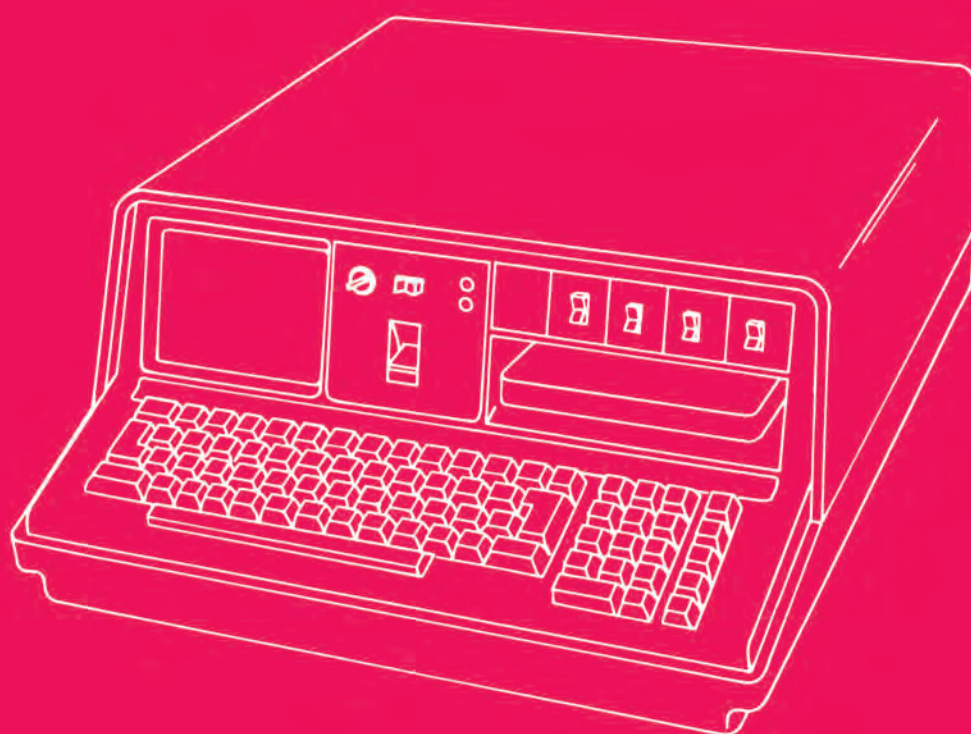




IBM 5100 BASIC Manuale di riferimento

5100



IBM 5100
Elaboratore Portatile
BASIC Manuale di riferimento

La presente pubblicazione è stata tradotta dalla corrispondente originale
in lingua inglese "IBM 5100 BASIC Reference Manual" (from n. SA21-9217-1)
reperibile presso la IBM Italia S.p.A.
Per eventuali aggiornamenti riferirsi alla fonte originale.

© International Business Machine Corporation, 1975
© IBM Italia S.p.A. - luglio 1976

SA13 - 0086 - 00

Indice Generale

CAPITOLO 1. FUNZIONAMENTO	1
Visione d'insieme dell'IBM 5100	1
Schermo video	2
Tastiera	3
Tasti speciali del 5100	4
Interruttori del 5100	5
Spie luminose del 5100	6
Correzione e stampa delle righe immesse	6
Capacità della memoria del 5100	7

CAPITOLO 2. COMANDI DEL SISTEMA	9
Il comando AUTO	11
Il comando GO	12
Il comando LIST	14
Il comando LOAD	16
File di dati creati tramite tastiera	19
Funzioni-tasto	19
Il comando MARK	22
Il comando MERGE	24
Il comando PATCH	26
Il comando RD = '	30
Il comando RENUM	31
Il comando REWIND	33
Il comando RUN	34
Il comando SAVE	36
Il comando UTIL	38
Tipi di file	39

CAPITOLO 3. DATI COSTANTI E VARIABILI ...	41
Assieme di caratteri BASIC	41
Caratteri Alfabetici	41
Caratteri Numerici	41
Caratteri Speciali	42
L'uso degli Spazi vuoti	42
Tratto di sottolineatura	42
Dati aritmetici	42
Formato dei dati aritmetici	43
Costante aritmetiche	45
Costanti intere	45
Variabili aritmetiche	45
Dati carattere	46
Costanti carattere	46
Variabili carattere	46
Insiemi	46
Dichiarazione degli Insiemi	48
Ridimensionamento degli Insiemi	49
Insiemi aritmetici	49

Insiemi di carattere	50
Sommario di Nominativi Convenzionali	50
Funzioni del Sistema	50
Espressioni	52
Espressioni aritmetiche ed operatori	52
Gerarchia aritmetica	54
Espressioni di carattere	55
Espressioni logiche	56
Espressioni di insiemi	57
Funzione sottostringa	57

CAPITOLO 4. ISTRUZIONI BASIC	59
Righe di istruzione	59
Istruzioni BASIC	60
CHAIN	62
CLOSE	63
DATA	64
DEF, FNEND, RETURN	66
Funzione di riga singola	66
Funzione multiriga	66
DIM	69
END	71
FNEND	72
FOR E NEXT	73
GET	75
GOSUB E RETURN	77
GOTO	79
IF	80
IMAGE	82
INPUT	83
LET	85
NEXT	87
OPEN	88
Numeri di file negativi	89
PAUSE	90
PRINTE	91
Zone di stampa	92
Spaziatura dei Valori Visualizzati o Stampati	92
Visualizzazione dell'operazione di riga	94
Memoria di transito per le righe di stampa	98
PRINT USING e maschera di stampa	100
Conversione dei Valori di riferimento dei Dati	101
Specifiche di formato	102
PUT	106
READ	108
REM	110
RESET	111

RESTORE	112
RETURN	113
STOP	114
USE	115
Operazioni di matrici	116
Istruzioni di assegnazione MAT	116
Assegnazione MAT (Valore scalare)	117
Assegnazione MAT (Semplice)	119
Assegnazione MAT (Addizione e sottrazione)	121
Assegnazione MAT (Moltiplicazione di matrici) ...	123
Assegnazione MAT (Moltiplicazione scalare)	125
Assegnazione MAT (Funzione d'identità)	127
Assegnazione MAT (Funzione inversa)	129
Assegnazione MAT (Funzione transposta)	131
MAT GET	133
MAT INPUT	135
MAT PRINT	137
Spaziatura dei Valori Visualizzati	137
Operazioni di Visualizzazione/Stampa di righe	138
MAT PRINT USING	140
MAT PUT	143
MAT READ	144

CAPITOLO 5. ULTERIORI INFORMAZIONI

SUL 5100	147
Compatibilità del BASIC 5100 con il BASIC dell'IBM 370/VS	147
Uso e precauzioni con le bobine a nastro	148
Segretezza dei dati	148

APPENDICE A. CARATTERI BASIC	149
---	------------

APPENDICE B. MESSAGGI DI ERRORE BASIC E CORREZIONI SUGGERITE DALL'OPERATORE	151
--	------------

APPENDICE C. PIANIFICAZIONE E PROCEDURE DI INSTALLAZIONE	160
Piano della installazione	160
Procedure di installazione	163
Procedure di installazione del 5100	163
Procedure di controllo BASIC	164

APPENDICE D. STAMPATRICE 5103	174
Come inserire i moduli	175
Come regolare il commutatore di comando delle copie in funzione dello spessore dei moduli	177

VISIONE D'INSIEME DELL'IBM 5100

L'IBM 5100 (Figura 1) è un elaboratore da tavolo progettato per un uso diretto da parte dell'utente nella risoluzione dei problemi. Il 5100 possiede uno schermo video, una doppia tastiera alfanumerica e numerica, un'unità nastro, interruttori e spie luminose. Lo schermo video e le spie luminose forniscono informazioni all'utente mentre la tastiera e gli interruttori permettono all'utente di comandare le operazioni che il sistema esegue. La Figura 1 mostra un 5100 con il doppio linguaggio BASIC/APL.

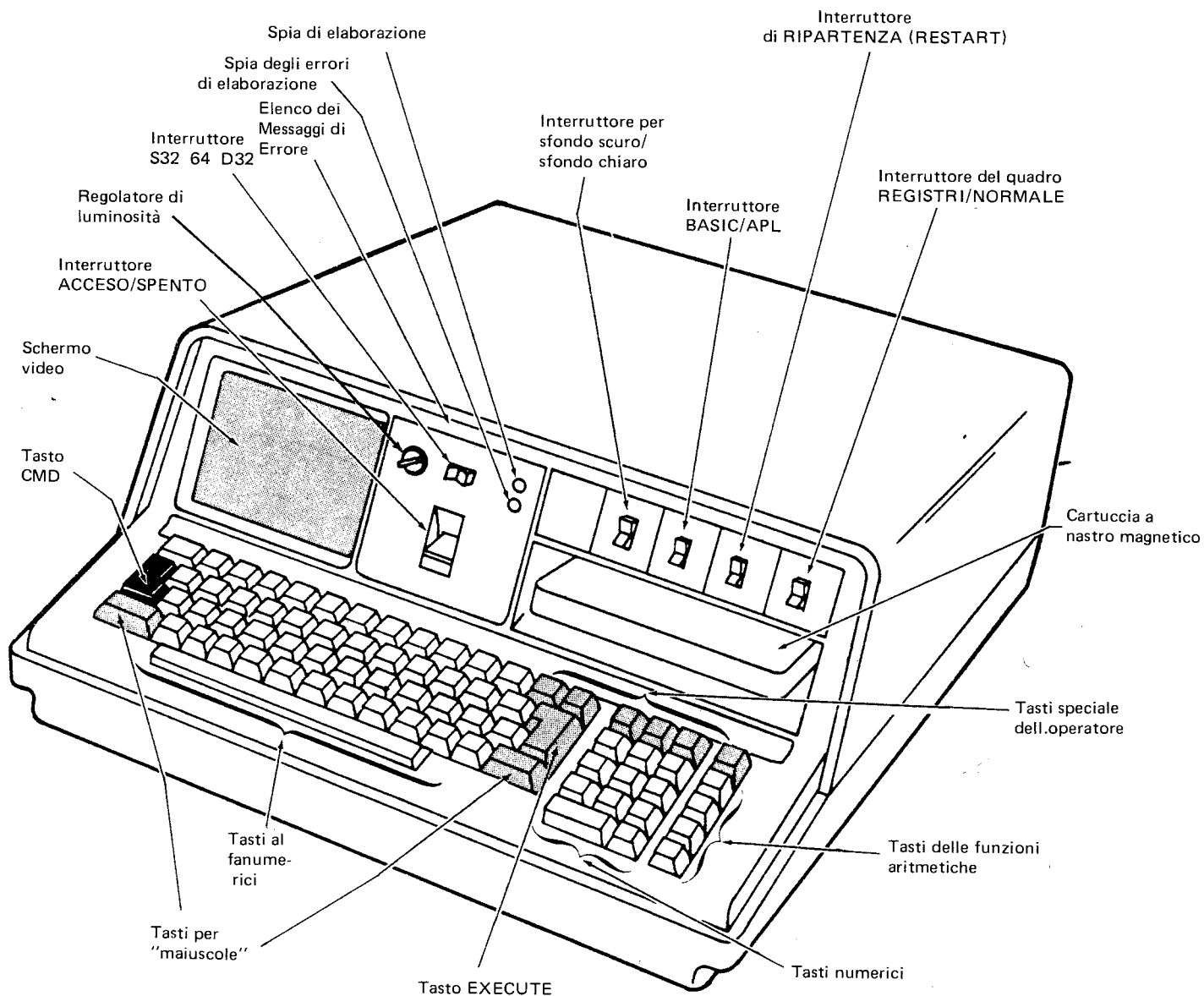


Figura 1. L'elaboratore IBM 5100

SCHERMO VIDEO

Lo schermo video (Figura 2) può visualizzare 16 righe di dati per volta, ciascuna di 64 caratteri. Vengono visualizzati sia i dati di immissione (l'informazione fornita dall'utente) sia i dati di emissione (le informazioni elaborate). L'ultima riga in basso contiene l'informazione sullo stato di funzionamento del sistema. Il numero che compare nell'angolo a sinistra in basso (NNNNN) indica il numero delle posizioni di caratteri (bytes) presenti nella memoria e disponibili per l'utente. La riga 1 (riga di immissione) contiene l'informazione immessa tramite tastiera. Il cursore (tratto orizzontale lampeggiante) indica dove verrà visualizzata la prossima informazione immessa tramite tastiera. Se il cursore si sposta in un punto che è già occupato da un carattere, tutto il carattere si metterà a lampeggiare. Allorchè il BASIC elabora una nuova immissione, tutte le righe dello schermo (fatta eccezione per l'ultima riga in basso), vengono spostate verso l'alto, in modo che una ulteriore informazione possa essere immessa ancora sulla riga 1.

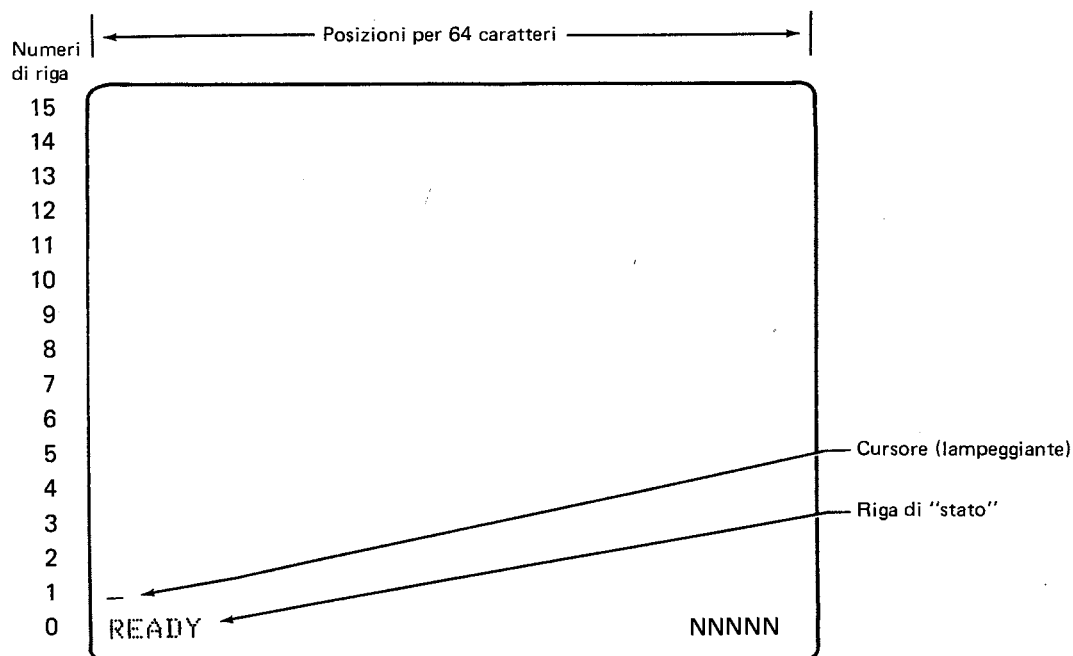


Figura 2. Schermo video del 5100

TASTIERA

La console del 5100 (Figura 3) è fornita di una doppia tastiera alfanumerica e numerica. I dati numerici possono essere immessi convenientemente mediante l'uso del dispositivo dei tasti numerici dell'elaboratore, posti a destra dei tasti alfanumerici. Gli stessi dati possono essere immessi mediante l'uso dei tasti numerici posti al disopra dei tasti alfanumerici. I tasti delle operazioni aritmetiche, posti a destra della tastiera si trovano anche fra i tasti alfanumerici.

