l'uso di dati di cui è conosciuto il risultato finale. Supponendo per esempio che nella HYP venga usata per errore la seguente espressione:

```

((A*2)+(B*2))*.5

```

Avrebbe dovuto essere *

Si otterrà una risposta, ma questa non sarà la giusta risposta per il calcolo dell'ipotenusa di un triangolo rettangolo.

Quando si controlla una funzione, può verificarsi uno dei seguenti casi:

- Il 5100 fornisce il risultato che ci si aspetta.
- Il 5100 comunica un messaggio di errore.
- Il 5100 fornisce un risultato, ma non quello che ci si attende.
- Non succede nulla.

Se il 5100 fornisce il risultato che ci si aspetta

Bene! La funzione va bene.

Nota: Anche se la funzione ha funzionato bene una volta, è opportuno controllarla ancora per essere sicuri che funzioni in qualsiasi problema la si voglia applicare.

Se il 5100 comunica un messaggio di errore

Bisogna consultare l'IBM 5100 APL Manuale di Riferimento, SA21 - 9213 per determinare cosa specifica e cosa bisogna fare per correggerlo.

Nota: Una situazione di errore interrompe l'esecuzione della funzione; è opportuno consultare il Capitolo 8 per ulteriori informazioni su cosa fare quando si interrompe l'elaborazione di una funzione.

Se il 5100 visualizza un risultato diverso da quello che ci si attende oppure se non succede nulla

In entrambi i casi vi sono due alternative:

- Si deve visualizzare la funzione al completo e controllare la presenza di eventuali errori. Come visualizzare la funzione al completo è spiegato più avanti in questo capitolo.
- Si usano i dispositivi di TRACE e STOP (spiegati più avanti) come aiuto per risolvere il problema.

Nota: Quando si usa una funzione definita dell'utente e non accade nulla (cioè non compaiono, sullo schermo nè il risultato nè il cursore) oppure un risultato ripetuto continuamente, vuol dire che probabilmente la funzione è reiterata indefinitamente. In questo caso si preme il tasto ATTN per arrestare (sospendere) l'esecuzione della funzione. Il Capitolo 8 contiene informazioni su ciò che deve essere fatto quando la propria funzione si arresta.

COME USARE LE FUNZIONI DI TRACE E DI STOP



Trace T_{Δ}

Il dispositivo di Trace permette di controllare l'esecuzione della propria funzione istruzione per istruzione. Cioè viene visualizzato il risultato finale relativo a ciascuna istruzione controllata. Si può sia eseguire la Trace di tutte le istruzioni oppure solo di alcune. Per usare la funzione di Trace bisogna immettere T_{Δ} , il nome della funzione, $\leftarrow$ ed il numero delle istruzioni di cui si richiede la Trace. Per esempio:

TAEXAMPLE $\leftarrow$ 1 2 3 4 5 6

Numero della istruzione da controllare

Nome della funzione di cui fare le Trace

La precedente istruzione può anche scriversi nella seguente maniera:

TAEXAMPLE $\leftarrow$ 1 6

Genera un vettore con i numeri da 1 a 6

Stop S_{Δ}

La funzione Stop permette di arrestare l'esecuzione della propria funzione immediatamente prima dell'esecuzione di una determinata istruzione. Cioè, l'elaborazione della funzione è temporaneamente sospesa (la sospensione delle funzioni verrà discussa con maggiori dettagli nel capitolo 8). Dopo che è stata arrestata l'esecuzione della funzione, il 5100 visualizza il numero della istruzione che dovrà essere elaborata successivamente. Per usare la funzione di Stop, bisogna immettere S_{Δ} , il nome della funzione, $\leftarrow$, ed il numero delle istruzioni dopo le quali verrà arrestata l'esecuzione della funzione. Per esempio:

SAEXAMPLE $\leftarrow$ 3 6

Numero delle istruzioni specificate

Nome della funzione

Dopo che si è interrotta l'esecuzione di una funzione, si può riprendere con $\square$ LC, che è quella particolare *variabile del sistema* che contiene il numero dell'istruzione ancora da eseguire (Vedere cap. 9 per maggiori dettagli e l'IBM 5100 APL, Manuale di Riferimento, SA21 - 9213 per una completa descrizione delle *variabili del sistema*).

Proviamo ora ad usare i dispositivi di TRACE e STOP per risolvere un problema in una funzione.

ESEMPI:

```

VHP←A HYPX B
[1] 'THE HYPOTENUSE IS'
[2] A2←A*2
[3] B2←B*2
[4] S←A2+B2
[5] HP←S*.5V

```

Definisce una propria funzione che calcola l'ipotenusa di un triangolo rettangolo.

Questa funzione qui ha un errore.

Si provi la propria funzione usando dati di cui è noto il risultato. Il risultato dovrebbe essere 5.

```

3 HYPX 4
THE HYPOTENUSE IS
12.5

```

Uso della funzione di Trace per risolvere il problema

```

TAHYPX←2 3 4 5
3 HYPX 4
THE HYPOTENUSE IS
HYPX[2] 9
HYPX[3] 16
HYPX[4] 25
HYPX[5] 12.5
12.5

```

Il 5100 risponde con il nome della propria funzione, il numero della istruzione ed il risultato dell'istruzione di cui si esegue la Trace.

Il risultato è giusto in ogni istruzione, tranne che nell'istruzione 5; perciò, esso è probabilmente errato.

```
TAHYPX←10
```

Per disinserire la funzione di Trace, si usi 10 come specifica di cui si richiede la Trace.

```
SAHYPX←4 5
```

Uso della funzione di Stop per risolvere il problema

```

3 HYPX 4
THE HYPOTENUSE IS

```

```
HYPX[4]
```

Il 5100 risponde con il nome della propria funzione e con il nome della specifica che deve essere successivamente elaborata.

Quando la propria funzione viene arrestata si possono immettere le variabili per controllare se contengono i valori che ci si aspetta

```
A2
```

```
9
```

```
B2
```

```
16
```

```
→□LC
```

Si prosegue l'esecuzione immettendo →□LC.

```
HYPX[5]
```

```
S
```

L'esecuzione si arresta alla specifica successiva a quella specifica per la funzione di STOP.

```
25
```

```
→□LC
```

```
12.5
```

Tutte le variabili contengono i valori corretti; perciò, l'errore deve essere nella specifica 5.

```
SAHYPX←10
```

Per disinserire il dispositivo di Stop, si usa 10 come istruzione cui si chiede lo Stop

Nota: La correzione di un errore in una funzione è discussa più avanti.

COSA FARE QUANDO SI COMMITTE UN ERRORE NELLA DEFINIZIONE DI UNA PROPRIA FUNZIONE

Se si commette un errore nel definire una propria funzione, si può correggerlo variando la funzione. Nel variare una funzione si può fare quanto segue:

- Visualizzare l'intera funzione.
- Aggiungere una o più istruzioni alla fine della funzione.
- Sostituire le istruzioni.
- Inserire una o più istruzioni.
- Cancellare una istruzione dalla funzione.
- Visualizzare una determinata istruzione, oppure da una determinata istruzione fino alla fine della funzione.
- Modificare una singola istruzione.

Se ci si accorge di aver commesso un errore allorché si sta definendo una propria funzione, si può correggerlo senza definire nuovamente la funzione (Il 5100 è già sullo stato di definizione della funzione). Se invece la definizione della funzione è stata chiusa bisogna dapprima riapirla. Per far ciò, bisogna immettere la ∇ seguita solo dal nome della funzione. Se viene immessa l'intera intestazione della funzione, si avrà un messaggio di errore.

Ora, si definisca una funzione immettendo le seguenti istruzioni:

```
VSTAT X
[1] N←pX
[2] (+/X)÷N
[3] L/X
[4] I/XV
```

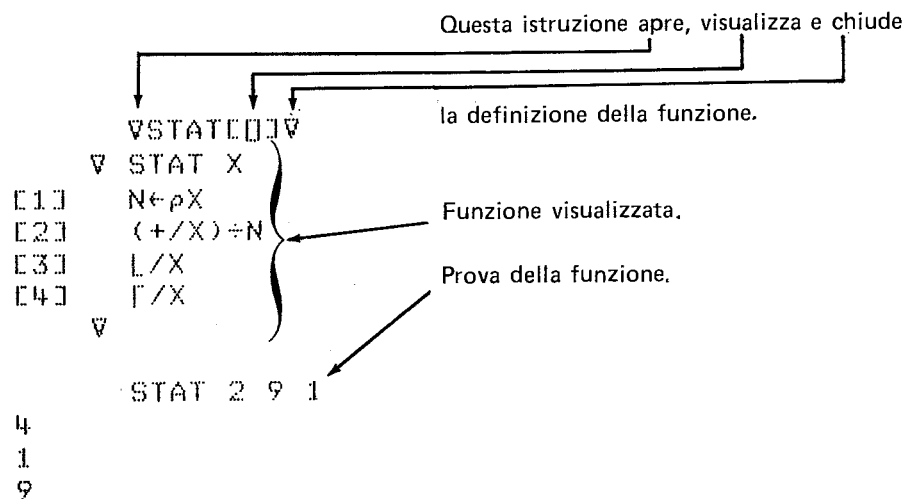
Questa funzione calcola la media, il minore e il maggiore degli elementi di un vettore. Si noti che questa funzione non ha una variabile risultato nell'intestazione: tuttavia anche così essa funziona. La ragione dell'uso di un variabile risultato di una media funzione sarà discussa più avanti.