E' da tener presente che in una tastiera solo BASIC (Figura 4), i simboli APL, che si trovano nella parte superiore dei tasti alfanumerici e che possono essere immessi come si fa per un carattere maiuscolo in una normale macchina da scrivere, possono essere immessi come dati, benchè essi non compaiano esplicitamente sulla tastiera BASIC.

Quando qualcuno di questi tasti viene battuto, i caratteri compaiono sulla linea di immissione dello schermo video.

Le parole chiave dei comandi BASIC sono stampate al disopra dei tasti alfanumerici. Queste parole vengono immesse iniziando dalla posizione che occupa il cursore allorchè viene tenuto pigiato il tasto CMD e viene battuto il relativo tasto voluto.

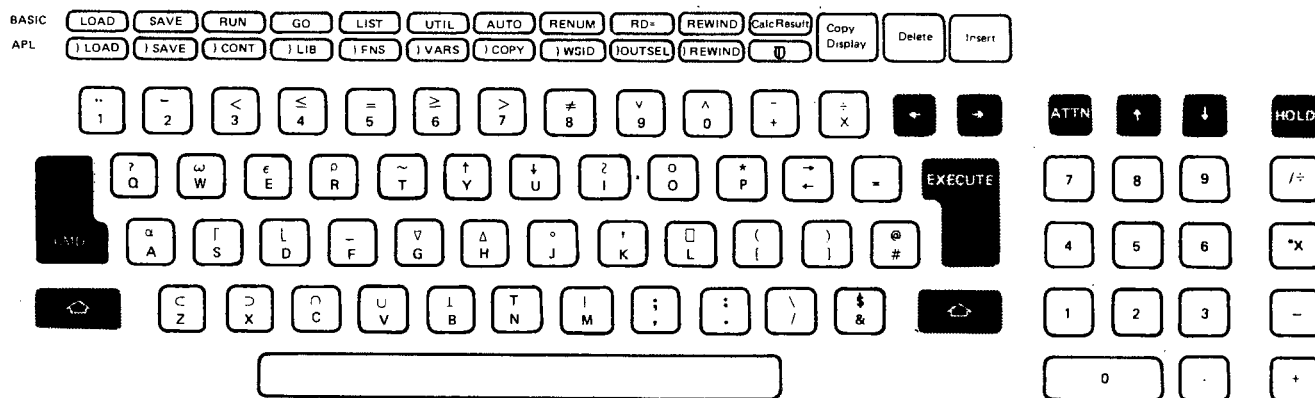


Figura 3. Tastiera BASIC/APL

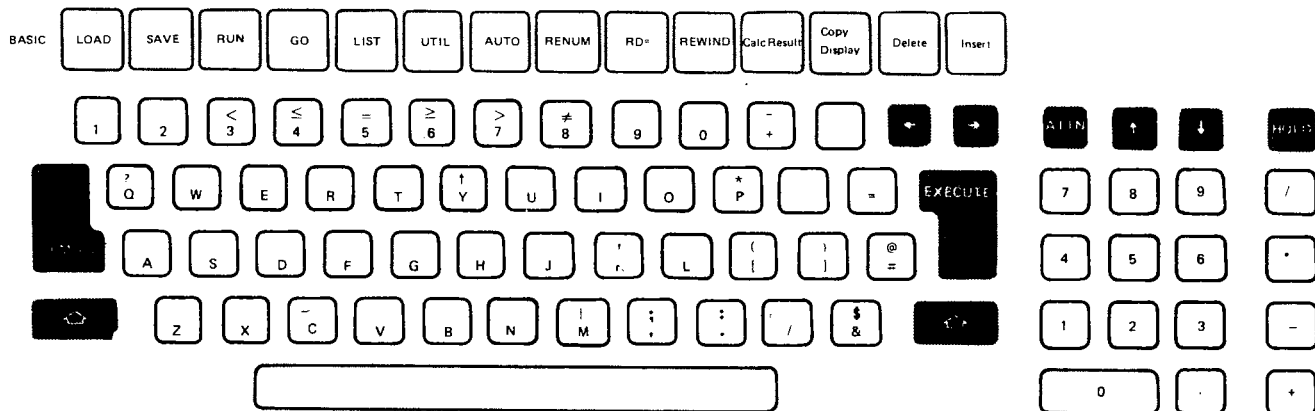


Figura 4. Tastiera solo BASIC

Tasti Speciali del 5100

I seguenti tasti possiedono delle funzioni speciali relativamente alle operazioni del 5100.



(Attenzione). Si può battere il tasto ATTN per interrompere le operazioni del sistema. Mentre si stanno immettendo dati da tastiera, si può battere ATTN per far scomparire tutto ciò che esiste, a partire dalla posizione del cursore incluso, fino alla fine della riga, dalla riga di immissione dello schermo video. Si può continuare l'immissione di informazioni. Inoltre, ATTN e HOLD sono gli unici tasti i quali sono operativi allorché lo schermo video si mette a lampeggiare a causa dell'individuazione di un errore. Dopo aver battuto ATTN per interrompere il lampeggiamento dello schermo video, tutti i tasti ritornano operativi e si può continuare.



(Comando). Quando si tiene piegato il tasto CMD, si attiva la riga superiore dei tasti alfanumerici, che determina l'immissione della parola chiave di comando sulla riga di immissione allorché viene contemporaneamente battuto il numero che si trova sotto l'etichetta. L'elaborazione del comando viene eseguita quando si batterà EXECUTE. Il tasto CMD si usa inoltre assieme ai tasti numerici (posti all'estrema destra della tastiera) per inizializzare funzioni-tasto. Evitare di tener premuto il tasto CMD e contemporaneamente battere HOLD; questa operazione attiva una funzione riservata al personale di manutenzione. Tener premuto il tasto CMD e contemporaneamente un tasto con una parola chiave di comandi BASIC posta davanti al tasto stesso, si immetterà il comando o l'istruzione a partire dalla posizione che in quel momento occupa il cursore.



Allorché si batte EXECUTE, il sistema elabora la riga di dati che è stata appena introdotta sulla riga di immissione. Inoltre, si può battere EXECUTE per ricominciare l'elaborazione interrotta. Battendo il tasto EXECUTE quando la riga di immissione è vuota si produce la stessa azione del comando GO (vedi Comando GO, Capitolo 2).



Allorché si batte HOLD, ogni elaborazione si arresta a meno che lo schermo video sia vuoto. L'elaborazione riprende quando si batte nuovamente HOLD. Pertanto il tasto HOLD permette di leggere i dati visualizzati sullo schermo durante una operazione di emissione. Mentre l'elaborazione si trova interrotta, il tasto CMD ed i tasti degli operatori aritmetici posti a destra della tastiera, sono riservati all'uso del personale di manutenzione.



(Caratteri "Maiuscoli"). Tenendo premuto il tasto dei caratteri "maiuscoli", si può scegliere come immissione un simbolo posto superiormente nel tasto premuto.



(Copia del contenuto dello schermo video — quando il tasto CMD è tenuto premuto). Allorché si tiene premuto il tasto CMD e si batte il tasto posto sotto l'etichetta Copia Schermo, tutti i dati che si trovano visualizzati sullo schermo video vengono stampati dalla stampatrice collegata. E' possibile

usare tale funzione quando il 5100 si trova in uno stato di mantenimento (HOLD) oppure ogni qualvolta il 5100 è in attesa di una immissione da tastiera. Sugli elaboratori con il doppio linguaggio BASIC/APL il tasto  viene usato per

attivare la funzione Copia Schermo. L'interruttore S32 64 D32 non ha effetto sui dati stampati.

I seguenti tasti possiedono la possibilità di ripetere l'operazione; ciò significa che continueranno a funzionare per tutto il tempo durante il quale verranno tenuti pigiati.



(Spazio indietro). Quando si batte questo tasto il cursore si sposta di una posizione a sinistra. Quando si fa spazio indietro con il cursore nella posizione 1 della riga di immissione, il cursore si sposta nella posizione alla estrema destra della riga stessa. Quando si tiene pigiato il tasto CMD e si batte questo tasto, viene cancellato immediatamente il carattere che si trova nella posizione occupata dal cursore. Tutti i caratteri che si trovano alla destra del cursore vengono spostati di una posizione a sinistra ogni volta che si batte tale tasto. Il cursore, in conclusione, non si sposta.



(Spazio avanti). Quando si batte questo tasto il cursore si sposta di una posizione a destra. Quando si fa spazio avanti oltre la fine della riga il cursore ritorna alla prima posizione della riga. Quando si tiene premuto il tasto CMD e si batte questo tasto, si inserisce immediatamente uno spazio vuoto nella posizione occupata dal cursore in quel momento e tutti i dati, a partire dalla posizione occupata dal cursore vengono spostati a destra di una posizione. Dato che il cursore dopo tale operazione non si è spostato, è pertanto possibile immettere al posto dello spazio vuoto un altro carattere. Se la riga contiene un carattere posto nella sua ultima posizione (la 64ma), la funzione di inserzione non viene eseguita.



(Avanzamento di riga). Quando si batte questo tasto, ciascuna riga visualizzata si sposta in alto di una riga (fatta eccezione della riga dello stato).



(Arretramento di riga). Quando si batte questo tasto, ciascuna riga visualizzata si sposta in basso di una riga fatta eccezione della riga dello stato).

INTERRUTTORI DEL 5100

Gli interruttori seguenti si trovano nella console del 5100, al disopra della tastiera:

ACCESO/SPENTO. Questo interruttore serve per accendere oppure spegnere l'elaboratore 5100. Quando si pone questo interruttore nella posizione di acceso, il 5100 diviene operativo in 10-15 secondi approssimativamente.

RIPARTENZA. E' un interruttore a pressione con ritorno a molla, che fa reinizializzare il 5100 allo stato di accensione. Negli elaboratori 5100 provvisti dei due linguaggi, la posizione dell'interruttore BASIC/APL determina quale dei due linguaggi venga reinizializzato.

BASIC/APL. Questo interruttore compare unicamente sugli elaboratori provvisti dei due linguaggi, e determina quale dei due linguaggi verrà inizializzato all'atto dell'accensione o della ripartenza.

SFONDO SCURO/SFONDO CHAROI. Questo interruttore a due posizioni imposta lo schermo video ad avere caratteri chiari su sfondo scuro oppure caratteri scuri su sfondo chiaro. Dopo aver azionato tale interruttore è opportuno regolare l'interruttore della LUMINOSITA'.

LUMINOSITA'. Questo regolatore serve a variare la intensità luminosa dei caratteri ovvero dello sfondo dello schermo.

REGISTRI/NORMALE. Questo interruttore è riservato all'uso del personale di manutenzione. Durante l'elaborazione del 5100 questo interruttore deve rimanere nella posizione NORMALE.

S32 64 D32. Le tre posizioni di questo interruttore sono:

S32 I primi 32 caratteri di sinistra dello schermo vengono visualizzati interponendo tra un carattere e l'altro uno spazio vuoto.

64 Vengono visualizzati fino a 64 caratteri per ciascuna riga, in posizioni contigue.

D32 Gli ultimi 32 caratteri di destra dello schermo video vengono visualizzati interponendo tra un carattere e l'altro uno spazio vuoto.

SPIE LUMINOSE DEL 5100

La console del 5100 possiede due spie luminose:

ERRORE DI ELABORAZIONE. Quando questa spia si accende vuol dire che è stato individuato un errato funzionamento del sistema e pertanto non sono più normalmente possibili ulteriori elaborazioni. Premere **RIPARTENZA**. Se si ripete la stessa situazione, chiamare il personale di manutenzione.

ELABORA. Questa spia si accende unicamente per far sapere che il sistema si trova nella fase di elaborazione anche se non compare nulla sullo schermo video. Dato che alcuni programmi necessitano di diversi minuti di elaborazione durante i quali lo schermo è come spento, la spia **ELABORA**, se accesa, ci assicura che il sistema sta eseguendo elaborazioni. Quando questa spia luminosa è accesa, il tasto **HOLD** non interrompe l'elaborazione.

CORREZIONE E STAMPA DELLE RIGHE IMMESSE

Se si individua un errore su una riga prima di aver battuto **EXECUTE**, onde immettere la riga nel 5100, si può far uso dei tasti di spazio indietro o spazio avanti per posizionare il cursore sull'errore e poi:

- Si può far uso delle funzioni di inserimento o di cancellazione per correggere l'errore.

- Si può battere il tasto ATTN per cancellare tutti i dati della riga a destra del cursore.
- Si può immettere il giusto carattere. E' da tener presente che i simboli APL (caratteri "maiuscoli" di molti tasti) della Figura 3 possono essere introdotti nella riga di immissione. Sulla tastiera degli elaboratori con il solo BASIC (Figura 4), tuttavia, questi simboli non compaiono quantunque possano essere immessi nella riga di immissione. Siccome diversi di questi simboli possono essere sovrapposti ad altri caratteri, si può usare la seguente sequenza di passi per sostituire un simbolo APL:
 1. Posizionare il cursore sul simbolo.
 2. Battere la barra di spaziatura per cancellare il simbolo.
 3. Far spazio indietro al cursore per portarlo nella posizione dello spazio vuoto.
 4. Immettere il giusto carattere.

Se si individua un errore su una riga dopo aver battuto EXECUTE per immettere la riga, si può far uso dei tasti di avanzamento od arretramento di riga per posizionare la riga che deve essere corretta nuovamente sulla linea di immissione, dopo di che si usano le procedure precedentemente esposte per correggere l'errore.

Se si desidera cambiare una riga intera, si può semplicemente immettere il numero della riga, poi reimmettere la riga e quindi battere EXECUTE. La nuova riga sostituirà la vecchia riga.

Se si desidera cancellare una o più righe di un programma, si deve immettere il numero della riga che si desidera cancellare, poi immettere DEL e battere EXECUTE. Per cancellare un certo numero di righe poste una successiva all'altra, immettere il numero della prima riga da cancellare, DEL, e infine il numero dell'ultima riga da cancellare; poi battere EXECUTE. Ad esempio per cancellare le righe dalla 20 alla 90 di un programma, basta immettere:

20 DEL 90

poi battere EXECUTE. Tener presente che DEL è sbagliato allorchè si tratta di Funzioni Tasto chiave (vedi).

CAPACITA' DELLA MEMORIA DEL 5100

Il modello base del 5100 (Modello B1) possiede una capacità di memoria di 16K (K = 1.024 bytes). La Figura 5 mostra come questa memoria sia distribuita per le diverse necessità. E' da tener presente che l'area di lavoro utilizzabile dall'utente è approssimativamente di 12.000 bytes, mentre 4.000 bytes vengono utilizzati per scopi interni. Nei seguenti altri modelli del 5100 la capacità di memoria è maggiore:

Modello B2 – 32K
 Modello B3 – 48K
 Modello B4 – 64K

In questi altri modelli la memoria aggiuntiva è stata assegnata all'area di lavoro dell'utente. Ad esempio nel Modello B4, l'area di lavoro è approssimativamente di 60.000 bytes.