Visualizzazione dell'intera funzione



Per visualizzare una funzione, si emetta `[□]` immediatamente dopo qualunque numero di istruzione, oppure come mostrato nel seguente esempio:

ESEMPIO:



Aggiunta di una o più istruzioni alla fine della funzione

Per aggiungere istruzioni ad una funzione, si apre la definizione della funzione e viene visualizzato il numero della prima linea disponibile. A questo punto è possibile introdurre l'istruzione che si vuole aggiungere.

ESEMPIO:

▽STAT ← Apertura della funzione.

[5] (Γ/X)-L/X▽ ← Si aggiunge questa istruzione per determinare la differenza fra il massimo ed il minimo valore degli elementi del vettore. Il simbolo ▽ chiude la funzione (si aggiunge solo una linea).

Il 5100 visualizza il numero della prima linea disponibile.

▽STAT [0]▽ ← Visualizzazione della funzione.

▽ STAT X

[1] N←ρX

[2] (+/X)÷N

[3] L/X

[4] Γ/X

[5] (Γ/X)-L/X

▽

Funzione visualizzata.

STAT 9 2 1 ← Prova della funzione.

4

1

9

8

Sostituzione di istruzioni all'interno di una funzione

Per sostituire una istruzione, è necessario racchiudere fra parentesi quadre [] il numero dell'istruzione da sostituire, seguito dalle nuove istruzioni.

ESEMPIO:

Questo sistema apre e visualizza la funzione.

```

VSTAT [ ]
  ▽ STAT X
[1] N←pX
[2] (+/X)÷N
[3] L/X
[4] I/X
[5] (I/X)-L/X
  ▽

```

Funzione visualizzata.

Si noti che è possibile specificare un altro numero di istruzione racchiudendolo fra parentesi quadre. Ora, si sostituisce la istruzione 2 con questa istruzione per determinare la media. Il simbolo ▽ chiude la funzione.

Il 5100 visualizza il numero della prima linea disponibile.

```

[6] [2] (+/X)÷pX▽

```

Visualizza la funzione modificata.

```

VSTATE[]▽
  ▽ STAT X
[1] N←pX
[2] (+/X)÷pX
[3] L/X
[4] I/X
[5] (I/X)-L/X
  ▽

```

Funzione visualizzata.

```

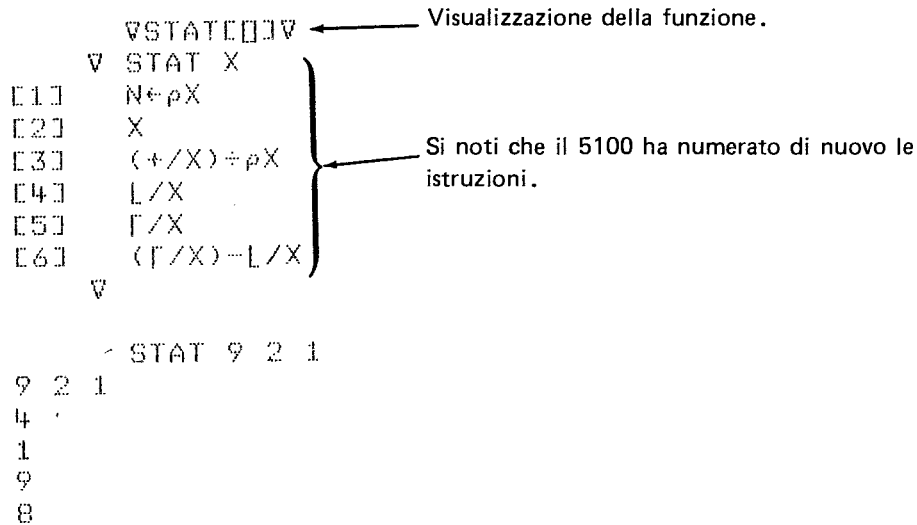
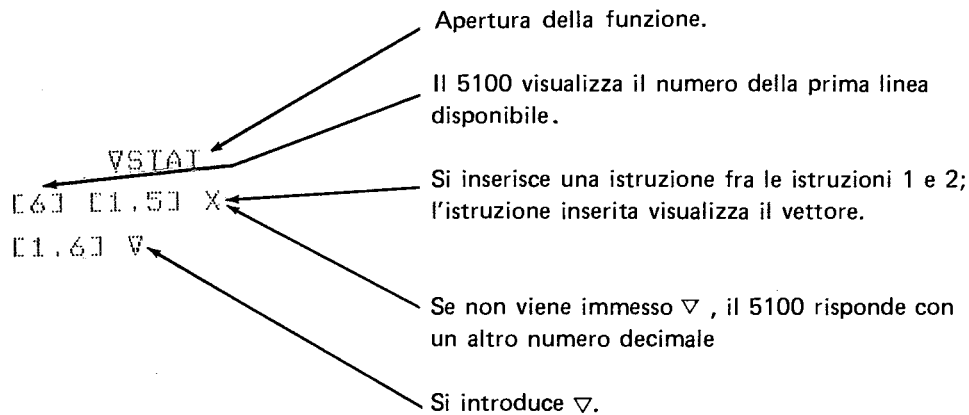
STAT 9 1 2
4
1
9
8

```

Inserimento di una o più istruzioni in una funzione

Per inserire istruzioni in una funzione, bisogna usare come numero della istruzione un numero decimale compreso fra i due numeri che indicano le istruzioni fra cui si vuol inserire le nuove istruzioni. Per esempio, per inserire una istruzione fra le istruzioni 1 e 2, si può usare il numero 1.5 o qualunque altro numero decimale compreso fra 1 e 2.

ESEMPIO:



Cancellazione di una istruzione da una funzione



Per cancellare una istruzione da una funzione, si introduce $[\Delta n]$, dove n è il numero dell'istruzione che si vuole cancellare.

ESEMPIO:

```

VSTATE[] ← Apertura e visualizzazione della funzione.
V STAT X
[1] N ← pX
[2] X
[3] (+/X) ÷ pX
[4] L/X
[5] I/X
[6] (I/X) - L/X
V
[7] [Δ4] ← Cancellazione dell'istruzione 4: non si vuole più
[5] V conoscere il numero più piccolo.

```

Funzione visualizzata.

Il 5100 visualizza il numero della prossima linea.

Nota: Il simbolo ∇ di chiusura *non* deve essere immesso nella stessa linea di $[\Delta n]$: bisogna immetterlo in un'altra linea, altrimenti apparirà un messaggio di errore.

```

VSTATE[]V ← Visualizzazione della funzione modificata.
V STAT X
[1] N ← pX
[2] X
[3] (+/X) ÷ pX
[4] I/X
[5] (I/X) - L/X
V
STAT 2 9 1
2 9 1
4
9
8

```

Funzione visualizzata — la linea 4 originale è stata cancellata e le istruzioni sono state nuovamente numerate.

Visualizzazione di una determinata istruzione, oppure da una determinata istruzione fino alla fine della funzione

Si è già visto come visualizzare l'intera funzione; si può anche visualizzare solo una istruzione, oppure ogni istruzione da una certa istruzione in poi, fino alla fine della funzione. Per visualizzare una istruzione, si introduce $[n \square]$, dove n è il numero della istruzione che si vuol visualizzare. Per visualizzare ciascuna istruzione da una certa istruzione in poi, si introduce $[\square n]$, con cui ogni istruzione, dalla n fino alla fine, viene visualizzata.