11.000 Bytes Area di Lavoro dell'Utente (dati e programmi)
1.000 Bytes Area degli IMF proposta (vedi Comando PATCH, Capitolo 2)
4.300 Bytes (Esigenze interne del 5100)

Figura 5. Assegnazione della memoria per un 5100 a 16K

Capitolo 2. Comandi del Sistema

Alcuni comandi di sistema sono elencati al disopra della riga superiore dei tasti numerici sulla tastiera alfanumerica, diciamo, della macchina da scrivere (Figura 4). Questi comandi permettono di comandare le operazioni delle unità periferiche (nastro e stampatrice), quali la memorizzazione di un programma su nastro, il caricamento di un programma dal nastro nel 5100, e l'elaborazione del programma già caricato. I comandi del sistema possono sia essere immessi carattere per carattere tramite tastiera, oppure può essere immesso ad un tempo l'intero comando tenendo abbassato il tasto CMD e battendo l'appropriato tasto numerico sotto il comando (fatta eccezione per MARK, MERGE, e PATCH, i quali, non sono elencati). Quest'ultima operazione inserisce la parola chiave mediante una singola battuta, prevenendo così possibili errori di battuta e agevolando le operazioni con la maggior rapidità. I comandi del sistema vengono utilizzati per far eseguire dal 5100 i seguenti tipi di operazioni:

- Elaborazione di programmi — Avvia oppure riprende l'elaborazione di un programma BASIC oppure di un comando.
- Operazioni su files — Carica o memorizza programmi o dati su nastro oppure fa la marcatura di nastri.
- Operazioni di programma — Esegue la lista o la rinumerazione delle istruzioni di programmi oppure fonde diversi programmi in uno unico.

I parametri necessari per un comando possono essere immessi dopo che è stata immessa la parola chiave di comando. L'esecuzione del comando inizia solo dopo aver battuto il tasto EXECUTE. La parola chiave e le maggiori funzioni dei comandi del sistema BASIC sono:

LOAD	Carica un'area di lavoro (programmi o dati)
RUN	Esegue un programma BASIC
GO	Riprende l'elaborazione interrotta
SAVE	Memorizza l'area di lavoro (programmi o dati)
MERGE	Fonde l'area di lavoro
RENUM	Rinumera le istruzioni
AUTO	Numera automaticamente le righe
LIST	Visualizza o stampa il contenuto dell'area di lavoro (programmi o dati)
UTIL	Elabora le funzioni di utilità (stampa/visualizza un catalogo dei Contenuti dei nastri oppure prepara il 5100 a operazioni di tale comunicazione).

MARK Esegue la marcatura di files su nastro

PATCH Applica gli IMF (messa a punto interna della macchina), correzione del nastro oppure la copiatura del nastro.

REWIND Riavvolge il nastro

RD = Specifica la posizione di arrotondamento dei valori numerici stampati.

La sintassi e la descrizione di ciascun comando sono esposti in dettaglio in questa sezione del manuale. Tener presente che la sintassi usata per i comandi BASIC è ugualmente usata nelle istruzioni BASIC (Capitolo 4). In questa sintassi, i parametri che devono essere specificati così come appaiono, sono riportati a lettere maiuscole. I parametri facoltativi sono racchiusi tra parentesi quadre ([]). Parametri racchiusi tra parentesi graffe ({ }) stanno a significare che si deve immettere solamente uno dei parametri racchiusi tra esse. I puntini (....) indicano che i parametri precedenti possono essere ripetuti. Virgole singole, apici e parentesi devono essere immesse allorchè compaiono. Le virgole devono essere immesse per separare i parametri, fatta eccezione quando si trovano tra lparola chiave ed il primo parametro.

Ciascun comando o istruzione immessa tramite tastiera è controllato relativamente ad errori di sintassi. Se viene individuato un errore di sintassi, viene visualizzata una freccia verso l'alto (↑) al disotto della posizione della riga dove l'errore è stato individuato; lo schermo video si mette a lampeggiare e la tastiera si blocca. I tasti ATTN e HOLD sono gli unici tasti che restano attivi quando lo schermo video si mette a lampeggiare. Battere ATTN per interrompere il lampeggiamento dello schermo e poi correggere l'errore indicato.

Alcune regole generali che si applicano ai comandi del sistema sono:

- Non c'è bisogno di alcun numero di istruzione.
- Ciascun comando deve iniziare su una nuova riga.
- La massima lunghezza del comando può essere di 64 caratteri.
- Gli spazi vuoti vengono ignorati fatta eccezione per la serie di caratteri racchiusi tra virgolette singole (apice).
- I parametri devono essere tra loro separati da una virgola.

AUTO [numero di riga [,incremento]]

IL COMANDO AUTO

Il comando AUTO permette di iniziare una numerazione automatica di righe delle istruzioni BASIC. La numerazione automatica semplifica il compito di immissione delle istruzioni in un programma BASIC. Si può specificare sia il numero della istruzione iniziale che l'incremento tra i numeri di istruzione. Dopo aver immesso una istruzione ed aver battuto EXECUTE, il successivo numero di istruzione viene formato e visualizzato. La sintassi del comando AUTO è la seguente:

AUTO [numero di riga [, incremento]]

in cui:

numero di riga è un numero intero positivo il quale specifica il primo numero di istruzione che deve essere creato. L'intervallo di questo numero è tra 1 e 9999. Se non viene specificato un numero iniziale, vengono creati un numero iniziale 0010 ed un incremento 10.

incremento è un numero intero positivo usato per incrementare i numeri di istruzione che si succedono. Se non viene specificato un numero di riga iniziale, non può essere specificato l'incremento. Se l'incremento non è specificato, viene assunto automaticamente uguale a 10.

Ciascun numero di istruzione creato dal comando AUTO è seguito da uno spazio vuoto, poi dal cursore, così come mostrato:

0010_

Nota relativa ad AUTO

La numerazione automatica prosegue fino a quando viene immesso sulla riga di immissione un qualsiasi altro dato valido, una parola di comando, per esempio, od un numero di istruzione oltre a quello visualizzato. In questo caso, per proseguire la numerazione automatica si deve dare un altro comando AUTO. Un'altra maniera per concludere la numerazione AUTO è di battere il tasto di avanzamento di riga che visualizzerà una nuova riga priva di numero.

Esempio

I seguenti esempi spiegano il comando AUTO:

AUTO (poi battere EXECUTE)

In questo esempio lo schermo video mostrerà il numero di istruzione 0010. Dopo aver battuto EXECUTE alla fine di una istruzione immessa, il numero di istruzione automaticamente aumenta di 10, creando numeri di istruzione 0020, 0030, 0040 e così via.

Nell'esempio che seguirà il numero di istruzione iniziale è 0320. Dopo aver immesso ciascuna istruzione, il numero di istruzione sarà automaticamente incrementato di 20, creando così i numeri 0340, 0360, 0380 e così via.

AUTO 320, 20 (poi battere EXECUTE)

GO [numero di riga] [, { RUN STEP TRACE [,PRINT] }] [, RD = n]
--

IL COMANDO GO

Si può usare il comando GO per risolvere quanto segue:

1. Riprendere l'elaborazione di un comando MARK che è stata interrotta.
2. Riprendere l'elaborazione di un programma BASIC che è stata interrotta da:
 - a. una istruzione PAUSE,
 - b. elaborazione passo-passo (vedi Comando RUN),
 - c. alcune condizioni di errore, oppure
 - d. aver battuto il tasto ATTN.
3. Finire una operazione di comando MARK o un programma BASIC prima che abbia raggiunto il suo normale punto finale.

Si può riprendere l'elaborazione battendo EXECUTE su una linea di immissione vuota. Il comando GO, tuttavia, permette di riprendere l'elaborazione in ciascuno dei tre stati di elaborazione (normale, passo-passo, oppure con traccia). Così, si può cambiare lo stato di elaborazione mediante il comando GO. L'elaborazione del programma può essere proseguita con la successiva istruzione sequenziale oppure al numero di istruzione specificato nel comando GO. Si può inoltre cambiare la posizione decimale per l'arrotondamento (vedi il Comando RD=).

Per concludere l'elaborazione di un comando del sistema od un programma BASIC, il comando GO possiede la seguente sintassi

GO END

END chiude i files di immissione e di emissione, conservando così l'integrità degli stessi. Dopo che i files sono stati chiusi, non vengono elaborate altre istruzioni di programma.

Per proseguire l'elaborazione di una operazione interrotta, il comando GO possiede la seguente sintassi:

$$\text{GO [numero di riga] } \left[, \left\{ \begin{array}{l} \text{RUN} \\ \text{STEP} \\ \text{TRACE [,PRINT]} \end{array} \right\} \right] [\text{,RD=n}]$$

dove:

numero di riga è il numero della istruzione alla quale si deve riprendere l'elaborazione. Se questo numero viene omesso, l'elaborazione inizia con la successiva istruzione sequenziale o dove il comando è stato interrotto.

RUN specifica che l'elaborazione prosegue in stato normale.

STEP specifica che l'elaborazione prosegue ora in stato di "passo-passo".

TRACE specifica che l'elaborazione prosegue nello stato di "trace".

Nota: Se non viene specificato nè RUN, nè STEP, nè TRACE, l'elaborazione proseguirà nello stato che era attivo allorchè l'elaborazione è stata interrotta.

PRINT specifica che i messaggi di "trace" devono essere visualizzati e stampati. Se PRINT viene omesso, i messaggi vengono unicamente visualizzati. Questa opzione si potrà specificare solamente con TRACE.

RD=n indica il numero di cifre a destra del punto decimale a partire dal quale inizia l'arrotondamento. n può andare da 1 a 13 ed è inizializzato a 6. Se non viene specificato, n conserva il suo ultimo valore.

Note relative a GO

- L'immissione del numero di istruzione è valida solamente durante l'elaborazione di un programma BASIC.
- Quanto la riga di immissione è vuota, battendo EXECUTE si causa la medesima operazione di un comando GO, fatta eccezione di quando GO è immesso in risposta ad un errore causato dal tentativo di marcare un file già marcato in precedenza. In questo caso, GO deve essere battuto unicamente nelle posizioni 1 e 2 della riga di immissione.

Esempio

Gli esempi che seguiranno mostrano una varietà dei comandi GO.

1. Per cambiare oppure riprendere la normale operazione di un programma BASIC:

GO RUN (poi battere EXECUTE)

2. Per cambiare il tipo di elaborazione da "passo-passo" o stato normale in stato di "trace":

GO TRACE (poi battere EXECUTE)

3. Per cambiare in stato di "passo-passo" ed iniziare l'elaborazione al numero di istruzione 620:

GO 620, STEP (poi battere EXECUTE)

$\text{LIST [PRINT] } \left[, \left\{ \begin{array}{l} \text{KEYx} \\ \text{numero di riga} \end{array} \right\} \right]$
--

IL COMANDO LIST

Il comando LIST permette di visualizzare o stampare il programma o le righe di dati prelevandoli dall'area di lavoro. Il contenuto dell'area di lavoro non viene modificato. Il contenuto deve essere BASIC, DATA oppure un gruppo KEY (che ha bisogno di una sintassi LIST diversa). La sintassi del comando LIST è:

$$\text{LIST [PRINT] } \left[, \left\{ \begin{array}{l} \text{KEYx} \\ \text{numero di riga} \end{array} \right\} \right]$$

dove:

PRINT specifica che la lista delle righe che sono nell'area di lavoro deve essere stampata piuttosto che visualizzata. Se non viene specificato PRINT, la lista sarà solo visualizzata.

KEYx specifica che verrà visualizzato o stampato la Funzione-Tasto indicata (x=0 a 9)

numero di riga specifica che un gruppo di 14 righe, che finisce con il numero di riga specificato, dovrà essere visualizzato. Se verrà specificato anche PRINT, sarà stampata l'intera area di lavoro ad iniziare con il numero di riga indicato. Se non viene immesso alcun numero di riga, si assume per la stampa il più piccolo numero di riga che si trova nell'area di lavoro. Questa immissione non è valida per un gruppo KEYx.

Quando si specifica la lista sullo schermo, la riga specificata nel comando LIST compare sulla riga 2 dello schermo video. Le tredici righe precedenti (o meno, se ne sono definite meno di 14) vengono visualizzate sulla riga al disopra della riga 2. Si può far uso dei tasti di avanzamento od arretramento di riga (↓ e ↑) per raggiungere una riga in particolare.

Quando viene richiesta la stesura della lista sulla stampatrice, viene stampata l'intera area di lavoro ad iniziare dal numero di riga specificato nel comando LIST. La stampa inizierà con il numero di riga più piccolo se nel comando LIST non è stato specificato esplicitamente un numero di riga.

Note relative a LIST

- Se il numero di riga specificato nel comando LIST non si trova nell'area di lavoro, il 5100 andrà a cercare il numero di riga successivo più piccolo per l'operazione LIST. Se non viene trovato un numero di riga più piccolo, verrà visualizzato un messaggio di errore.
- Se la lunghezza di riga supera i 64 caratteri, la riga successiva conterrà, allorchè sarà visualizzata, tutto ciò che supera i 64 caratteri in quella precedente. Se venisse stampata, l'intera riga sarà stampata con un massimo di 128 caratteri. La lunghezza della riga non influisce sull'elaborazione della istruzione.
- Se si effettua la lista di righe di dati più lunghe di 118 caratteri (bytes), quanto supera i 118 caratteri non viene visualizzato.

Esempio

Gli esempi che seguono mostrano alcuni comandi LIST:

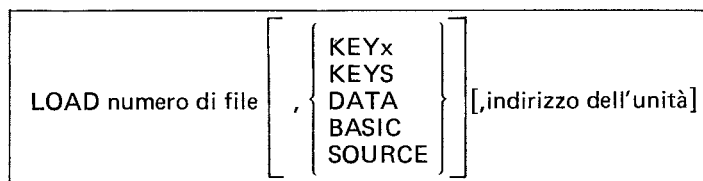
- Per visualizzare il primo gruppo delle righe che sono nell'area di lavoro:

LIST (poi battere EXECUTE)
- Per visualizzare il numero di riga 250 e le 13 righe precedenti:

LIST 250 (poi battere EXECUTE)
- Per stampare il numero di riga 250 e tutte le righe che seguono nell'area di lavoro:

LIST PRINT, 250 (poi battere EXECUTE)
- Per visualizzare le prime 14 righe che si trovano nel gruppo associato alla funzione-tasto 5:

LIST KEY5 (poi battere EXECUTE)



IL COMANDO LOAD

Il comando **LOAD** provoca il caricamento da nastro magnetico, di un file precedentemente memorizzato, nell'area di lavoro per effettuare delle modifiche e/oppure eseguirne l'elaborazione. **LOAD** può inoltre essere usata per preparare l'area di lavoro per una immissione tramite tastiera di un nuovo programma BASIC, di dati, oppure di una Funzione-tasto. La sintassi del comando **LOAD** è come sotto esposto:

$$\text{LOAD numero del file} \left[, \left\{ \begin{array}{l} \text{KEY}_x \\ \text{KEYS} \\ \text{DATA} \\ \text{BASIC} \\ \text{SOURCE} \end{array} \right\} \right] [\text{,indirizzo dell'unità}]$$

dove:

numero del file è un valore intero. L'immissione di uno 0 indica che si immetterà tramite tastiera un nuovo file. Un numero maggiore di 0 indica il numero del file (memorizzata su nastro) che si vuole caricare.

KEY_x, **KEYS**, **DATA**, **BASIC** o **SOURCE** specificano il tipo di dati che si vogliono caricare nell'area di lavoro. Il **BASIC** è assunto quando si carica tramite tastiera. Il **BASIC** o **SOURCE** è ciò che si assume quando si carica da nastro.

indirizzo dell'unità è il codice di indirizzo dell'unità contenente un file memorizzato e che deve essere caricato. Se non specificato, si assume che si all'unità nastro incorporata (indirizzo E80). L'indirizzo del dispositivo dell'Unità Nastro Ausiliaria è E40.