ESEMPIO:

```

VSTATE[3]V ← Visualizza l'istruzione 3.
[3]    (+/X)÷pX

VSTATE[4]V ← Visualizza ciascuna istruzione della 4 fino alla
[4]    [ /X                               fine della funzione.
[5]    ([ /X)-L/X
```

Modifica di una singola istruzione

Si possono correggere errori di immissione in un'istruzione di una funzione nello stesso modo in cui si correggono gli errori di immissione fatti durante la battitura nello stato di esecuzione. Cioè, sono usate le stesse procedure per l'inserzione, la cancellazione o la sostituzione di caratteri. Nel correggere errori di battuta nello stato di definizione di funzione, si può correggere l'errore o introducendo ex-novo l'intera istruzione, oppure visualizzando dapprima l'istruzione e quindi operando su essa.

ESEMPIO:

```

VSTATE2[] ← Apre la funzione e visualizza l'istruzione 2.
[2] X
[2] N ← Immettere una N per sostituire la X nella linea vi-
[3] ∇      sualizzata. (Ora si vuol conoscere il numero di ele-
      menti del vettore).
      Il 5100 risponde con [3]; si immette ∇ di chiusura.

```

```

VSTATE[] ∇ ← Visualizza la funzione.
∇ STAT X
[1] N ← ρX
[2] N
[3] (+/X) ÷ ρX ← N ha sostituito la X.
[4] ⍒/X
[5] (⍒/X) - L/X
∇

STAT 2 9 1

3
4
9
8

```

Come variare l'intestazione della funzione

E' possibile variare l'intestazione della funzione nella stessa maniera con cui si varia qualunque altra istruzione della funzione. Per far ciò, si specifica l'istruzione 0 come quella da variare.

ESEMPIO:

```

VSTATE0]STAT1 X ∇ ← L'intestazione originale è sostituita
VSTATE1[] ∇      da questa intestazione.
∇ STAT1 X
[1] N ← ρX
[2] N
[3] (+/X) ÷ ρX
[4] ⍒/X
[5] (⍒/X) - L/X
∇

```

Visualizza la funzione.

Nota: Non ci si preoccupi a questo punto se viene visualizzato il messaggio di errore SI DAMAGE: questo messaggio di errore e la risposta dell'utente suggerite sono descritte sul "IBM 5100 APL Reference Manual, SA21 - 9213".

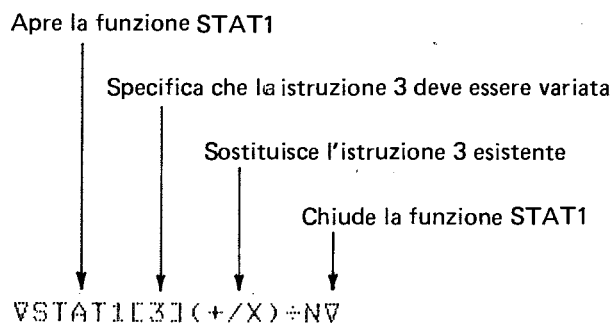
```

VSTATE[] ∇ ← Non si può visualizzare la funzione
DEFN ERROR STAT perchè essa non ha più
VSTAT      questo nome.
      ^

```


Una via più rapida, per aggiungere, sostituire o inserire una istruzione in una funzione

Se la propria funzione è chiusa e si deve aggiungere, sostituire o inserire solo una istruzione, si può usare una sola istruzione. Per esempio, la seguente istruzione apre, cambia e chiude la definizione di funzione:



ESEMPIO:

```

      VSTAT1[0]V ← Visualizza la funzione STAT1.
V STAT1 X
[1] N←pX
[2] N
[3] (+/X)÷pX
[4] I/X
[5] (I/X)-L/X
V

      VSTAT1[6]'THIS STATEMENT WAS ADDED 'V
      ← Aggiunge una istruzione alla funzione.

      STAT1 2 9 1 ← Prova la funzione.

3
4
9
8
THIS STATEMENT WAS ADDED

      VSTAT1[3](+/X)÷NV ← Sostituisce una istruzione.
      VSTAT1[3.5]L/XV ← Inserisce una istruzione.
      VSTAT1[0]V ← Visualizza la funzione modificata.
V STAT1 X
[1] N←pX
[2] N
[3] (+/X)÷N
[4] L/X
[5] I/X
[6] (I/X)-L/X
[7] 'THIS STATEMENT WAS ADDED
V

      STAT1 2 9 1

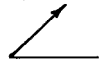
3
4
1
9
8
THIS STATEMENT WAS ADDED

```

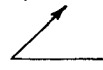
TIPI DI INTESAZIONI DI FUNZIONI

Come le funzioni APL incorporate, ci possono essere funzioni definite dall'utente con uno o due argomenti. Ci possono anche essere funzioni definite dall'utente senza alcun argomento. Il numero di argomenti richiesto da una funzione è definito dall'intestazione. Per esempio:

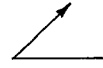
▽ RISULTATO ← ARGOMENTO1 FUNZIONE ARGOMENTO2

 Questa funzione richiede due argomenti.

▽ RISULTATO ← FUNZIONE ARGOMENTO

 Questa funzione richiede un argomento.

▽ RISULTATO ← FUNZIONE

 Questa funzione non ha argomenti.

Quando una funzione viene eseguita, il valore usato per un argomento è assegnato al nome della variabile che appare come argomento nell'intestazione della funzione. Questa variabile è poi usata nella funzione. Per esempio, si può avere la seguente funzione:

```
VR←A DIVIDE B
[[[ R←A÷BV
```

Se si immette 10 DIVIDE 2, il valore 10 è assegnato ad A ed il valore 2 è assegnato a B. Ora quando l'istruzione $A \div B$ è eseguita, il risultato è 5.

Nota: Per alcune funzioni definite dall'utente (come per alcune funzioni incorporate) è importante immettere gli argomenti nell'ordine opportuno. Per esempio, se si immette 2 DIVIDE 10, il risultato sarebbe 0.2 invece di 5.

Quando si definisce una funzione con un argomento, l'argomento deve essere a destra del nome della funzione; altrimenti, l'argomento verrà considerato il nome della funzione, e viceversa.

ESEMPI:

<pre> VR←A AREA1 B [1] R←A×BV 12 AREA1 12 144 </pre>	← Due argomenti—questa funzione determina l'area di un rettangolo.
<pre> VR←SQRT X [1] R←X×.5V A←1 4 9 16 25 36 SQRT A 1 2 3 4 5 6 </pre>	← Un argomento—questa funzione determina la radice quadrata di un numero. ← L'argomento può essere un vettore.
<pre> VR←DICE [1] R←?6 6 [2] V DICE 1 5 DICE 3 4 </pre>	← Nessun argomento—Questa funzione simula il lancio di due dadi. ← Il risultato può essere qualunque coppia di numeri compresi fra 1 e 6.

PERCHE' SI USA UNA VARIABILE RISULTATO?

Finora, nel trattare le funzioni definite dall'utente, di solito sono state definite funzioni aventi una variabile risultato. Una variabile risultato è un nome di variabile sotto il quale il risultato di una funzione è temporaneamente memorizzato per l'uso in una istruzione APL. Quando la funzione definita dall'utente ha una variabile risultato, si dice che essa ha un risultato esplicito. Senza un esplicito risultato, la funzione definita Non può essere usata in una espressione APL.

La seguente funzione ha una variabile risultato; perciò, essa ha un risultato esplicito.

```

      VR←QTY ITEMX COST
[1] RESULT←COST×QTY VR

```

Risultato variabile
 ↑
 Risultato variabile

La variabile deve apparire sia nell'*intestazione della funzione* sia nel corpo della funzione (deve essere cioè inclusa nell'istruzione in cui il risultato finale viene determinato).

ESEMPI:

```

▽QTY ITEM COST
C1] COST←QTY▽
  10 ITEM .60
0.06

```

Definisce una funzione senza un risultato esplicito.

```

0.06 STORE←10 ITEM .60
VALUE ERROR
STORE←10 ITEM 0.6
^
10+10 ITEM .60

```

Il risultato della funzione non può essere usato in espressioni APL.

```

0.06
VALUE ERROR
10+10 ITEM 0.6
^

```

```

▽RESULT←Q ITEM C
C1] RESULT←C←Q▽

```

Definisce una funzione con un risultato esplicito.

```

10 ITEM .60
0.06

```

```

STORE←10 ITEM .60
STORE
0.06
10+10 ITEM .60
10.06

```

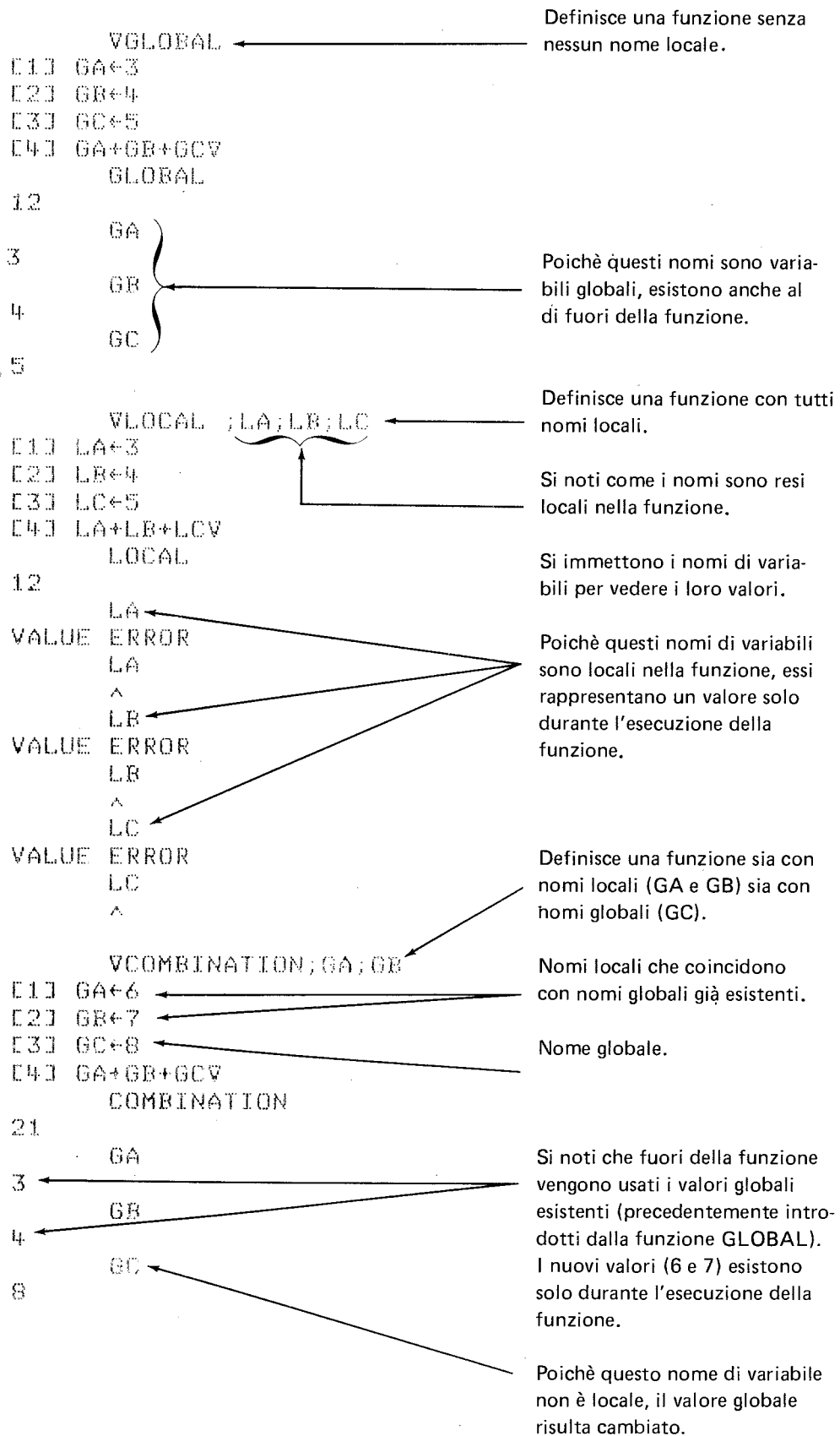
Il risultato della funzione può ora essere usato in una espressione APL.

Si ricordi che se si vuole usare la propria funzione in calcoli successivi, è necessario assegnare ad essa una variabile risultato.

NOMI LOCALI E GLOBALI

Un nome che appare in una funzione definita dall'utente può essere o *locale* o *globale*. Un nome globale durante l'esecuzione della funzione ha lo stesso valore che possiede al di fuori della funzione. Un nome locale possiede un valore soltanto mentre una funzione è attiva. Qualunque nome che appare nell'intestazione della funzione (tranne il nome della funzione) è un nome locale. Fin qui si è visto che la intestazione può contenere una variabile risultato e degli argomenti. Poichè questi nomi di variabile sono contenuti nell'intestazione, essi sono locali per la funzione. Anche altri nomi possono essere resi locali nella funzione fondamentale ponendoli nell'intestazione, dopo l'argomento di destra (se esiste) con un punto e virgola. Per esempio, l'intestazione ∇ LOOP R; I; J rende l'argomento di destra R e le variabili I e J locali nella funzione.

ESEMPI:



Probabilmente ci si chiederà perchè sia opportuno introdurre nomi di variabili locali in una funzione. Di seguito sono elencati alcuni motivi per far ciò:

- Per esempio, si definisca una funzione chiamata COUNT che usa una variabile chiamata X. In seguito, si assegna il risultato di un calcolo importante ad una variabile globale chiamata X. Se ora si esegue COUNT accade che:
 1. se X è una variabile locale per COUNT, il valore globale di X non è variato
 2. se X non è locale in COUNT, il valore globale di X (il risultato del calcolo importante) viene variato.
- E' possibile risparmiare area di lavoro in memoria non memorizzando i valori delle variabili che non si usano al di fuori di una funzione.

SALTO, IDENTIFICAZIONE E ITERAZIONE (LOOPING)

Salto e identificazione

Le istruzioni in una funzione definita dall'utente sono normalmente eseguite nell'ordine indicato dal numero progressivo delle istruzioni stesse e la esecuzione finisce alla fine dell'ultima istruzione della sequenza. Tuttavia, questo normale ordine di esecuzione può essere modificato con un *salto* (trasferimento ad un altro punto nella sequenza). Il salto è indicato da una freccia verso destra → seguita da una identificazione che specifica l'istruzione a cui saltare.

Per esempio, l'espressione → START significa salto ad una istruzione identificata con START. Quando si assegna una identificazione ad una istruzione, l'identificazione è seguita da due punti (:) e deve precedere l'istruzione. I due punti separano l'identificazione dalle istruzioni vere e proprie.

```
[2] START:N+N+1
```

```
[5] →START
```

Nell'esempio precedente, l'identificazione START è assegnata alla seconda istruzione della funzione. In questo caso, START ha il valore 2; tuttavia, se la funzione viene listata e l'istruzione non è più la seconda nelle sequenze, a START viene automaticamente assegnato il valore del nuovo numero dell'istruzione. Quando poi la funzione viene fatta eseguire, l'esecuzione dell'istruzione 5 fa effettuare un salto fino all'istruzione identificata con START.

Le identificazioni sono locali per la funzione, cioè possono essere solo all'interno della funzione stessa. Di seguito elencate alcune regole che si applicano esclusivamente all'uso delle identificazioni:

- Esse *non* devono comparire nell'intestazione della funzione
- Non si devono assegnare valori ad esse.

Ci sono due tipi di istruzioni di salto che possono essere usati - i salti incondizionati ed i salti condizionati:

- *I salti incondizionati* sono salti che vengono fatti ogni volta che l'istruzione di salto viene eseguita. Si è già visto un esempio di salto incondizionato, $[5] \rightarrow \text{START}$, in cui il salto all'istruzione identificata con START viene fatto ogni volta che l'istruzione 5 viene eseguita. Un altro uso comune del salto incondizionato è $\rightarrow 0$, che porta a termine l'esecuzione della funzione.
- *I salti condizionati* sono salti che vengono fatti in dipendenza di alcune condizioni che esistono al tempo in cui l'istruzione di salto viene eseguita. I salti condizionati sono usati per esempio, per saltare ad una istruzione se una condizione è vera, altrimenti continuare con l'istruzione seguente (fall through). Questo tipo di salto può essere immesso nella seguente maniera:

$\rightarrow (\text{CONDITION})/N$

Il salto all'istruzione N è fatto se la condizione è vera; altrimenti viene eseguita l'istruzione successiva. Per esempio, l'APL esegue l'istruzione di salto $\rightarrow (I \geq N)/\text{START}$ nella maniera seguente:

1. Per prima cosa, viene esaminata la condizione $(I \geq N)$; il risultato è 1 se la condizione è vera e 0 se la condizione è falsa.
2. Il risultato della prima fase viene poi usato come argomento di sinistra per la funzione di compressione (/):
 - a. se il risultato della fase 1 è 1, START viene scelto dall'argomento di destra e viene eseguito il salto all'istruzione identificata con START.
 - b. Se il risultato della fase 1 è 0, nulla viene scelto dall'argomento di destra (il risultato è un vettore vuoto). Un salto ad un vettore vuoto porta all'esecuzione della successiva istruzione in sequenza (fall through).