L'immissione di un comando **LOAD0**, azzerà l'intera area di lavoro, incluse tutte le funzioni-tasto. Un comando **LOAD** che specifichi **BASIC** oppure **DATA** come secondo parametro, azzererà unicamente nell'area di lavoro il programma specificato, senza alterare le Funzioni-tasto esistenti. Se il nuovo tipo di file è un file **KEY_x**, il nuovo file di Funzione-tasto viene aggiunta all'area di lavoro senza alterare l'area di lavoro esistente. Se i files **KEY_x** esistono già nella memoria, i files **KEY_x** caricati sostituiranno quelli vecchi. L'area di lavoro viene assegnata al tipo di file specificato nel comando **LOAD**. La riga 0 visualizzerà **READY**, nel caso si tratti di un file tipo programma **BASIC**; **DATA** per un tipo di file di dati, e **KEY** per un tipo di file di Funzioni-tasto. Si può cominciare immettendo il tipo specificato di righe nell'area di lavoro. Il tipo di righe immesso (**BASIC**, **DATA** oppure **KEY_x**) deve possedere lo stesso tipo di definizione di file dell'area di lavoro altrimenti la riga non sarà accettata e comparirà sulla riga 0 un messaggio di errore. **KEYS** e **SOURCE** non sono opzioni valide quando si carica tramite tastiera (va bene **KEY_x**).

Un comando **LOAD** diverso da zero specifica che si deve caricare dal nastro nell'area di lavoro un file memorizzato. Qualsiasi contenuto precedente dell'area di lavoro (fatta eccezione per le Funzioni-tasto) viene cancellato quando il tipo di file viene specificato come **BASIC**, **DATA** oppure **SOURCE**. Se il nuovo tipo di file è un file **KEYx**, allora la nuova Funzione-tasto viene aggiunta all'area di lavoro senza modificare l'area di lavoro esistente. Se nella memoria esistono già files **KEYS/KEYx**, i nuovi files **KEYS/KEYx** caricati sostituiranno quelli vecchi. Una copia dei files memorizzati è caricata nell'area di lavoro e viene visualizzata la seguente informazione:

- L'identificazione del file (se esiste) specificata dall'utente
- Il numero di aree di memoria di 1024 byte contigui nel file (numero di blocchi)
- Il numero di aree di memoria contigue non occupate, di 1024 byte ciascuna
- Il primo numero di riga (per **BASIC** o **SOURCE**)
- L'ultimo numero di riga (per **BASIC** o **SOURCE**)
- I numeri **KEYx** (per i tipi di files **KEYx**)
- L'ammontare di area di lavoro rimanente (in bytes) non assegnata.

Se il nuovo tipo di file è **KEYx**, la nuova Funzione-tasto viene aggiunta all'area di lavoro. Se esistono già nella memoria i corrispondenti files **KEYx**, essi vengono sostituiti dalla nuova Funzione-tasto. Ad esempio, se una nuova funzione **KEY6** è caricata da nastro, essa sostituirà la funzione **KEY6** esistente nell'area di lavoro.

Note relative a **LOAD**

- Il tipo di file **KEYS** ("carica tutte le Funzioni-tasto nel file specificato") non va bene quando il numero di file è zero (**LOAD0**).
- Il tipo di file **SOURCE** ("carica il programma nel formato origine dell'utente dal file specificato") non va bene quando si carica tramite tastiera (**LOAD0**).
- Quando si carica un file di interscambio (tipo **02**), si assume **SOURCE** a meno che sia specificato esplicitamente **DATA**.
- Se il tipo di file viene omissso, si assume **BASIC** se si carica da tastiera, e si assume **BASIC** oppure **SOURCE** se si carica a partire da un file memorizzato su nastro (dipendendo dal tipo di file). Se nell'ultimo esempio il tipo di file non è **BASIC** oppure **SOURCE**, viene visualizzato un messaggio di errore.
- Il formato per immettere dei dati dopo un **LOAD0**, **DATA** è identico a quello dei dati in una specifica **DATA**. **LOAD0**, **DATA** inoltre fornisce la numerazione di riga automatica e le parole chiave **DATA** per ciascuna riga. Lo schermo mostrerà:

- Se viene interrotta l'elaborazione di un comando LOAD (quando si batte ATTN) mentre si carica un file memorizzato, l'operazione finisce ed il contenuto dell'area di lavoro viene cancellato.
- Comandi dell'elaboratore sono validi se viene specificato DATA (LOADx, DATA).
- LOAD0, DATA fornisce automaticamente un numero di istruzione a quattro cifre e la parola chiave DATA sulla riga di immissione nel file di dati.

Note Relative a LOAD SOURCE

- Quando capita un errore di sintassi durante un comando LOAD SOURCE, si può scegliere tra una delle opzioni che seguono:
 1. Correggere la riga e battere EXECUTE per proseguire.
 2. Avanzare di una riga per ignorare la riga e proseguire il caricamento.
- Se si carica un file SOURCE che contenga una riga più lunga di 64 caratteri, si commetterà un errore. L'errore deve essere corretto oppure bisogna usare l'avanzamento di riga (che fa ignorare la riga) prima di poter continuare a caricare.
- Il comando LOAD SOURCE non è completato fino a che compare il messaggio READY oppure fino a che l'operazione venga completata da qualcosa che non sia un errore di sintassi.

Esempio

Gli esempi che seguono mostrano una varietà dei comandi LOAD:

- Per preparare una immissione di un file di dati da tastiera:
LOAD0, DATA
- Per preparare l'immissione di tastiera della funzione-tasto 6:
LOAD0, KEY6
- Per caricare un programma memorizzato (file 3) dell'unità nastro E40 (Unità Nastro Ausiliaria):
LOAD3, E40
- Per caricare un programma memorizzato nel formato "source" dell'utente dal file 6 sull'unità nastro E80:
LOAD6, SOURCE
oppure
LOAD6

FILE DI DATI CREATI TRAMITE TASTIERA

E' possibile creare un file di dati direttamente tramite tastiera con l'uso del comando LOAD. Per prima cosa immettere LOAD0, DATA, poi battere EXECUTE. Il 5100 risponde con la numerazione automatica delle righe (ad iniziare da 0010) seguita da DATA. Si possono allora immettere costanti numeriche o carattere, separate da virgole, così come si farebbe per una istruzione DATA standard (vedi DATA, Capitolo 4). Si indica la fine della riga di un file di dati battendo il tasto EXECUTE. Se si commette un errore di battuta, lo schermo video si metterà a lampeggiare e la tastiera si bloccherà. Battere semplicemente ATTN, poi correggere l'errore, il quale è indicato dalla freccia verso l'alto (↑). Una tipica riga di file di dati è:

0110 DATA 312.41, 'JONES', 419.21, 'BALANCE'

Dopo che sono state immesse nell'area di lavoro tutte le righe del file di dati, queste possono essere memorizzate su nastro mediante il comando SAVE. Le righe del file di dati vengono memorizzate senza i numeri di riga o la parola chiave DATA. Quando viene eseguita la lista delle righe del file di dati, viene nuovamente visualizzata la parola chiave DATA.

Si può accedere ai files di dati creati tramite tastiera, durante l'elaborazione di un programma, mediante le istruzioni GET o MAT GET. Con l'eliminazione dei numeri di riga e della parola chiave DATA, il file di dati diventa una serie continua di dati.

Un file di dati memorizzato su nastro può essere corretto caricandolo inizialmente nell'area di lavoro con un comando LOAD. Nel memorizzarlo su nastro, vengono eliminati i numeri di riga e DATA.

Funzioni-tasto

Per il nostro uso nel richiamare programmi o comandi di nostra scelta sono disponibili dieci funzioni-tasto. I tasti delle funzioni-tasto sono solo i tasti numerici da 0 a 9 posti alla destra della tastiera. Le funzioni richiamate mediante questi tasti vengono definite in un comando LOAD. Per prima cosa si immette un comando LOAD il quale specifichi il tasto al quale si deve assegnare una funzione. Ad esempio, per assegnare una funzione al tasto 6, immettere:

LOAD0, KEY6 (poi battere EXECUTE)

Poi si deve definire la Funzione-tasto vera e propria, che non è costituita da altro che dalle istruzioni associate alla funzione che viene definita. Queste istruzioni verranno elaborate quando si terrà premuto il tasto CMD e si batterà il tasto 6. Una delle quattro istruzioni valide per l'istestazione della Funzione-tasto deve essere preceduta da un numero di riga pari a 999x, in cui x = 0 a 9 rappresenta il tasto della funzione. Nell'esempio precedente, il numero di riga sarebbe 9996. Le istruzioni di testata sono:

- NULL la quale specifica che non si sta definendo una funzione per un tasto particolare o che si sta cancellando una funzione esistente.

- CMD la quale specifica che il resto della riga è rappresentato da un comando del sistema od una istruzione dell'elaboratore. Questo gruppo chiave può contenere un solo comando del sistema od una sola istruzione dell'elaboratore. L'operazione specificata viene elaborata immediatamente (senza battere EXECUTE).
- REM che specifica che il gruppo chiave è costituito da una serie di istruzioni BASIC. Queste saranno elaborate non appena, tenendo premuto il tasto CMD, sarà battuto il tasto specificato (oppure allorchè la funzione-tasto specificata sarà posta come riferimento in GOSUB). Non sono però permesse le istruzioni DATA, END, STOP, DEF e FNEND. Così anche non si dovranno immettere READ, MAT READ. Inoltre, una Funzione-tasto non potrà contenere un riferimento ad una funzione definita dall'utente. L'ultima istruzione BASIC deve poi essere RETURN o CHAIN; RETURN non deve contenere una espressione. Allorchè sarà posto come riferimento in GOSUB (999x), il programma salta e ricerca la Funzione-tasto REM. Solo se REM non viene trovato, il programma ricerca l'istruzione 999x nel programma.
- TXT, specifica che la stringa di caratteri (racchiusi tra apici) la si deve inserire nella riga di immissione ad iniziare dalla posizione corrente del cursore. Ad esempio, assumendo che una serie di caratteri 'RATE' venga assegnata alla funzione chiave 5, ogni volta che il tasto 5 della funzione chiave è battuto, tenendo contemporaneamente premuto CMD, la costante RATE viene inserita nella riga di immissione.

Negli esempi che seguono, al tasto 6 verranno assegnati un certo numero di funzioni. In primo luogo il comando LOAD0, KEY6 prepara l'area di lavoro per l'assegnazione della funzione-tasto al tasto 6.

- 9996 NULL indica che il tasto 6 non avrà associata alcuna funzione o, se l'aveva essa viene ora cancellata.
- 9996 CMD REWIND E80 indica che il tasto 6 farà in modo che il nastro che si trova nella unità nastro incorporata venga riavvolto ogni volta che, tenendo premuto il tasto CMD, venga battuto il tasto 6.
- 9996 REM KEY6 FUNCTION
0001 FOR X = 100 a 360 STEP 10
0002 A = X/12
0003 I = 2*A
0004 PRINT FLP,I,X
0005 NEXT X
0006 RETURN

Queste istruzioni BASIC verranno elaborate ogni volta che, tenendo premuto il tasto CMD, verrà battuto il tasto numerico 6.

- 9996 TXT'SPINDLES AND SPANGLES' indica che tale serie di caratteri verrà inserita nella riga di immissione ad iniziare dalla posizione corrente del cursore ogni volta, che tenendo premuto il tasto CMD, verrà battuto il tasto 6.

Un ulteriore utilizzo della funzione-tasto è mostrato qui sotto:

```
9996 CMD A=&PI*R↑2
```

In questo esempio è possibile calcolare l'area di un cerchio assegnando un valore ad R (il raggio) e battendo il tasto numerico 6 tenendo premuto il tasto CMD.

Tener presente che le istruzioni in una Funzione-tasto REM possono essere cancellate mediante riferimento alla Funzione-tasto e facendo uso della funzione di correzione DEL.

Per esempio:

```
KEY6, 10DEL
```

cancella la riga 10 dalla Funzione-tasto 6.

```
KEY6, 10DEL90
```

cancella le righe dalla 10 alla 90 della Funzione-tasto 6.

Allo stesso modo si possono aggiungere nuove istruzioni:

```
KEY6, 10PRINT A
```

aggiunge la riga 10 oppure sostituisce nella Funzione-tasto 6 la vecchia riga 10.

Tener presente che KEYx non può essere utilizzato per correggere una istruzione di testata di una Funzione-tasto.

IL COMANDO MARK

Il comando MARK si può usare per inizializzare uno o più files di nastro con un determinato numero di aree di memoria contigue di 1024 bytes ciascuna. Se si arriva alla fine del nastro prima di aver completato l'operazione di marcatura, vengono visualizzati l'ultimo numero di file ed il numero di aree di 1024 byte usate nel file. Il tasto ATTN non è operativo durante le operazioni di marcatura. Se si fa il tentativo di marcare un file che è già stato marcato in precedenza, viene visualizzato un messaggio di errore (Appendice B). Per proseguire, battere GO. Tener presente che si deve usare il tasto di avanzamento di riga per spostare la riga di immissione prima di immettere GO, tra l'altro accettato solo nelle prime due posizioni di riga. Se si finisce l'operazione immettendo qualsiasi cosa al posto di GO nelle posizioni 1 e 2, il file precedentemente marcato resta immutato.

Nota: Se un file già presente su nastro viene marcato, le informazioni originarie che si trovano nel file marcato nonché quelle esistenti nei files che seguono quello rimarcato, non possono più essere nuovamente usate.

La sintassi del comando MARK è come mostrato qui oltre:

MARK K-caratteri, files, file iniziale [, indirizzo dell'unità]

dove:

K-caratteri è il numero di aree di memoria di 1024 byte contigue che si devono usare nel nastro.

files è il numero di files consecutivi che si devono marcare.

file iniziale è il numero di file iniziale a partire dal quale deve iniziare la marcatura.

indirizzo dell'unità è l'indirizzo dell'unità nastro in cui risiede il file. In sua mancanza si assume che l'indirizzo sia E80.

Note Relative a MARK

- Il comando MARK lo si deve immettere carattere per carattere tramite tastiera.
- Il comando MARK può essere dato ogni qualvolta l'unità nastro specificata non sia altrimenti attiva.
- Nastri che contengono errori CRC devono essere rimarcati (reinizializzati) a partire da un file che preceda l'errore CRC. Generalmente è necessario un comando REWIND prima di questo MARK.
- Si può determinare la grandezza necessaria per un file confrontando l'ammontare dell'area di lavoro prima e dopo avere immesso dei dati o programmi nell'area di lavoro.
Tener presente che ciascuna espressione numerica necessita in media

di 16 posizioni di memoria (bytes) mentre ciascuna espressione di caratteri necessita di 21 posizioni di memoria meno una posizione per ciascun spazio vuoto di coda.

Esempio

Un esempio di comando MARK è mostrato quì sotto:

MARK, 3, 6, 1

In questo esempio saranno marcati 6 files, ad iniziare dal file 1, con ciascun file che occupa 3072 bytes sul nastro che risiede nell'unità nastro incorporata (E80).