Nell'esempio seguente, vengono usati due variazioni di una stessa funzione per determinare la somma di ogni numero da 1 fino al valore dell'argomento (ciascuna funzione fa uso di un metodo di salto differente).

ESEMPI:

```

      VS←SUM2 N
[1] S←0
[2] I←1
[3] CHECK : →(I>N)/0 ← Salto a 0 (porta a termine la funzione) o pas-
[4] S←S+I                saggio all'istruzione successiva.
[5] I←I+1
[6] →CHECKV ← Salto incondizionato a CHECK.
      SUM2 5
15

```

```

      VS←SUM3 N
[1] S←0
[2] I←0
[3] CHECK : S←S+I
[4] I←I+1
[5] →(I≤N)/CHECKV ← Salto a CHECK o passaggio all'istruzione
      SUM3 5                successiva.
15

```

ITERAZIONE (looping)

L'esecuzione di uno stesso segmento di una funzione è chiamato *processo iterativo, (loop)*; quando esso è presente in un programma, bisogna fornire la maniera di uscirne fuori.

ESEMPIO:

↖ LOOP ← Questa funzione esegue un
processo iterativo continuo.

```
[1] I←0
[2] LABEL: 'THIS PROGRAM CONTAINS A LOOP'
[3] I←I+1
[4] →LABEL
```

```
    LOOP
THIS PROGRAM CONTAINS A LOOP
THIS PROGRAM CONTAINS A LOOP
THIS PROGRAM CONTAINS A LOOP
THIS PROGRAM CONTAINS A LOOP
THIS PROGRAM CONTAINS A LOOP
THIS PROGRAM CONTAINS A LOOP
```

Nota: Per finire l'esecuzione
di LOOP, premere il tasto
ATTN.

LOOP[3] ← Nome della funzione ed numero
della istruzione dove l'esecuzione
fermata viene visualizzata.

↖ LOOP[4]→(I≠3)/LABEL ↖ Fornisce una maniera di
uscire fuori dal processo
iterativo.

↖ LOOP[] ↖ Visualizza la funzione.

```
↖ LOOP
[1] I←0
[2] LABEL: 'THIS PROGRAM CONTAINS A LOOP'
[3] I←I+1
[4] →(I≠3)/LABEL
```

```
    LOOP
THIS PROGRAM CONTAINS A LOOP
THIS PROGRAM CONTAINS A LOOP
THIS PROGRAM CONTAINS A LOOP
```

Il processo iterativo viene
eseguito per tre volte.

COME INTRODURRE DATI DURANTE L'ESECUZIONE DI UNA FUNZIONE



Fin qui sono state definite delle funzioni per le quali i dati sono stati forniti come argomenti. Questo metodo di fornire i dati limita a due gli argomenti di immissione, e l'utente deve conoscere la struttura della funzione, così da immettere i dati nell'ordine corretto, tuttavia, è possibile definire funzioni definite dall'utente le quali visualizzano richieste di dati da immettere, nel corso dell'esecuzione. Questo tipo di funzione permette di introdurre una quantità qualsiasi di dati, ed è anche possibile definire la propria funzione così da specificare quale tipo di dati debba essere immesso. Per far ciò, si usa il simbolo  (quad) o  (quad quote) nella funzione per richiedere dati da tastiera. Quando in una funzione viene incontrato il simbolo , l'esecuzione si arresta e viene visualizzato  : per indicare che il sistema aspetta da tastiera dati costituiti da numeri o da caratteri (i dati costituiti da caratteri devono essere racchiusi fra apici). Quando in una funzione viene incontrato il simbolo , l'esecuzione si arresta, appare il cursore, ed il sistema aspetta dati da tastiera; in questo caso, però, qualunque cosa sulla linea di immissione dalla posizione 1 fino al cursore o all'ultimo carattere immesso (qualunque sia la sua posizione sulla linea di immissione) è considerato un dato di caratteri, *anche se* i dati non sono stati racchiusi fra apici.

ESEMPIO:

Si immette la seguente funzione definita dall'utente per determinare il punteggio finale di una partita di pallacanestro.

```

VBASEBALL
[1] 'ENTER THE NAME OF THE VISITING TEAM'
[2] VISIT+
[3] 'ENTER THEIR SCORE BY INNING'
[4] VSCORE+
[5] 'ENTER THE NAME OF THE HOME TEAM'
[6] HOME+
[7] 'ENTER THEIR SCORE BY INNING'
[8] HSCORE+
[9] 'THE FINAL SCORE WAS:'
[10] VISIT
[11] +/VSCORE
[12] HOME
[13] +/HSCOREV
    
```

I dati immessi da tastiera sostituiscono il simbolo  o  e vengono assegnati alle variabili.

I punteggi parziali sono: REDS - 0 1 0 2 0 3 2 5 0
 BLUES - 0 0 0 2 3 1 3 0 0

ESEMPIO (continuazione)

Ora si esegua la funzione:

```
BASEBALL
ENTER THE NAME OF THE VISITING TEAM
REDS
```

```
ENTER THEIR SCORE BY INNING
□:
```

```
0 1 0 2 0 3 2 5 0
```

```
ENTER THE NAME OF THE HOME TEAM
BLUES
```

```
ENTER THEIR SCORE BY INNING
□:
```

```
0 0 0 2 3 1 3 0 0
```

```
THE FINAL SCORE WAS:
```

```
REDS
```

```
13
```

```
BLUES
```

```
9
```

Si noti come i messaggi identificano il tipo di dati da immettere

Questo dato costituito da caratteri non è racchiuso fra apici, poichè è stato richiesto da un ☐ nella funzione

Questo non è un dato costituito da caratteri, poichè è stato richiesto da un ☐ e non è racchiuso fra apici

Nota: A ☐ : indica che i dati da tastiera sono richiesti da ☐ nella funzione ; se non c'è ☐ : (linea di spazi) indica che i dati da tastiera sono richiesti da ☐ nella funzione.

Quando si usano funzioni interattive, può accadere che sia necessario non soddisfare una richiesta di immissione dati. Normalmente, premendo il tasto ATTN si ferma l'esecuzione della funzione; tuttavia premendo il tasto ATTN durante una richiesta di immissione dati non si interrompe la funzione (la funzione continua ad attendere dati da tastiera). Perciò l'APL fornisce un modo per non soddisfare una richiesta di immissione dati. Per non soddisfare una richiesta ☐ si immette → , che interrompe l'esecuzione della funzione.

Per evitare di soddisfare una richiesta ☐ , bisogna introdurre il carattere ☐ . Questo carattere viene generato tenendo premuto il tasto CMD e premendo una volta il tasto ☐ . Questo fa fermare la funzione.

Cosa bisogna fare quando la propria funzione si ferma verrà discusso nel Capitolo 8.

ESEMPIO:

```

BASEBALL
ENTER THE NAME OF THE VISITING TEAM
REDS
ENTER THEIR SCORE BY INNING
□:

```

→

```

BASEBALL
ENTER THE NAME OF THE VISITING TEAM
→

```

```

ENTER THEIR SCORE BY INNING
□:

```

1 1 1 1 1 1 1 1 1

```

ENTER THE NAME OF THE HOME TEAM

```

⏏

INTERRUPT

```

BASEBALL[6] HOME+⏏

```

^

Si usi la funzione BASE BALL per mostrare come evitare di soddisfare una richiesta di dati.

Immettendo → come risposta ad una richiesta □ porta a termine l'esecuzione della funzione.

Si cerchi di evitare un □ immettendo →. Questa immissione viene trattata come un carattere, ed usata come nome della squadra ospite.

Si immettono alcuni numeri cosichè la successiva richiesta □ venga visualizzata.

Immettendo il carattere ⏏ (tenendo premuto CMD e premendo il tasto  una volta)

fa fermare la funzione.

Capitolo 8. Cosa fare quando la propria funzione si interrompe.

L'esecuzione della funzione definita dall'utente si ferma quando:

- si preme il tasto ATTN.
- si usa l'esecuzione controllata stop.
- un errore viene incontrato nella funzione.
- si immette un carattere  (il tasto CMD ed  in seguito ad
- una richiesta  di immissione dati.

Una funzione che si è fermata per una delle precedenti ragioni è chiamata una funzione sospesa. Una funzione sospesa è ancora attiva, perchè la sua esecuzione può ricominciare più avanti.

Vediamo cosa si può fare quando la propria funzione si interrompe.

Quando si preme il tasto ATTN

Quando si preme il tasto ATTN durante l'esecuzione della funzione definita dall'utente, l'esecuzione si interrompe solo alla fine dell'istruzione che sta eseguendo. In questo caso, il 5100 visualizza il nome della funzione ed il nome della successiva istruzione da eseguire.

Dopo che la funzione si è fermata, si può agire in una delle seguenti maniere:

- variare la funzione.
- eseguire di nuovo la funzione.
- eseguire un'altra funzione definita dall'utente.
- eseguire comandi del sistema, tranne)SAVE,)COPY e)PCOPY. I comandi del sistema sono descritti nel "IBM 5100 APL Manuale di Riferimento SA21 - 9213".
- far finire la funzione immettendo →.

Generalmente, dopo aver formato una funzione premendo il tasto ATTN, si vuole più tardi riutilizzarla. Per far ciò, si immette $\rightarrow \square$ LC. $\square$ LC è una variabile del sistema che contiene il numero della successiva istruzione che deve essere eseguita. Consultare l'IBM 5100 APL Manuale di Riferimento SA21 - 9213" per una descrizione completa della variabile del sistema $\square$ LC.

Nota: Se si vuole riprendere l'esecuzione ad una istruzione diversa da quella che segue immediatamente l'ultima istruzione eseguita, si immette $\rightarrow n$ (dove n è numero dell'istruzione da cui si vuol riprendere l'esecuzione).

ESEMPI:

```
VSFUNCTION;COUNT
[1] COUNT←0
[2] LOOP: 'THIS FUNCTION CONTAINS A LOOP'
[3] COUNT←COUNT+1
[4] →LOOP
[5] 'THIS FUNCTION LOOPED'
[6] COUNT
[7] 'TIMES'V
```

Definisce una funzione con un processo iterativo continuo.

```
SFUNCTION
THIS FUNCTION CONTAINS A LOOP
THIS FUNCTION CONTAINS A LOOP
THIS FUNCTION CONTAINS A LOOP
THIS FUNCTION CONTAINS A LOOP
THIS FUNCTION CONTAINS A LOOP
THIS FUNCTION CONTAINS A LOOP
```

Si preme il tasto ATTN per fermare l'esecuzione della funzione.

```
SFUNCTION[3]
```

La funzione viene sospesa all'istruzione il cui numero è visualizzato fra parentesi sullo schermo.

```
VSFUNCTION[4]→(COUNT<3)/LOOPV
```

Si varia la funzione cosicchè essa non contiene più un processo iterativo continuo.

```
→DLC
```

Si riprende l'utilizzo della funzione.

ESEMPI - continua

```
THIS FUNCTION LOOPED  
7  
TIMES
```

Il valore mostrato
qui sullo schermo
è il numero delle
iterazioni.

```
          SFUNCTION  
THIS FUNCTION CONTAINS A LOOP  
THIS FUNCTION CONTAINS A LOOP  
THIS FUNCTION CONTAINS A LOOP  
THIS FUNCTION LOOPED  
3  
TIMES
```

Ora si faccia di
nuovo eseguire la
funzione.

Nota: Quando il tasto ATTN viene premuto due volte durante l'esecuzione di una istruzione o espressione APL (sia dentro sia fuori di una funzione definita dall'utente), l'esecuzione dell'istruzione o espressione si arresta immediatamente. Viene visualizzato il messaggio INTERRUPT, l'istruzione in esecuzione ed il carattere (^) che indica dove l'istruzione si è arrestata. Si può usare questo metodo per interrompere istruzioni la cui esecuzione dura a lungo. Tuttavia, qualunque risultato generato dall'istruzione o espressione prima dell'interruzione può non esistere più dopo l'interruzione stessa.

Quando si usa la funzione di stop

La funzione di Stop è già stata trattata nel Capitolo 7. Usando la funzione di Stop (come pure usando il tasto ATTN), si può agire in una delle seguenti maniere:

- variare la funzione
- eseguire di nuovo la funzione
- eseguire un'altra funzione definita dall'utente
- eseguire comandi del sistema, tranne)SAVE,)COPY e)PCOPY.
- riprendere l'esecuzione della funzione immettendo → □ LC.
- finire la funzione immettendo →

QUANDO VIENE TROVATO UN ERRORE NELLA FUNZIONE

Una funzione definita dall'utente si può fermare anche in casi diversi da quelli visti fino ad ora (ATTN e Stop). Cioè, il 5100 ferma automaticamente l'esecuzione della funzione e visualizza un messaggio di errore quando viene trovato un errore nella funzione. I messaggi d'errore e le risposte suggerite all'utente per ogni errore sono descritti nel "IBM 5100 APL Manuale di Riferimento SA21 - 9213".

Errori nelle funzioni definite sono talvolta difficili da trovare e da correggere. Il messaggio visualizzato indica dove l'esecuzione dell'istruzione è stato fermato, e perchè; ma le ragioni per cui il 5100 ha sospeso l'esecuzione a quel punto possono dipendere da un errore (sia un errore di immissione, sia un errore nella soluzione del problema) fatto in un punto precedente della funzione, nella stessa istruzione o anche in un'istruzione precedente. Qui di seguito ci sono alcuni consigli per aiutare a trovare errori in una istruzione o espressione in cui è stata sospesa l'esecuzione o che comunque fornisce risultati erranei.

- Controllare ogni espressione (istruzione) immessa cercando eventuali errori d'immissione.
- Analizzare l'esecuzione dell'espressione da destra a sinistra. Si ricordi che l'APL esegue una espressione da destra a sinistra e risolve le espressioni fra parentesi (da destra a sinistra) quando le incontra.
- Si usi la funzione ρ per controllare che le dimensioni degli argomenti sono quelle che ci si aspetta che siano. Per esempio, si supponga che si abbia una funzione di nome CAT che unisce due vettori insieme in modo da formare un unico vettore; invece, uno dei due argomenti immessi è una matrice.
- Battere i nomi delle variabili per controllare i valori degli argomenti (i nomi locali in una funzione sospesa possono essere visualizzati, poichè la funzione è ancora attiva).
- Decomporre l'espressione ed eseguirla apiccoli segmenti. Le frecce verso l'alto  e verso il basso  (i tasti di avanzamento ed arretramento di riga) renderanno più facile la decomposizione dell'espressione: cioè, si può eseguire l'espressione proprio come la esegue l'APL (da destra a sinistra, con le parentesi risolte quando vengono incontrate). Per far ciò, si immetta la prima operazione eseguita dall'APL il cui risultato viene visualizzato. Poi si preme il tasto di arretramento  tre volte ed il tasto di avanzamento

una volta per eliminare il risultato precedente dello schermo (cosicchè non sia più sulla linea di immissione quando il tasto EXECUTE viene nuovamente premuto) e per porre le istruzioni già immesse in una posizione opportuna per aggiungere altre operazioni. Ora si può aggiungere l'operazione successiva all'istruzione, e quella successiva, finchè si trova l'errore.

E' importante mantenere un elenco (o uno stampato con l'uso della stampatrice IBM 5103 o una copia fatta a mano) di quello che è stato fatto quando si cerca la causa di un errore. Infatti se non lo si riesce a trovare e si pensa che il problema sia causato dal 5100, questo elenco aiuterà il tecnico di manutenzione a determinare l'eventuale guasto.

Quando una funzione si è arrestata poichè si è verificato un errore, come quando si preme il tasto ATTN o la funzione di arresto controllato, è possibile agire in una delle seguenti maniere:

- visualizzare o listare la funzione.
- eseguire di nuovo la funzione.
- eseguire un'altra funzione definita dall'utente.
- eseguire i comandi del sistema, tranne)SAVE,)COPY e)PCOPY.
- riprendere l'esecuzione della funzione immettendo → □ LC.
- finire la funzione immettendo → .

QUANDO SI IMMETTE UN CARATTERE ① PER UNA RICHIESTA DI DATI □

Nel capitolo 7 si è usato il carattere ① per evitare una immissione dati in seguito ad una richiesta □ e arrestare l'esecuzione di una funzione. In questo caso, il 5100 visualizza il messaggio INTERRUPT, il nome della funzione e l'istruzione che ha richiesto l'immissione di dati. Dopo che la funzione si ferma, è possibile eseguire le stesse operazioni che si fanno quando la funzione si ferma per qualche altra ragione. Tuttavia, molto spesso è necessario portare a termine l'esecuzione della funzione immettendo →.