MERGE numero di file [, [dal numero di riga] , [al numero di riga] , [nuovo numero di riga] [, indirizzo dell'unità]]

IL COMANDO MERGE

Il comando MERGE permette di fondere assieme tutto o parte di un file memorizzato assieme a dati o ad un programma (memorizzato nel formato BASIC) nell'area di lavoro. In questa maniera, ad esempio si può aggiungere la stessa routine a svariati diversi programmi, oppure si possono aggiungere gli stessi dati a svariati diversi files di dati. Solamente istruzioni BASIC (in un file BASIC) o file DATA possono essere prese. L'area di lavoro e il file memorizzato su nastro devono essere dello stesso tipo. Se questi files sono differenti, il comando MERGE non viene elaborato, e viene visualizzato un messaggio di errore. Le righe da un file su nastro vengono aggiunte alle righe dell'area di lavoro della sequenza espressa dai numeri di riga. Se una riga che si trova in un file su nastro ed una riga che si trova nell'area di lavoro possiedono lo stesso numero di specifica, allora la riga del file sostituisce la riga dell'area di lavoro. Il file che è stato preso potrebbe superare la grandezza dell'area di lavoro. In questo caso verrà visualizzato un messaggio di errore (vedi Appendice B). In questo caso verrà visualizzato un messaggio di errore (vedi Appendice B).

Una volta che le righe sono state fuse, è possibile specificare che le righe del file fuso vengano rinumerate, ad iniziare da un numero di istruzione di nostra scelta e con incrementi uguali agli incrementi del file originario. Dopo che l'operazione di fusione è stata completata, lo schermo mostrerà il messaggio READY, assieme al numero di bytes non usati nell'area di lavoro. La sintassi del comando MERGE è come mostrato qui oltre:

MERGE numero di file , [dal numero di riga] , [al numero di riga] , [nuovo numero di riga] [, indirizzo dell'unità]

dove:

numero di file è il numero del file memorizzato su nastro che deve essere fuso.

dal numero di riga rappresenta la prima riga del file memorizzato su nastro che deve essere fuso. Se non viene immesso alcun numero si assume che sia la prima riga del file.

al numero di riga è l'ultima riga del file memorizzato che deve essere fuso. Se non viene immesso alcun numero, si assume che sia l'ultima riga del file.

nuovo numero di riga rappresenta il numero dalla prima riga che si deve usare nella rinumerazione delle righe del file memorizzato. Se non viene immesso alcun numero, il file che è stato fuso non sarà rinumerato.

indirizzo dell'unità è l'indirizzo dell'unità nastro nella quale risiede il file memorizzato. In mancanza di esso si presume sia quello dell'unità nastro incorporata (E80).

Note Relative a MERGE

Questo comando deve essere immesso carattere per carattere tramite tastiera.

I parametri che vengono omessi li si deve indicare mediante virgole consecutive. Per esempio:

MERGE 6, 2, 200, ,E80



Nuovo numero di riga omesso

Esempio

Un esempio del comando MERGE è:

MERGE 6,2,200,10,E40

In questo esempio, i dati che si trovano nel file numero 6 nell'unità nastro ausiliaria, saranno fusi con i dati che si trovano nell'area di lavoro. Saranno fuse le righe da 2 a 200 del file. Una volta che il comando MERGE è stato elaborato, le righe del file memorizzato saranno rinumerate ad iniziare dalla riga numero 0010.

IL COMANDO PATCH

Il comando PATCH permette di applicare la messa a punto interna (IMFs) del 5100. La messa a punto interna della macchina viene fornita e dalla IBM con una cartuccia magnetica di supporto al cliente. Il comando PATCH fornisce la possibilità di:

- Copiare gli IMF (Internal Machine Fix) in una nuova cartuccia.
- Caricare gli IMF nella memoria e poi tornare all'APL o BASIC.
- Visualizzare la versione EC (Engineering Changes: modifiche tecniche) di ciascun modulo interprete.
- Eseguire la correzione di un nastro allorché accadono gli errori di lettura (007) in files dei seguenti tipi:
 - Interscambio di dati (tipo di file 01)
 - Interscambio generale (tipo di file 02)
 - BASIC "Origine" (tipo di file 02)
 - Dati interni APL (tipo di file 08)
- Copiare il contenuto di una cartuccia a nastro in un'altra.

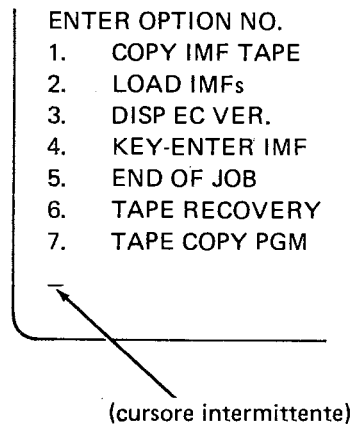
I programmi di copia degli IMF, caricamento degli IMF, visualizzazione della versione EC e correzione del nastro vengono provvisti su una cartuccia nastro fornita assieme al 5100. Questa cartuccia contiene i seguenti files:

- Il file 1 contiene:
 - Il programma per copiare gli IMF
 - Il programma per caricare gli IMF
 - Il programma per visualizzare la versione EC.
- Il file numero 2 contiene gli IMF per il 5100.
- Il file numero 3 contiene il programma per la correzione di un nastro.
- Il file numero 4 contiene il programma per copiare un nastro.

Per ottenere uno di questi programmi:

- Inserire la cartuccia nastro nell'unità nastro incorporata
- Battere PATCH.
- Battere EXECUTE

Non esistono parametri per il comando PATCH. Allorchè il comando PATCH viene immesso ed elaborato, vengono visualizzate le seguenti opzioni:



Lo schermo mostra le opzioni disponibili, ciascuna delle quali è identificata da un numero.

Per cominciare l'elaborazione della opzione scelta:

- Immettere il numero di opzione.
- Battere EXECUTE.

L'immissione di un numero diverso da tutti quelli visualizzati sullo schermo determina una nuova visualizzazione delle opzioni.

Una volta che è stato immesso un numero di opzione, vengono visualizzate, se esistono, delle richieste aggiuntive per la opzione scelta.

Opzione 1. Copiare su nastro gli IMF

L'opzione per copiare su nastro gli IMF permette di copiare dalla cartuccia i seguenti files:

- File numero 1 che contiene:
 - Il programma per copiare gli IMF
 - Il programma per caricare gli IMF
 - Il programma per visualizzare la versione EC
- File numero 2 che contiene gli IMF per il 5100. Gli IMF vengono copiati dal file 2 nella seguente maniera:
 - Copia tutti gli IMF che si applicano al BASIC o all'APL.
 - Copia tutti gli IMF che si applicano al 5100 su cui è in funzione il programma del nastro per copiare gli IMF.
 - Copia gli IMF per numero di problema.
 - Copia gli IMF per numero di problema che si applica al 5100 su cui è in funzione il programma del nastro per copiare gli IMF.

- File 3, che contiene il programma per la correzione di un nastro, e il File 4, che contiene il programma per copiare un nastro. Questi files possono essere copiate unicamente dal personale di manutenzione.

Nota: Le cartucce nastro nelle quali si devono copiare i files 1 e 2, debbono essere marcate. Per determinare con quale lunghezza marcare le files, far uso del comando UTIL.

Copiando gli IMF, si può creare una cartuccia che contenga unicamente i soli IMF che si applicano al nostro 5100.

Il programma per copiare gli IMF su nastro emette delle richieste ed attende che gli si risponda. Pertanto rispondere, se necessario, a queste richieste emesse dal programma. Vedere in Appendice una lista incrociata inglese-italiano dei messaggi emessi dal programma PATCH.

Opzione 2. Caricamento degli IMF

L'opzione caricamento degli IMF permette di caricare gli IMF nella memoria e poi tornare al linguaggio che è in esecuzione nel 5100.

Gli IMF possono essere caricati nella seguente maniera:

- Caricare tutti gli IMF che si applicano al 5100 su cui è in esecuzione il programma
- Caricare gli IMF per numero di problema. Gli IMF vengono caricati nella memoria unicamente se questi si applicano al 5100 su cui il programma è in esecuzione.

Gli IMF occuperanno memoria nell'area di lavoro (spazio di lavoro), pertanto gli IMF non saranno applicati se il problema non riguarda le loro operazioni oppure se il problema può essere aggirato mediante una istruzione oppure un comando BASIC. Gli IMF permarranno nell'area di lavoro fino a che l'elaboratore sarà stato spento oppure sarà pigiato RESTART. Inoltre, se gli IMF vengono caricati nella memoria, le prestazioni del 5100 saranno ridotte significativamente.

Il programma per caricare gli IMF emette delle richieste ed attende che gli si risponda. Pertanto rispondere se necessario alle richieste; vedere in Appendice una lista incrociata inglese-italiano dei messaggi emessi dal programma PATCH.

Opzione 3. Visualizzazione della versione EC

L'opzione per visualizzare la versione EC è in primo luogo riservata all'uso del personale di manutenzione. Questa opzione visualizza la versione EC di ciascun modulo interprete e modulo d'identificazione. Per ciascun modulo vengono visualizzate quattro cifre. Le prime due cifre rappresentano l'identificazione del modulo, le successive due rappresentano la versione EC.

Il programma emette delle richieste ed attende che gli si risponda. Pertanto rispondere se necessario alle richieste emesse dal programma.

Opzione 4. Immissione a tasti degli IMF

L'opzione immissione a tasti degli IMF permette al personale di manutenzione di immettere un IMF tramite tastiera. Dopo averlo immesso tramite tastiera, l'IMF viene copiato su nastro. Lo si può allora caricare nella memoria mediante l'opzione 2 (caricamento degli IMF).

Opzione 5. Fine del Lavoro

La scelta di questa opzione determina un ritorno al linguaggio con cui il 5100 stava elaborando. Dopo il completamento delle opzioni 1, 3 e 4 il programma provvede a far visualizzare tutte le opzioni. Pertanto si può allora scegliere l'opzione 5.

Opzione 6. Correzione del Nastro

L'opzione correzione del nastro permette di correggere uno o più file di dati nei quali sono avvenuti degli errori di lettura. Il programma di correzione del nastro può essere usato nei seguenti tipi di files:

- Interscambio di dati (tipo di file 01)
- Interscambio generale (tipo di file 02)
- BASIC "Origine" (tipo di file 03)
- Dati interni APL (tipo di file 08)

La correzione del nastro è in grado di correggere quanti dati si desidera in un file. Alcuni dei dati che sono nella registrazione in cui sono capitati degli errori di lettura non possono essere corretti. Alcuni dei dati che precedono o che seguono la registrazione in cui sono capitati degli errori di lettura potrebbero anche non essere corretti.

La correzione del nastro presume che la cartuccia contenente i files, che hanno causato gli errori di lettura, sia inserita nell'unità nastro incorporata.

Il programma correzione del nastro emette delle richieste ed attende che gli si risponda. Pertanto rispondere se necessario alle richieste emesse dal programma. Vedere in proposito una lista incrociata dei messaggi tradotti (inglese-italiano) in Appendice.

Opzione 7. Programma per Copiare un Nastro

L'opzione per copiare un nastro permette di copiare tutto il contenuto (fino alla fine di un nastro marcato) di una cartuccia in un'altra cartuccia. La opzione per copiare un nastro può utilizzare, se disponibile, l'unità nastro ausiliaria. La stessa opzione inoltre, provvede a marcare il nastro sul quale si copia, sia se esso è già marcato che nuovo.

La opzione per copiare un nastro emette delle richieste ed attende che si risponda alle sue richieste. Vedere a proposito l'Appendice per una lista di messaggi incrociati inglese-italiano emessi durante questa fase.

IL COMANDO = RD

E' possibile usare il comando RD= per specificare il numero di cifre a destra del punto decimale che debbono essere stampate o visualizzate. Se deve essere stampato o visualizzato un numero maggiore di cifre, queste saranno arrotondate, al numero specificato. All'accensione dell'elaboratore l'arrotondamento è inizialmente impostato alla sesta decimale. L'arrotondamento lo si può modificare mediante il comando RD= oppure nei comandi RUN o GO. L'intervallo di arrotondamento (n) va da 1 a 13 cifre. Un errore compare per $n \leq 0$.

RENUM [KEYx] [,primo numero di riga, incremento]

IL COMANDO RENUM

E' possibile usare il comando RENUM (rinumerare) per creare dei nuovi numeri di istruzione per tutte le istruzioni BASIC od i dati che si trovano nell'area di lavoro. Simile al comando AUTO, la rinumerazione inizia con 0010 ed ha degli incrementi di 10, a meno che si specifichi diversamente. Inoltre, tutti i riferimenti ai numeri di istruzione che si trovano nelle istruzioni GOTO, IF, PRINT USING, GOSUB, GET e MAT GET vengono trasformati nei nuovi numeri. Nelle funzioni-tasto (vedi Comando LOAD), ciascun gruppo viene rinumerato nella stessa maniera come se fosse un'area di lavoro separata; inoltre sarà rinumerato unicamente se è una funzione di programma BASIC. (RENUM non altera il record della testata della Funzione-tasto).

La sintassi del comando RENUM è come segue:

RENUM [KEYx] [, primo numero di riga, incremento]

dove:

KEYx specifica una Funzione-tasto (x=da 0 a 9) che deve essere rinumerata.

primo numero di riga è un numero intero (1-9989) che identifica il numero a partire dal quale inizierà la rinumerazione. Se tale numero non viene specificato si assumono come valori un numero iniziale uguale a 0010 ed un incremento di 10.

incremento è un numero intero il quale specifica l'incremento che devono avere i numeri successivi di istruzione, se però non viene specificato un numero iniziale di istruzione, non è possibile specificare l'incremento.

Note relative a RENUM

- Si deve immettere o un numero di istruzione iniziale od un incremento, oppure non si immette nulla.
- I numeri di istruzione 9990-9999 non vengono modificati da un comando RENUM.
- Si avrà un errore se in una rinumerazione si genererebbe un numero di riga maggiore di 9989.

Esempio

L'esempio che segue mostra l'elaborazione di un comando RENUM:

RENUM 20, 10 (poi battere EXECUTE)

Prima	Dopo
0010 INPUT A, B	0020 INPUT A, B
0011 Q = INT (A/B)	0030 Q = INT (A/B)
0020 IFQ > 0 THEN 30	0040 IFQ > 0 GOTO 0060
0025 GOTO 10	0050 GOTO 0020
0030 PRINT Q	0060 PRINT Q
0035 GOTO 10	0070 GOTO 0020
0040 END	0080 END

REWIND [indirizzo dell'unità]

IL COMANDO REWIND

Il comando REWIND permette di riavvolgere l'unità nastro specificata. La sintassi del comando REWIND è la seguente:

REWIND [indirizzo dell'unità]

dove:

indirizzo dell'unità è l'indirizzo dell'unità nastro che deve essere riavvolta (E80 per l'unità nastro incorporata oppure E40 per l'unità nastro ausiliaria). In mancanza di essi si assume E80.

Esempio

Un esempio di comando REWIND è del tipo seguente:

REWIND E40

$\text{RUN} \left[\left\{ \begin{array}{l} \text{STEP} \\ \text{TRACE} [\text{PRINT}] \end{array} \right\} \right] [\text{P=D}] [\text{RD=n}]$

IL COMANDO RUN

Il comando RUN inizia l'elaborazione di un programma BASIC a cominciare dalla istruzione elaborabile fornita del numero di riga più piccolo. E' necessario che il programma sia già presente nell'area di lavoro e che l'area di lavoro sia stata definita per contenere un programma BASIC (vedi Comando LOAD per il caricamento di programmi da nastro e la definizione del tipo di area di lavoro).

I programmi BASIC Possono essere elaborati in tre maniere:

Modo normale. Tutte le istruzioni del programma vengono elaborate senza interruzione.