DETERMINAZIONE DI QUALI FUNZIONI SONO SOSPESE

L'indicatore di stato contiene il nome della funzione ed il numero dell'istruzione da eseguirsi subito dopo quella in cui è avvenuto l'arresto, per ogni funzione sospesa. Per visualizzare l'indicatore di stato, si immette)SI o)SIV. Si vede a "IBM 5100 APL Manuale di Riferimento SA21 - 9213" per maggiori dettagli sull'indicatore di stato.

USO DEL TASTO DI MANTENIMENTO HOLD PER ARRESTARE TEMPORANEAMENTE L'ESECUZIONE

HOLD

Sono già stati trattati i modi con cui una funzione dell'utente può essere sospesa. Si può interrompere di una funzione l'esecuzione anche premendo una volta il tasto HOLD (mantenimento). In effetti, ciò ferma l'intero sistema, che non elabora più nessun dato. Per riprendere l'elaborazione dopo che si è premuto il tasto HOLD, bisogna premere il tasto HOLD di nuovo. Questo tasto è utile quando l'informazione sullo schermo cambia rapidamente; cioè, è possibile fermare l'elaborazione, leggere l'informazione visualizzata e poi riprendere l'elaborazione.

ESEMPI:

```

      VHOLDIF ← Definisce una funzione
C1] H←0
C2] 'PRESS THE HOLD KEY TO STOP PROCESSING
C3] LOOP:H←H+1
C4] H
C5] →(H*25)/LOOPV

```

```

      HOLDIF
PRESS THE HOLD KEY TO STOP PROCESSING
1
2 }
3 } ← Il valore visualizzato qui sullo schermo in-
: } dica quante volte la funzione ha eseguito il
: } processo iterativo prima che l'elaborazione
: } sia fermata.

```

Ora si preme il tasto HOLD di nuovo per riprendere l'elaborazione.

Nota: Se il 5100 non elabora nessun dato o nessuna funzione definita dall'utente, e il cursore non è intermittente, il tasto HOLD potrebbe essere stato premuto, fermando così ogni elaborazione.

Fin qui è stata usata solamente l'area di memoria attiva del 5100. Essa è quella parte della memoria del 5100 in cui vengono eseguite le operazioni; è anche il posto dove vengono memorizzate le variabili e le funzioni definite dall'utente. Quando si pone l'interruttore 5100 ACCESO/SPENTO nella posizione SPENTO o si preme RIPARTENZA, i dati nell'area di memoria attiva vengono persi. Tuttavia, prima di spegnere il 5100 o di premere RIPARTENZA, è possibile salvare i dati dall'area di memoria copiando il contenuto dell'area stessa sulla cassetta nastro. Essa compie le funzioni di una libreria: cioè, si può trascrivere il contenuto dell'area attiva di memoria sul nastro (come quando si pone un libro in uno scaffale) e più tardi, portare le informazioni memorizzate su nastro nell'area di memoria attiva (come quando si prende un libro da uno scaffale per consultarlo).

La libreria su nastro consiste di uno o più files (ogni file è come un libro) e come ogni libro di una libreria ha un nome, così ogni file che contiene informazioni sul nastro può avere un nome (identificazione del file).

I comandi del sistema IBM 5100 sono il mezzo con cui si controlla l'area di memoria ed il nastro (libreria). Si dia uno sguardo alle scritte sopra la tastiera alfanumerica: questa parola chiave di comando del sistema possono essere immesse semplicemente premendo il tasto CMD insieme al tasto opportuno corrispondente alla scritta voluta. Ogni parola chiave di comando del sistema può essere immessa anche carattere per carattere. Si noti che ogni comando del sistema comincia con un simbolo). Ci sono alcuni comandi che tuttavia non appaiono fra le scritte sopra la tastiera alfanumerica. Tutti i comandi del sistema 5100 sono descritti in maggior dettaglio nell' "IBM 5100 APL Manuale di Riferimento, SA21 - 9213".

Nell'esempio seguente si vedrà come operano alcuni comandi del sistema. Per prima cosa, bisogna inserire una cassetta a nastro nel 5100. Ci si assicuri che il nastro non contenga dati utili per una elaborazione successiva e che l'indicatore SAFE (Figura 2) non punti verso SAFE. Ora si inserisca la cassetta

Assicurarsi che l'intermittente
SAFE sia in questa posizione

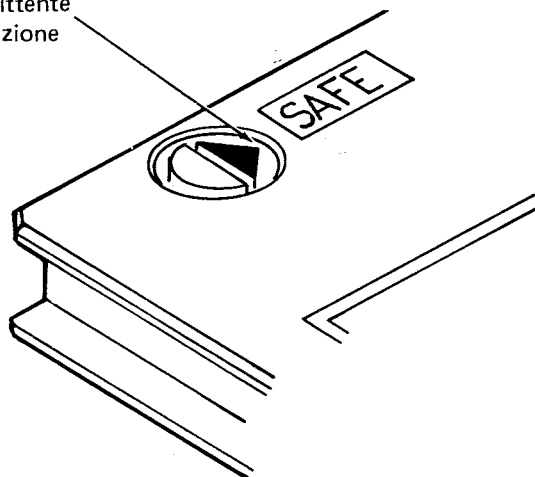


Figura 2. L'interruttore SAFE

Inserire la cassetta a nastri nel 5100
in questa maniera

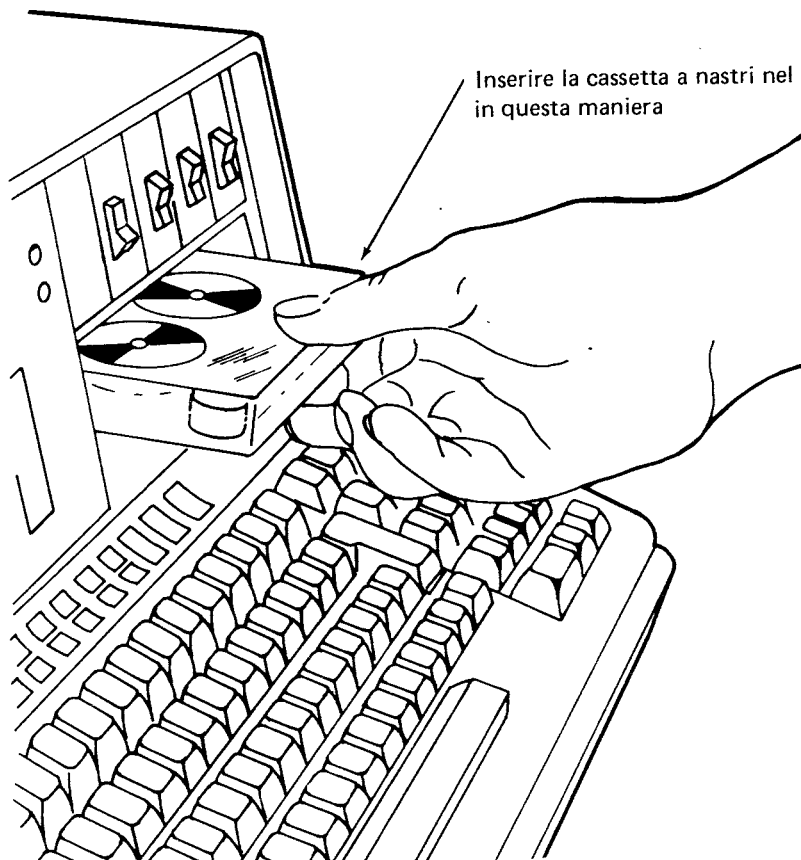


Figura 3. Inserzione di una cassetta corretta a nastri nel 5100

ESEMPI:

Premere RESTART sul 5100: tutti i dati nell'area di memoria attiva sono ora persi.

CLEAR WS

← Questo messaggio viene visualizzato quando il 5100 è di nuovo pronto per immettere dati.

Si immettono le seguenti funzioni e variabili per memorizzarle su disco:

```

      VEXAMPLE;R:NAME
[1] 'THIS FUNCTION COUNTS THE CHARACTERS IN YOUR NAME'
[2] 'NOW ENTER YOUR NAME'
[3] NAME←0
[4] 'THERE ARE'
[5] NAME
[6] 'CHARACTERS IN YOUR NAME'V

```

VARIABLE←'LET'S SAVE THIS DATA'

Si provi ora la funzione EXAMPLE per vedere se funziona.

)FNS
EXAMPLE

← Il comando del sistema)FNS visualizza nomi di funzioni definite dall'utente nell'area di memoria attiva.

)VARS
VARIABLE

← Il comando del sistema)VARS visualizza i nomi globali di variabile nell'area di memoria attiva.

Prima di poter usare un nastro, bisogna dare il formato dei files che si vogliono usare.