Modo "Passo-Passo". Il 5100 si ferma immediatamente prima dell'esecuzione di ciascun "passo" del programma (istruzione). Vengono visualizzati le parole STEP nonchè il numero della istruzione che deve essere successivamente elaborata. Per elaborare la successiva istruzione di programma o per cambiare il modo di elaborazione, bisogna eseguire un comando GO (vedi Comando GO). Il sistema "passo-passo" permette anche di visualizzare o modificare i valori delle variabili tra un passo e l'altro con lo scopo di correggere gli errori.

Modo "trace". Viene visualizzato e/o stampato il numero di riga di ciascuna istruzione elaborata, mentre l'istruzione stessa viene elaborata.

Sia il sistema "passo-passo" che quello con "trace" sono utilizzabili per la localizzazione di errori di programmazione. Il sistema "trace" fornisce una visione più rapida dei passi del programma durante l'elaborazione. Il sistema "passo-passo" tra l'altro, permette di esaminare il contenuto delle variabili tra un passo e l'altro del programma nonchè modificare i passi del programma od i dati.

La sintassi del comando RUN è come sotto esposta:

$$\text{RUN} \left[\left\{ \begin{array}{l} \text{STEP} \\ \text{TRACE} [\text{PRINT}] \end{array} \right\} \right] [\text{P=D}] [\text{RD=n}]$$

dove:

STEP specifica l'esecuzione "passo-passo".

TRACE specifica una elaborazione continuativa delle istruzioni, ma viene visualizzato e/o stampato il numero di riga di ogni istruzione appena elaborata. Tener presente che per il modo "normale" non bisogna specificare STEP o TRACE.

PRINT indica che i messaggi di "trace" vengono stampati e visualizzati (è possibile unicamente con TRACE).

P=D specifica che le uscite dei programmi nei quali si fa riferimento alla stampatrice, vengono invece dirette allo schermo video.

RD=n permette di specificare il numero delle cifre (n) alla destra del punto decimale a cui arrotondare l'emissione stampata. Il valore n può andare da 1 a 13 ed è inizialmente, cioè all'accensione dell'elaboratore, impostato a 6 cifre.

Note Relative a RUN

- Il comando RUN sarà scartato dal 5100 se il tipo di file presente nell'area di lavoro è DATA.
- Il comando RUN inizializza tutte le variabili e gli elementi di insiemi a zero oppure a spazi vuoti, a seconda si tratti di variabili numeriche o carattere.
- Allorchè viene identificato P=D tutte le istruzioni PRINT FLP, PRINT USING FLP, MAT PRINT FLP, e MAT PRINT USING FLP saranno rispettivamente interpretate come istruzioni PRINT, PRINT USING, MAT PRINT, e MAT PRINT USING durante quella fase.

Esempio

Alcuni campioni di comandi RUN sono i seguenti:

1. Per iniziare una normale elaborazione di un programma:

RUN (poi battere EXECUTE)
2. Per iniziare una elaborazione con "trace" di un programma e stampare i passi elaborati:

RUN TRACE, PRINT (poi battere EXECUTE)

$$\text{SAVE numero di file} \left[\left\{ \begin{array}{l} \text{KEYx} \\ \text{KEYS} \\ \text{SOURCE} \end{array} \right\} \right] [\text{,indirizzo dell'unità}] [\text{'file ID'}]$$

IL COMANDO SAVE

Il comando SAVE permette di memorizzare il contenuto dell'area di lavoro in un file specificato. Dopo il completamento del comando SAVE lo schermo mostrerà il messaggio READY e la seguente informazione riguardo al file:

- Numero di aree di memoria in blocchi di 1024 byte contigui che sono stati assegnati al file.
- Numero di aree di memoria in blocchi di 1024 byte contigui non utilizzati rispetto a quelli messi a disposizione del comando MARK.

Se il file viene riempito dai dati, prima che tutta l'area di lavoro sia memorizzata, verrà visualizzato un messaggio di errore (vedi Appendice B) e il file verrà reimpostato come file non utilizzabile. Pertanto per memorizzare l'area di lavoro, sarà necessario reimmettere un altro comando SAVE specificando però un'altra file che possieda una adeguata quantità di spazio utile.

La sintassi del comando SAVE è la seguente:

$$\text{SAVE numero di file} \left[\left\{ \begin{array}{l} \text{KEYx} \\ \text{KEYS} \\ \text{SOURCE} \end{array} \right\} \right] [\text{,indirizzo dell'unità}] [\text{'file ID'}]$$

dove:

numero di file è un numero intero che specifica il file in cui si deve memorizzare il contenuto dell'area di lavoro.

KEYx specifica che si deve memorizzare una funzione-tasto (x=da 0 a 9) (vedi Funzioni-tasto).

KEYS specifica che si deve memorizzare il contenuto di tutte le Funzioni-tasto definite.

SOURCE specifica che si deve memorizzare il contenuto dell'area di lavoro nello stesso formato di caratteri in cui è stato immesso dall'utente.

indirizzo dell'unità è l'indirizzo dell'unità nastro in cui risiederà il file specificato. Si assume, se non specificato, che l'indirizzo sia E80.

'file ID' è una serie di caratteri, costituita da non più di 17 caratteri chiusi tra apici che si possono usare per identificare il file che contiene i dati memorizzati. Può essere inserita come riferimento in istruzioni OPEN (vedi OPEN, Capitolo 4).

Note Relative a SAVE

- Se KEYx, KEYS, o SOURCE non vengono immessi nel comando SAVE, i dati presenti nell'area di lavoro verranno memorizzati come un programma BASIC oppure come file DATA, in funzione di come è stata precedentemente definita l'area di lavoro.
- Se si interrompe l'elaborazione del comando SAVE battendo il tasto ATTN, l'operazione si conclude. Se l'interruzione avviene prima che il file sia stato cambiato, il file resta inalterato. Se l'interruzione avviene dopo che il file sia stato cambiato, il file viene reinizializzato come file non utilizzabile.
- Gli IMF non vengono memorizzati col comando SAVE. Bisogna applicare gli IMF con il comando PATCH prima di caricare l'area di lavoro con un qualunque programma memorizzato su nastro che necessiti degli IMF.

Esempio

Un campione di comando SAVE è il seguente:

SAVE4, KEYS

In questo esempio, il contenuto della Funzione-tasto 8 è memorizzato nel file 4, che si trova nella cartuccia a nastro nell'unità nastro incorporata.

UTIL	{	MODE COM	}
	{	[PRINT,] [DIR [intero]] [,indirizzo dell'unità]	}

IL COMANDO UTIL

Si può far uso del comando UTIL per fare la lista del catalogo di una cartuccia a nastro oppure per cambiare lo stato di elaborazione del 5100 da BASIC a telecomunicazioni. Per fare la lista del catalogo di una cartuccia a nastro è necessario avere nell'area di lavoro un po' più di spazio. Se non vi è disponibilità di un sufficiente spazio di lavoro durante l'elaborazione di un comando UTIL, verrà visualizzato un messaggio di errore (Appendice B). Bisogna allora azzerare l'area di lavoro (mediante, ad esempio, un comando LOAD0) e reimmettere un comando UTIL per continuare. L'informazione relativa a ciascun file che si trova nella cartuccia a nastro viene stampata o visualizzata mediante una riga per ciascun file. Se si indica di stampare la lista sulla stampatrice, vengono stampate senza interruzione tutte le informazioni relative ai files. Si può interrompere la stesura della lista, sia che avvenga sullo schermo che sulla stampatrice, battendo il tasto ATTN, che concluderà il comando, oppure battendo HOLD e poi ribattendolo nuovamente per continuare.

Viene stampata o visualizzata, relativamente ad ogni file, la seguente informazione:

- Il numero di file.
- L'identificazione che è stata assegnata al file, se esiste (vedi il Comando SAVE).
- Tipo di File (vedi Tipi di File).
- Numero di aree di memoria di 1024 bytes contigui assegnati al file.
- Numero di aree di memoria di 1024 bytes contigui non utilizzati che vi sono nel file.
- Un numero (0-9) che indica il numero di aree di 512 bytes difettosi presenti nella file, oppure un asterisco, se il numero delle aree è maggiore di 9. Questo valore può indicare il momento in cui sarebbe opportuno trasferire un file in un'altro file o un'altra cartuccia per evitare danneggiamenti ai dati causati da aree difettose presenti sul nastro.
- Il primo e l'ultimo numero di istruzione di files BASIC SOURCE.
- I numeri di Funzioni-tasto memorizzati nel file.

Tipi di File

I tipi di files validi di cui viene fatta la lista durante il comando UTIL sono:

Tipo	File
0	File marcato o non usato
1	File di interscambio di dati
2	File di interscambio generale
3	File "origine" BASIC
4	File di area di lavoro BASIC
5	File K AYS BASIC
6	File di trasferimento APL
7	File SAVE APL
8	Dati interni APL
16	Rappezzatura/correzione di nastro/copiatura di nastro
17	Diagnostica
18	IMF
72	Stampa del contenuto della memoria del nastro.

La sintassi del comando UTIL è la seguente:

$$\text{UTIL} \left\{ \begin{array}{l} \text{MODE COM} \\ [\text{PRINT,}] [\text{DIR [intero]}] [, \text{indirizzo unità}] \end{array} \right\}$$

dove:

MODE COM specifica si vuole attivare lo stato di tale comunicazione. Questa deve essere l'unica immissione relativa a un comando UTIL che voglia attivare lo stato di tale comunicazione.

PRINT indica che venga stampata la lista. Se non viene specificata la stampatrice, l'emissione viene solo visualizzata.

DIR, intero, specifica di effettuare la lista del catalogo. Il numero intero, che serve ad identificare il numero del file da cui iniziare la stesura della lista, è facoltativo. Se non è stato immesso il numero intero, la stesura della lista inizierà a partire dalla corrente posizione in cui si trova il nastro.

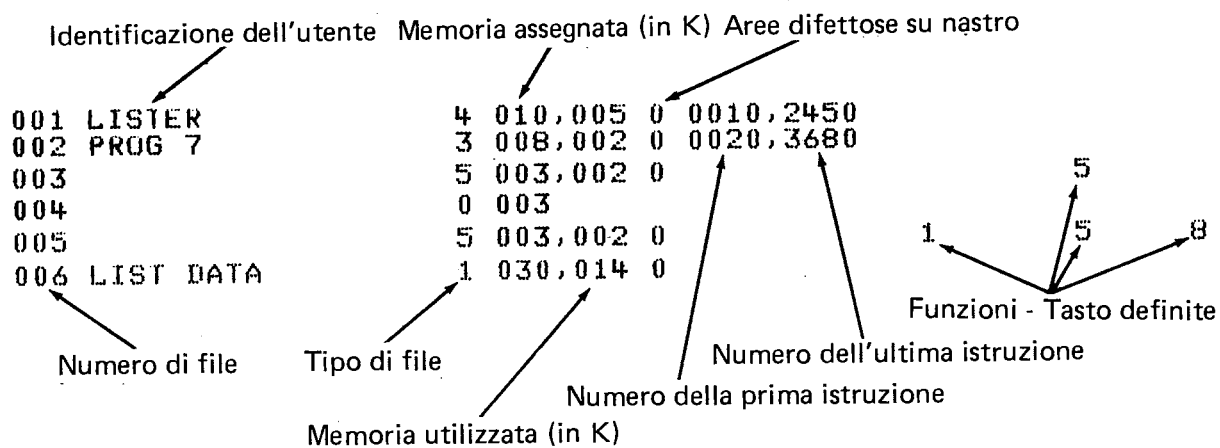
indirizzo dell'unità specifica l'indirizzo dell'unità nastro da cui prelevare la stesura della lista. In sua assenza si assume l'indirizzo E80.

Esempio

L'esempio seguente mostra un comando UTIL il quale indica che si deve stampare un indirizzario, presente nell'unità nastro incorporata, a partire dal file numero 1.

UTIL PRINT, DIR1

L'emissione stampata risultante viene mostrata nel seguente esempio:



Capitolo 3. Dati Costanti e Variabili

ASSIEME DEI CARATTERI BASIC

L'insieme dei caratteri BASIC viene utilizzato per rappresentare i dati aritmetici od i caratteri immessi tramite tastiera come costanti e variabili di dati.

I caratteri forniti di un significato sintatticamente valido nel linguaggio BASIC, cadono in tre categorie: caratteri alfabetici, numerici e speciali.

Caratteri Alfabetici

I caratteri alfabetici del linguaggio BASIC sono le lettere maiuscole dell'alfabeto inglese (A-Z) nonché i seguenti tre caratteri detti "riempitivi" dell'alfabeto:

- @ (il segno commerciale at)
- # (il numero o segno di libbra)
- \$ (il simbolo del dollaro)

Caratteri Numerici

In linguaggio BASIC i caratteri numerici sono le cifre che vanno da 0 a 9.

Caratteri speciali

Nell'linguaggio BASIC esistono 22 caratteri speciali:

Carattere	Nome
	Spazio vuoto
=	Segno di uguale o simbolo di assegnazione
+	Segno più
-	Segno meno
*	Asterisco o simbolo di moltiplicazione
/	Barra o simbolo di divisione
↑	Freccia verso l'alto o simbolo per elevazione a potenza
(	Parentesi sinistra
)	Parentesi destra
,	Virgola
.	Punto o punto decimale
;	Apice
:	Punto e virgola
;	Due punti
&	"E" commerciale
?	Punto interrogativo
>	Simbolo di maggiore di
<	Simbolo di minore di
≠	Simbolo di diverso da
≤	Simbolo di minore od uguale a
≥	Simbolo di maggiore od uguale a
	Segno OR o barra verticale

L'uso degli Spazi Vuoti

Si può liberamente far uso degli spazi vuoti, in un programma, per migliorare la possibilità di lettura. Gli spazi vuoti non possiedono un significato di sintassi, fatta eccezione di quando vengono usati in costanti carattere ed in specifiche maschere di stampa (vedi PRINT USING e IMAGE, Capitolo 4) i quali esprimono il formato dei dati che si devono stampare o visualizzare.

Tratto di sottolineatura

Tenere presente che il tratto di sottolineatura (simbolo superiore del tasto F) è, nel 5100 un valido carattere che può essere sovrapposto.

DATI ARITMETICI

I dati aritmetici sono quei dati dotati di un valore numerico. Tutti i numeri in linguaggio BASIC sono espressi a base 10; cioè vengono trattati come numeri decimali.

Grandezza

La grandezza di un numero è rappresentata dal suo valore assoluto. L'intervallo della grandezza dei numeri permessi nei programmi BASIC del 5100 va da numeri maggiori od uguali a $.1E-99$ a numeri minori od uguali a $1E+99$.

Precisione

In linguaggio BASIC la precisione di un numero intero o di un numero in virgola fissa è costituito dal numero massimo di cifre significative che può contenere. La precisione di un numero in virgola mobile è costituita dal numero di cifre significative che si trovano a sinistra di E (vedi Formato in virgola mobile). La precisione dei numeri nel 5100 è di 13 cifre.

Formato dei dati aritmetici

I dati aritmetici possono essere immessi, visualizzati o stampati in uno qualsiasi dei tre formati: interi, a virgola fissa od a virgola mobile. L'appropriato formato per un determinato numero dipende dalla sua grandezza e dal livello di precisione aritmetico che si richiede.

I numeri in qualsiasi formato si trovino possono essere positivi o negativi. I numeri negativi debbono essere preceduti dal segno meno. Quando non è specificato alcun segno, un numero viene trattato come numero positivo; in sostanza i segni positivi sono facoltativi.

Formato intero

I numeri espressi in un formato intero (formato I) vengono scritti come un numero di cifre facoltativamente preceduto da un segno. Esempi di numeri in formato intero sono:

0
+ 2
- 23
2683

Formato in virgola fissa

I numeri espressi nel formato in virgola fissa (formato F) sono scritti come numeri costituiti da un numero facoltativo di cifre, precedute da un segno (facoltativo) e seguite da un punto decimale. Il punto decimale può a sua volta esser seguito da un certo numero di cifre. Queste cifre sono necessarie nel caso in cui non vi siano cifre che precedano il punto decimale. Esempio di numeri nel formato di virgola fissa sono:

33
33.00
-.3
+3.56

Formato in virgola mobile

Numeri espressi nel formato invirgola mobile (formato E) vengono scritti con un segno facoltativo, seguito da un numero intero od un numero in virgola fissa, seguito dalla lettera E. La lettera E deve essere seguita dalla caratteristica (esponente) costituita da una o due cifre e facoltativamente preceduta dal segno.

Il valore di un numero in virgola mobile è uguale al numero alla sinistra di E moltiplicato per 10 elevato alla potenza rappresentata dal numero che si trova a destra di E. Questa notazione corrisponde alla notazione scientifica standard in cui i numeri vengono espressi come una potenza di 10. Tener presente viceversa che mentre il numero 10^7 lo si può accettare in una notazione scientifica, il numero E7 non è valido numero invirgola mobile. Il valore 10^7 nel formato BASIC in virgola mobile deve essere espresso come 1E7. Pertanto, il formato BASIC in virgola mobile ha bisogno di un numero posto alla sinistra di E.

Esempi di numeri in un formato in virgola mobile sono:

Numeri in virgola mobile	Equivalenti numeri decimali
.25E-4	.000025
+1.0E+5	100000
5E-7	.0000005
-15.33E6	-15330000

Scelta di un Formato Aritmetico

E' possibile immettere valori aritmetici mediante tastiera nei formati più convenienti per le nostre applicazioni. Il numero un milione, ad esempio, lo si può immettere in una qualsiasi delle seguenti maniere:

1000000
1000000.00
1E+6

La grandezza dei numeri di valori aritmetici è limitata unicamente dalla grandezza ($\geq .1E-99$ e $< 1E+99$). Tener presente, viceversa, che la materiale lunghezza dei valori che si immettono non ha limiti, benchè immissioni che superino 13 cifre verranno troncate a destra a causa della precisione del 5100, che è di 13 cifre. Pertanto numeri molto piccoli oppure molto grandi possono essere immessi nel formato E. Ad esempio si può immettere:

1.4E12

o l'equivalente valore nel formato I o F (14 seguito da 11 zeri).

E' possibile far uso di opportune istruzioni BASIC per controllare il formato dei valori aritmetici visualizzati o stampati (vedi PRINT USING e IMAGE, Capitolo 4). Pertanto, la forma dei valori originariamente immessi non influenzerà il formato dell'emissione.

Costanti Aritmetiche

Una costante aritmetica è sia un numero intero, sia un numero in virgola fissa o mobile il cui valore non venga mai alterato durante l'elaborazione del programma. Così, il numero intero 1 nella seguente istruzione è una costante:

$$X = X + 1$$

Costanti Interne

Una costante interna è una costante aritmetica con un valore predefinito. Diversamente dalle normali costanti aritmetiche, le costanti interne possiedono come riferimento dei nomi; sebbene siano come le normali costanti aritmetiche, il loro valore non viene mai alterato durante l'elaborazione del programma. Le sei costanti interne sono:

Costante	Nome	Valore
Pi (π)	&PI	3.141592653590
Natural log	&E	2.718281828459
Square root of 2	&SQR2	1.414213562373
Centimeters per inch	&INCM	2.540000000000
Kilograms per pound	&LBKG	0.453592370000
Liters per gallon	&GALI	3.785411784000

I nomi delle costanti interne possono unicamente essere usati come parti di espressioni aritmetiche, e possono essere preceduti da un segno più o meno, ad esempio:

$2 * \&PI$ (poi battere EXECUTE)

Il risultato è 6.283185.

Variabili Aritmetiche

Una variabile è un dato fornito di un nome ed il cui valore è soggetto al cambiamento durante l'elaborazione del programma. Le variabili aritmetiche prendono il nome mediante una singola lettera di tutto l'alfabeto (A-Z, $\llcorner$, #, e \$) oppure mediante una singola lettera di tutto l'alfabeto seguita da una sola cifra (0-9). Esempi di nomi di variabili aritmetiche sono A, A5, #1, e #7.

Quando viene elaborato un programma, il valore iniziale delle variabili aritmetiche è impostato a zero. L'unica eccezione è che le variabili assegnate in una istruzione USE non vengono inizializzate (vedi USE, Capitolo 4) per un programma al quale sono concatenate (vedi CHAIN, Capitolo 4).

DATI CARATTERE

Nel linguaggio BASIC i dati carattere sono dati sprovvisti di un valore numerico. Simili ai dati aritmetici; i dati carattere possono essere nella forma di costanti o variabili.

Costanti Carattere *Stringhe*

Una costante carattere è costituita da una serie di caratteri chiusi in una coppia di apici. Una lettera qualsiasi, una cifra od un carattere speciale può essere incluso in una costante carattere. Un apostrofo, viceversa, deve essere indicato mediante l'uso di due apici. Ad esempio, 'IT'S' rappresenta IT'S. Le seguenti sono tutte valide costanti carattere:

```
'YES'  
'THE SQUARE OF X IS'  
'12345'  
'A B'
```

La lunghezza di una costante carattere, allorchè viene visualizzata o stampata, è rappresentata dal numero di caratteri che contiene, inclusi i vuoti, ma escludendo gli apici che li racchiudono. Ciascuna coppia di apici usati per rappresentare un apostrofo viene conteggiata come un solo carattere. Il numero massimo di caratteri in una singola costante carattere è limitato unicamente dal numero massimo di caratteri che è possibile inserire in una riga di immissione che è di 64 (incluse le virgolette).

Variabili Carattere

Una variabile carattere è un dato, definito da un nome, il cui valore può essere soggetto a cambiare durante l'elaborazione del programma. Variabili carattere vengono definite mediante una lettera singola di tutto l'alfabeto (A-Z, @, #, e \$), seguita dal simbolo di dollaro (\$). Esempi di variabili carattere sono A\$, #\$, e \$\$.

Quando un programma viene elaborato, il valore iniziale della variabile carattere viene impostato a 18 caratteri di spazi vuoti. L'unica eccezione è che le variabili assegnate alla comune area di memoria non siano inizializzate (vedi USE, Capitolo 4) per un programma con cui sono concatenate (vedi CHAIN, Capitolo 4). Queste variabili vengono inizializzate a spazi vuoti da un comando LOAD o RUN (vedi Capitolo 2).

Quando le costanti carattere vengono assegnate alle variabili carattere, i valori delle costanti vengono regolati ad una lunghezza di 18 caratteri. Costanti più lunghe vengono troncate a destra mentre costanti più corte vengono allineate a sinistra e riempite alla destra con degli spazi vuoti.

INSIEMI

Un insieme è costituito da una serie di dati (elementi) cui ci si riferisce mediante un singolo nome. Possono essere raggruppati tra loro, per formare un insieme, solamente elementi di dati dello stesso tipo (numerici o caratteri).

Gli insiemi possono essere sia mono che bidimensionali. Un insieme monodimensionale lo si può pensare come una riga di dati in successione l'uno dopo l'altro. Un insieme bidimensionale lo si può pensare come una matrice di righe e colonne. La Figura 6 mostra una rappresentazione schematizzata di ambedue i tipi di insiemi:

Insieme monodimensionale chiamato A			
A(1)	A(2)	A(3)	A(4)
Insieme bidimensionale chiamato B			
B(1,1)		B(1,2)	B(1,3)
B(2,1)		B(2,2)	B(2,3)
B(3,1)		B(3,2)	B(3,3)
B(4,1)		B(4,2)	B(4,3)

Figura 6. Rappresentazione Schematizzata di insiemi mono e bidimensionali

Ciascun elemento lo si riferisce all'insieme mediante il nome dell'insieme seguito da un indice posto in parentesi il quale sta ad indicare la posizione che l'elemento occupa nell'insieme. La forma generale che si usa per riferirsi ad un elemento di un insieme è:

nome dell'insieme (righe, colonne)

in cui il nome dell'insieme è il nome di tutto l'insieme mentre righe e colonne sono espressioni aritmetiche positive qualsiasi i cui valori interi troncati sono maggiori di zero e minori od uguali alla corrispondente dimensione dell'insieme.

L'espressione in un indice che si riferisca ad un elemento di un insieme monodimensionale fornisce la posizione dell'elemento nella riga, procedendo nel conteggio da sinistra verso destra. In questo modo, al terzo elemento di un insieme monodimensionale di nome A, ci si può riferire mediante il simbolo A(3), come in questo esempio:

$$A(3) = 25$$

La prima espressione in un indice che fa riferimento ad un elemento di un insieme bidimensionale fornisce il numero della riga che contiene l'elemento. Le righe sono numerate dall'alto verso il basso. La seconda espressione che si trova in un indice fornisce il numero delle colonne. Le colonne sono numerate da sinistra verso destra. Così, al secondo elemento che si trova nella quarta riga di un insieme di nome B si può fare riferimento con il simbolo B(4,2), come nel seguente esempio:

$$B(4,2) = 1.53E6$$

Le dimensioni di un insieme ed il numero di elementi in ciascuna dimensione vengono stabiliti quando viene dichiarato l'insieme.

Dichiarazione degli Insiemi

Gli insiemi possono essere dichiarati sia esplicitamente, mediante l'uso delle istruzioni USE o DIM, oppure implicitamente mediante un riferimento ad un elemento di un insieme che non sia stato esplicitamente dichiarato.

Quando un insieme viene dichiarato esplicitamente, vengono specificate nelle istruzioni USE o DIM le dimensioni ed il massimo numero di dati che possono essere contenuti in ciascun dimensione.

Quando un insieme viene dichiarato implicitamente, mediante un riferimento ad uno dei suoi elementi, nel caso in cui il suo nome non sia comparso in una precedente istruzione USE o DIM, esso avrà un numero di dimensioni come specificato al momento in cui la si usa e ciascuna dimensione conterrà dieci elementi. Per esempio, nel caso in cui non esista per un insieme di nome A una precedente istruzione esplicita USE o DIM, l'istruzione:

$$A(3) = 50$$

stabilirà un insieme monodimensionale contenente 10 elementi, il terzo dei quali avrà un valore intero di 50. I restanti elementi saranno inizializzati a zero.

Similmente, nel caso in cui non esista per un insieme di nome B una precedente istruzione esplicita USE o DIM, l'istruzione:

$$B(5,6) = 6.913$$

stabilirà un insieme bidimensionale contenente 10 righe e 10 colonne (100 elementi) con l'elemento della sesta colonna e della quinta riga uguale a 6.913.

Insiemi con dimensioni che contengano più di 10 elementi non possono essere dichiarati implicitamente. Così, senza una appropriata e precedente istruzione USE o DIM, le seguenti osservazioni causeranno entrambe delle condizioni di errore:

$$A(15) = 22.4$$

$$B(3,20) = 66.6$$

Dopo che un insieme è stato dichiarato, esplicitamente od implicitamente, non lo si può più esplicitamente dimensionare con una istruzione DIM in un qualsiasi successivo punto del programma.

Ridimensionamento degli Insiemi

Insiemi numerici e di caratteri possono però essere ridimensionati tenendo presenti le seguenti regole:

- Sia gli insiemi monodimensionali che quelli bidimensionali possono essere ridimensionati.
- Il numero totale degli elementi in un insieme dopo il ridimensionamento non deve superare il numero originariamente specificato quando fu dichiarato l'insieme.
- E' possibile cambiare il numero delle dimensioni.
- Il valore massimo per una dimensione è 255.
- Un insieme può essere ridimensionato in una istruzione di assegnazione MAT, in una MAT READ, in una MAT INPUT od in una istruzione MAT GET.
- Le nuove dimensioni per un insieme possono essere specificate sia con una costante che con una espressione.

Insiemi Aritmetici

Un insieme aritmetico contiene unicamente dati numerici e può avere una, oppure due dimensioni.

Il nome degli insiemi aritmetici viene assegnato mediante una singola lettera di tutto l'alfabeto (A-Z, @, #, e \$). Così, la lettera A può essere usata per dare il nome ad una variabile aritmetica od ad un insieme aritmetico, od ambedue, mentre il simbolo A2 può unicamente essere usato per dare il nome ad una variabile aritmetica. Ad esempio:

$$\begin{aligned}A &= 6 \\ A(1) &= 9\end{aligned}$$

in cui la variabile $A = 6$ (scalare) e $A(1) = 9$ (elemento di un vettore aritmetico). Tutti gli elementi di un insieme aritmetico vengono inizialmente impostati a zero quando il programma viene elaborato (eccettuati quelli assegnati alla comune area di memoria per essere usati in un programma con cui sono concatenati; vedi USE e CHAIN, Capitolo 4).

Prima di essere usati in una qualsiasi delle istruzioni per l'elaborazione delle matrici, un insieme aritmetico deve essere stato precedentemente dimensionato, esplicitamente oppure implicitamente. Gli insiemi aritmetici possono essere ridimensionati come precedentemente descritto.

Insiemi di Caratteri

Un insieme di caratteri contiene unicamente dati carattere e può avere una o due dimensioni.

Insiemi di caratteri, come le variabili di caratteri uniche, prendono il nome da una singola lettera di tutto l'alfabeto, seguita dal simbolo di dollaro (\$). Pertanto il nome D\$ lo si può riferire sia ad una variabile unica di carattere sia ad un insieme di caratteri, sia ad ambedue. Il nome D\$ (2,4) fa riferimento al quarto elemento della seconda riga di un insieme di caratteri bidimensionale (matrice). Ad esempio:

```
D$ = 'JONES'
D$ (2,4) = 'SMITH'
```

Insiemi di caratteri possono essere usati in immissione, emissione e come istruzioni uniche di assegnazione di matrici (quando non viene elaborata alcuna operazione aritmetica) e possono essere ridimensionati come precedentemente spiegato.

Sommario di Nominativi Convenzionali

La Figura 7 mostra un riassunto dei nominativi convenzionali precedentemente spiegati per le variabili e gli insiemi. Il simbolo ext presente in Figura 7 sta per un'lettera di tutto l'alfabeto (A-Z @, #, e \$). L'informazione in parentesi è facoltativa.

Tipo di Dato	Nome	Esempio
Variabili aritmetiche	ext [cifre]	A, \$5
Insiemi aritmetici	ext	A, \$, #
Variabili carattere	ext \$	A\$, \$\$, @\$
Insiemi di carattere	ext \$	A\$, \$\$, @\$

Figura 7. Conversioni per la denominazione di variabili e insiemi.

FUNZIONI DEL SISTEMA

Il linguaggio BASIC dell'IBM 5100 include le funzioni del sistema che elaborano un certo numero di operazioni matematiche comunemente usate. Inoltre è possibile definire e fornire di nome alcune nostre funzioni mediante l'uso della istruzione DEF (vedi DEF, FNEND, RETURN, Capitolo 4).