Il comando)MARK definisce il formato dei files sul nastro. Questo comando specifica:

- la grandezza del file di cui si dà il formato
- il numero dei files
- il numero di partenza dei files

```
)MARK 16 3 1  
MARKED 0003 0016
```

APL risponde con MARKED, il numero dell'ultimo file che è formato e le dimensioni dei files. Se il file che si vuole usare è già stato assegnato, si otterrà il messaggio ALREADY MARKED. In questo caso, si immetta GO e si prema il tasto EXECUTE per ridare il formato dei files su nastro.

I files vengono posti in blocchi di 1024 bytes. Per esempio, la grandezza dei files di cui è stato dato ora il formato è di sedici blocchi di 1024 bytes (cioè 16384 bytes). Si veda "IBM 5100 APL Manuale di Riferimento, SA21 - 9213" ulteriori informazioni sul formato dei files.

Ora si scrive il contenuto della memoria su nastro.

```
)CONTINUE 1001 INFO  
CONTINUED 1001 INFO
```

Questo diviene il nome del file su nastro.

Questo specifica i numeri dell'unità a nastri e del file (unità 1 file 001) in cui viene scritto il contenuto della parte attiva della memoria.

```
)CLEAR  
CLEAR WS
```

Non è necessario spegnere il 5100 o premere RESTART per cancellare i dati in memoria si può usare questo comando del sistema.

I dati posti su nastro possono essere portati nuovamente nella memoria attiva.

```
)LOAD 1001 INFO  
LOADED 1001 INFO
```

Il nome del file su nastro.

Numeri dell'unità a nastri del file.

)FNS
 EXAMPLE
)VARS
 VARIABLE

Ora i dati memorizzati su nastro sono nuovamente nella memoria attiva.

Gli altri comandi del sistema sono descritti nel "IBM 5100 APL Manuale di Riferimento, SA21 - 9213". Si consiglia di usarli, per vedere come funzionano.

Fin qui, si è appreso come copiare l'intero contenuto della memoria attiva su nastro. Tuttavia si può anche copiare una variabile per volta su di un file su nastro. Questi nastri possono essere in seguito letti nello stesso ordine con cui sono stati scritti. Per maggiori dettagli, si veda il Capitolo 8, "Immissione ed emissione di dati su nastro e Stampante", "IBM 5100 APL Manuale di Riferimento, SA21 - 9312".

CHE COSA SONO LE VARIABILI DEL SISTEMA?

Le variabili del sistema sono variabili all'interno della memoria attiva che controllano il sistema. Tutte cominciano con il simbolo $\square$ e sono impostate ad un valore iniziale del 5100 durante la fase dei controlli iniziali. Si confronti "IBM 5100 APL Manuale di Riferimento SA21 - 9213" per una descrizione completa di ogni variabile del sistema. Nell'esempio seguente, si può vedere come il valore di alcune variabili del sistema può essere cambiato, e quale sia l'effetto di questa operazione su alcune funzioni APL.

ESEMPLI:

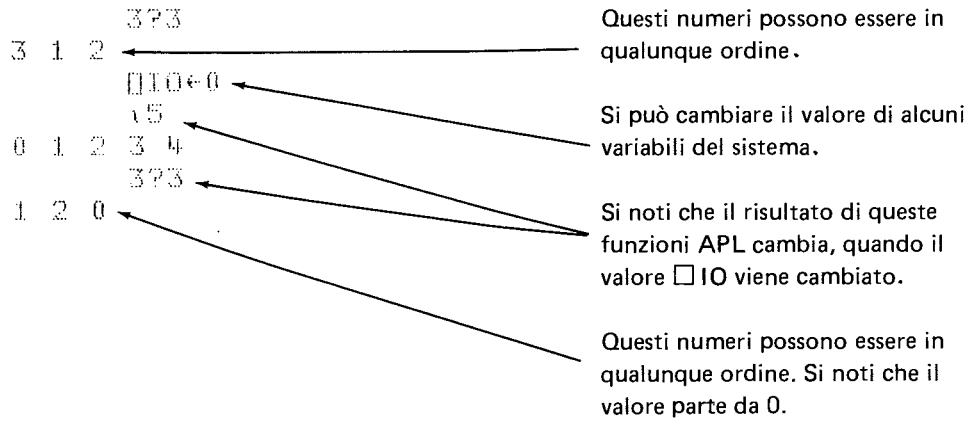
La variabile del sistema *origine di indice (index origin)* $\square$ IO determina il valore iniziale di un indice. Il suo valore può essere 0 o 1; cioè significa che l'indice del primo elemento di un vettore o di una matrice è uno 0 o un 1, a seconda del valore di $\square$ IO. Questa variabile del sistema influisce sulle funzioni APL: $\uparrow$? Δ Ψ

$\square$ IO
 1
 15
 1 2 3 4 5

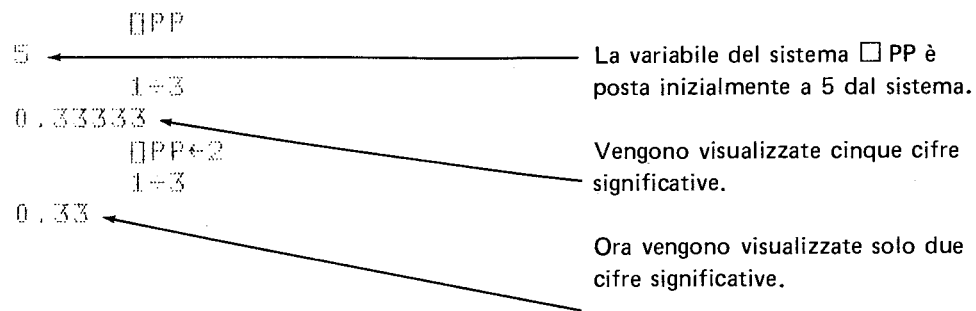
Si può visualizzare il valore di una variabile del sistema nella stessa maniera con cui si visualizza qualunque altra variabile.

La variabile del sistema $\square$ IO è posta inizialmente al valore 1 del sistema.

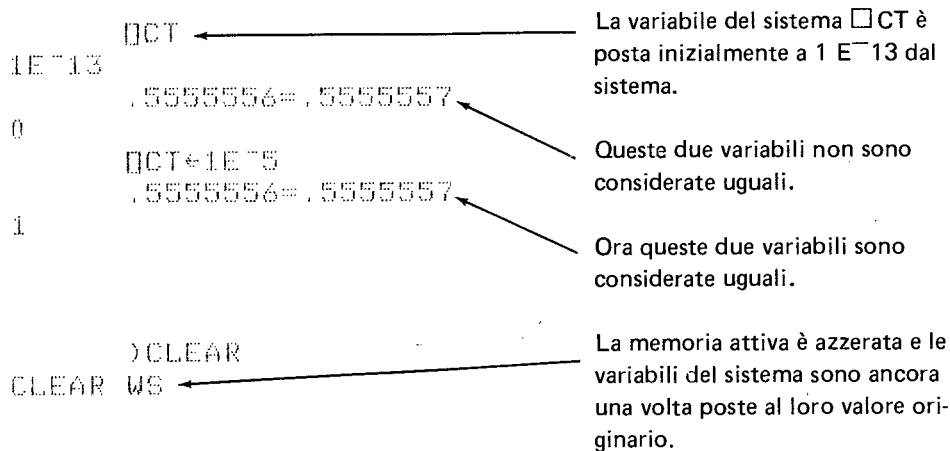
Questo è il risultato quando $\square$ IO è posta ad 1.



La variabile del sistema $\square$ PP, precisione di emissione (printing precision), determina il numero delle cifre visualizzate.



La variabile del sistema $\square$ CT, tolleranza di confronto (comparision tolerance), determina il modo con cui trattare gli argomenti delle funzioni relazionali, L e T.



Gradiremmo i Vostri commenti . . .

Questa pubblicazione fa parte di una biblioteca che costituisce la fonte di riferimento per analisti, programmatori ed operatori di sistemi IBM.

I vostri commenti (potete scriverli sul retro di questo foglio) saranno attentamente considerati dalle persone responsabili della preparazione e della pubblicazione di questo materiale. Tutti i commenti ed i suggerimenti diventeranno proprietà della IBM.

piegare

piegare

IBM Italia
Documentazione Tecnica Applicativa DIB
Via Pirelli, 18
20124 MILANO

Francatura a carico
del destinatario da
addebitarsi sul conto
di credito n. 2548
presso l'Ufficio Po-
stale di Milano A.D.
(Aut. Dir. Prov. P.T.
di Milano n. 126807/
LA III del 21-9-1963)

piegare

piegare