Le funzioni del sistema mostrate in Figura 8 possono essere usate in ogni punto di una espressione BASIC in cui si può far uso di riferimenti a costanti, variabili od elementi di insiemi aritmetici, così come mostrato nei seguenti esempi:

```
A = SIN(&PI) + 1
B = SQR (X+ 3)
R1 = RND
```

Ciascuna delle funzioni del sistema, eccettuata STR, possiede un argomento singolo (facoltativo con RND), il quale può essere una qualsiasi espressione aritmetica (spiegata più oltre in questo capitolo) e fornisce un risultato singolo. Un errato argomento produce un errore. L'argomento per la funzione DET deve essere un riferimento ad una matrice aritmetica quadrata. La precisione minima per queste funzioni del sistema è di 10 cifre. In figura 8 è riportata la lista delle funzioni del sistema, incluse nel 5100.

Nome della Funzione	Descrizione
ABS(x)	Valore assoluto di x
ACS(x)	Arco coseno (in radianti) di x
ASN(x)	Arco seno (in radianti) di x
ATN(x)	Arco tangente (in radianti) di x
COS(x)	Coseno di x radianti
CSC(x)	Cosecante di x radianti
DEG(x)	Numero di gradi di x radianti
DET(x)	⁽¹⁾ Determinante di un insieme aritmetico
EXP(x)	Esponenziale di x e di x
HCS(x)	Coseno iperbolico di x
HSN(x)	Seno iperbolico di x
HTN(x)	Tangente iperbolica di x
INT(x)	Parte intera di x
LGT(x)	Logaritmo di x a base 10
LOG(x)	Logaritmo di x in base e
LTW(x)	Logaritmo di x in base 2
RAD(x)	Numero di radianti in x gradi
RND [(x)]	⁽²⁾ Numero a caso compreso tra 0 e 1
SEC(x)	Secante di x radianti
SGN(x)	Segno di x (-1, 0, o + 1)
SIN(x)	Segno di x radianti
SQR(x)	Radice quadrata di x
STR(X \$, y, z)	Sottostringa della variabile carattere X\$, che inizia dal y ^{mo} carattere e che si estende per z caratteri
TAN(X)	Tangente di x radianti

⁽¹⁾ La massima grandezza di una matrice è 50x50. Un risultato minore di 1E-20 è assunto uguale a zero.

⁽²⁾ Tener presente che il numero a caso di una costante fornirà sempre lo stesso risultato, ma può essere usato per iniziare un sistema di numeri diverso. RND senza un argomento fornisce un numero a caso prelevato dallo stesso sistema di numeri.

Figura 8. Funzioni del sistema

ESPRESSIONI

In linguaggio BASIC una espressione è una rappresentazione qualsiasi di un valore aritmetico o carattere. Le costanti, le variabili, gli insiemi, i riferimenti ad elementi di insiemi nonché riferimenti a funzioni, sono considerati tutte espressioni. Le espressioni possono inoltre essere formate combinando una qualsiasi di queste rappresentazioni di valori con simboli chiamati operatori.

Un operatore specifica o la relazione esistente tra dati, oppure l'operazione aritmetica che deve essere elaborata su questi; specifica inoltre se sono positivi o negativi. Ad esempio, i simboli $>$, $*$, e $+$ sono operatori i quali vogliono dire rispettivamente maggiore di, moltiplicazione e addizione (oppure valore positivo).

Una classe speciale di espressioni, dette espressioni logiche, viene usata con l'istruzione IF per sapere se una determinata relazione tra due valori è vera o falsa.

Espressioni che si riferiscono ad interi insiemi, invece che ad elementi individuali di un insieme, sono dette espressioni di insiemi. Una espressione che non contiene un riferimento ad un intero insieme viene chiamata espressione scalare.

Espressioni Aritmetiche ed Operatori

Una espressione aritmetica può essere una variabile aritmetica, un elemento di insieme, una costante od un riferimento ad una funzione; oppure ancora può essere una serie di elementi separati tra loro da operatori e parentesi. Alcuni esempi di espressioni aritmetiche sono:

A1
X3/(-6)
X+Y + Z
SIN(R)
-6.4
 $-(X-X\uparrow 2/2 + X)$

Il valore di una espressione aritmetica lo si ottiene svolgendo le operazioni implicite mediante i dati specificati.

I cinque operatori aritmetici sono:

Simbolo	Significato
$\uparrow$ oppure **	Elevazione a potenza
*	Moltiplicazione
/	Divisione
+	Addizione
-	Sottrazione

Tener presente che il sistema memorizza comunque il simbolo ** come $\uparrow$.

Gli operatori positivo/negativo sono:

- + Positivo
- Negativo

Le regole speciali per gli operatori aritmetici e per le azioni che ne risultano sono come segue:

Elevazione a potenza: L'espressione $A \uparrow B$ viene definita come la variabile A elevata alla potenza B.

1. Se $A = B = 0$, si avrà un errore.
2. Se $A = 0$ e $B < 1$, si avrà un errore.
3. Se $A < 0$ e B non è un numero intero, si avrà un errore a causa dell'elevazione a potenza di un numero negativo da parte di una potenza frazionaria.
4. Se $A = 0$ e $B = 0$, $A \uparrow B$ è posto uguale a 1.
5. Se $A = 0$ e $B > 0$, $A \uparrow B$ è posto uguale a 0.

Moltiplicazione e Addizione: $A * B$ e $A + B$, moltiplicazione e addizione rispettivamente, sono ambedue commutabili; in altre parole, $A * B = B * A$ e $A + B = B + A$. Tuttavia la moltiplicazione e l'addizione non sono sempre associative, in quanto sono di ordine inferiore nell'arrotondare gli errori; ad esempio, $A * (B * C)$ non fornisce necessariamente lo stesso risultato di $(A * B) * C$.

Divisione: A / B viene definito come A diviso B. Se $B = 0$, si avrà un errore (Overflow).

Sottrazione: $A - B$ viene definito come A meno B. Non esistono condizioni speciali.

Operatori Positivo/Negativo: I segni + e — possono anche venir usati come operatori positivo/negativo, che possono essere usati unicamente in due situazioni:

1. Dopo una parentesi sinistra e prima di una espressione aritmetica.
2. Come primo carattere di una intera espressione aritmetica che non sia preceduta da un operatore.

Per esempio:

$-A + (-B)$ e $B - (-2)$ sono giuste.
 $A + -B$ oppure $B - -2$ sono sbagliate.

Gerarchia Aritmetica

Le espressioni aritmetiche vengono elaborate in accordo con la gerarchia degli operatori coinvolti. Le operazioni che si trovano in parentesi vengono elaborate per prime. Le operazioni con un livello di priorità più alto vengono elaborate prima di quelle con un livello di priorità più basso. Le operazioni che stanno sullo stesso livello di priorità vengono elaborate da sinistra verso destra. La gerarchia degli operatori è dunque:

Operatore	Gerarchia
1) Racchiusa in parentesi	La più alta
2. $\uparrow$ ovvero $**$	
3. Positivo $+$ e Negativo $-$	
4. $*$ e $/$	
5. Addizione $+$ Sottrazione $-$	La più bassa

Una espressione viene elaborata dopo averla ridotta ai suoi componenti di sottoespressione. Una sottoespressione viene definita come un gruppo che può essere letto operando-operatore-operando, in cui operando è uno dei seguenti:

- Un semplice riferimento ad un dato (costante o variabile).
- Un riferimento di insieme con indice.
- Un riferimento di funzione.
- Una sottoespressione tra parentesi.

Ad iniziare con il primo operatore che deve essere elaborato in accordo con la gerarchia, gli operandi della sua sottoespressione vengono ridotti a semplici riferimenti di dati in un ordine da sinistra verso destra. Questo procedimento viene ripetuto tante volte quanto si desidera, in un ordine da sinistra verso destra oppure in un ordine di priorità decrescente, oppure ambedue, dei restanti operatori fino a che l'intera espressione sia stata ridotta ad un semplice riferimento al risultato valutato.

Gli esempi che seguono, illustrano la successione dei passaggi di lavorazione dell'elaborazione di quattro espressioni aritmetiche in accordo con le regole che sono state illustrate. In ciascuna espressione, alle variabili A, B e C sono stati assegnati rispettivamente i valori 4, 6 e 2.

Espressione	Valutazione e Risultato
$-A \uparrow 2 + B/C * 2.5$	$-4 \uparrow 2 + 6/2 * 2.5$ $- 16 + 6/2 * 2.5$ $-16 + 6/2 * 2.5$ $-16 + 3 * 2.5$ $-16 + 7.5$ -8.5
$(-A \uparrow 2) + B/C * 2.5$	$(-4 \uparrow 2) + 6/2 * 2.5$ $- 16 + 6/2 * 2.5$ $-16 + 6/2 * 2.5$ $-16 + 3 * 2.5$ $-16 + 7.5$ -8.5
$-A \uparrow (2 + B/C) * 2.5$	$-4 \uparrow (2 + 6/2) * 2.5$ $-4 \uparrow (2 + 3) * 2.5$ $-4 \uparrow 5 * 2.5$ $- 1024 * 2.5$ $-1024 * 2.5$ -2560
$-A \uparrow ((2 + B)/C) * 2.5$	$-4 \uparrow ((2 + 6)/2) * 2.5$ $-4 \uparrow (8/2) * 2.5$ $-4 \uparrow 4 * 2.5$ $- 256 * 2.5$ $-256 * 2.5$ -640

Espressioni di Caratteri

Una espressione di caratteri è una costante carattere, una variabile carattere, un singolo elemento di un insieme di caratteri od una funzione sottostringa. (STR). Gli unici operatori sempre associati con espressioni di caratteri sono gli operatori logici precedentemente descritti. Qui oltre sono riportati alcuni esempi di valide espressioni di caratteri:

$D\$(4)$
 $A\$$
 $STR(A\$,1,3)=B\$\$$

Espressioni Logiche

Una espressione logica serve per confrontare i valori di due espressioni aritmetiche o di due espressioni di carattere. Le espressioni che devono essere confrontate vengono valutate e poi confrontate in accordo con la definizione dell'operatore logico specificato. Per ciò concerne il risultato, l'espressione logica può risultare o soddisfatta (vera) o non soddisfatta (falsa). Le espressioni logiche possono comparire in un programma BASIC solamente come parte di una istruzione IF.

Gli operatori logici e le loro definizioni sono:

Operatore	Significato
=	Equal
≠ or <>	Not equal
≥ or >=	Greater than or equal
≤ or <=	Less than or equal
>	Greater than
<	Less than

Nota: Il sistema memorizza <> come ≠, <= come ≤ e >= come ≥. Nei confronti vengono usate tutte e 13 le cifre significative di ogni numero.

Il formato generale di una espressione logica è:

e_1 operatore-logico e_2

dove e_1 ed e_2 sono espressioni qualsiasi fuorchè insiemi o espressioni logiche, ed operatore-logico è uno qualsiasi degli operatori precedentemente descritti. Ambedue le espressioni e_1 ed e_2 debbono essere dello stesso tipo di dati (caratteri od aritmetici), ed in un singola espressione logica possono essere confrontate solamente due espressioni.

Quando i dati di caratteri compaiono in una espressione logica, vengono elaborati in accordo con la successione logica di comparazione (vedi Appence A) carattere per carattere, da sinistra verso destra. Pertanto le seguenti espressioni logiche saranno tutte soddisfatte:

'ABC' = 'ABC'
'ABLE' < 'BALL'
'123' > 'BALL'
'\$12' < '7'

Quando vengono confrontati operandi di caratteri di differenti lunghezze, l'operando più corto lo si considera come esteso alla sua destra mediante spazi vuoti fino alla lunghezza dell'operando più lungo. Così, nel terzo esempio precedente, i valori confrontati sono 123b e BALL, in cui b è un carattere vuoto.

Espressioni di Insieme

In matematica, una matrice è un gruppo di valori aritmetici disposti in un sistema di righe e di colonne, ed un vettore è costituito da una serie di valori aritmetici disposti su una singola riga. In linguaggio BASIC, viceversa, una matrice è un insieme mono o bidimensionale unicamente di dati numerici. Espressioni di insiemi vengono usate per svolgere operazioni sull'intera collezione degli elementi di insieme numerico o di carattere piuttosto che su ciascun elemento individualmente (operazioni scalari) (vedi Specifiche di assegnazione MAT, Capitolo 4).

Funzione Sottostringa

Una stringa di caratteri è una sequenza di caratteri in una espressione di caratteri. La funzione stringa (STR) permette di estrarre, combinare o sostituire determinati caratteri nella espressione. La funzione STR viene usata con l'istruzione LET per realizzare quanto segue:

- Estrazione di caratteri. per esempio nelle seguenti specifiche:

```
10 A$ = 'PRODUCTION CONTROL'  
20 STR(B$,1,10) = STR(A$,1,10)
```

la riga 20 estrae i primi 10 caratteri da A\$ e li assegna ai primi 10 caratteri di B\$. Nella istruzione 20, A\$ è la espressione dalla quale si devono estrarre i caratteri, 1 è la posizione del primo carattere che deve essere estratto, e 10 è il numero di caratteri che si devono estrarre.

- Combinazione di caratteri. Per esempio in queste istruzioni:

```
10 A$ = 'PRODUCTION'  
20 B$ = 'CONTROL'  
30 LET STR(A$,12,7) = B$
```

l'istruzione 30 colloca il contenuto di B\$ (CONTROL) assieme al contenuto di A\$ (PRODUCTION) dopo l'undicesimo carattere, che è uno spazio vuoto. Nella riga 30, A\$ è l'espressione in cui saranno inclusi i caratteri, 12 è la prima posizione che deve essere riempita dai caratteri addizionali, e 7 è il numero di caratteri che devono essere inclusi.

- Sostituzione di caratteri. Per esempio in queste istruzioni:

```
10 LET A$ = 'PART 789'  
20 LET B$ = 'PART 1234'  
30 LET STR(A$,6,4) = STR(B$,6,4)
```

la riga 30 sostituisce il 789 di A\$ con il 1234 di B\$. Di nuovo, i numeri 6 e 4 della riga 30 specificano rispettivamente il primo carattere che deve essere sostituito ed il numero di caratteri che devono essere sostituiti. Questi numeri non devono essere gli stessi da ambedue le parti del segno di uguale. Tener presente che la lunghezza dell'operando alla sinistra del segno di uguale dovrebbe essere più corta od uguale alla lunghezza dell'operando alla destra del segno di uguale.

Le istruzioni BASIC permettono di immettere dati, specificare come tali dati vadano manipolati e che ci si aspetta un risultato. Le istruzioni BASIC sono eseguibili o non eseguibili. Quelle eseguibili causano un'azione visibile del programma, quale l'assegnazione di un valore o la stampa. Quelle non eseguibili invece, descrivono solo una informazione necessaria al programma ed all'utente ma non creano azioni visibili.

Le istruzioni eseguibili e non eseguibili possono essere mischiate tra loro allorchè viene battuto un programma BASIC da tastiera. Il massimo numero di istruzioni permesse in un unico programma BASIC è limitato unicamente dalla grandezza della memoria di lettura e scrittura dell'IBM 5100, nonché dal tipo stesso delle istruzioni.

Ciascuna istruzione di un programma BASIC deve iniziare con un numero di riga. Il numero determina l'ordine di elaborazione delle istruzioni nel programma. Tutte esse vengono elaborate in ordine di numero senza riguardo all'ordine in cui sono state immesse e senza che la sequenza di elaborazione venga alterata da salti, iterazioni o sottoprogrammi.

L'intervallo ammesso per i numeri delle istruzioni dei programmi BASIC del 5100 va da 1 a 9999. Non è necessario immettere gli zeri prima del numero (0020, ad esempio) in quanto il 5100 gestisce i numeri di riga come numeri interi di quattro cifre ed inserisce gli zeri che precedono le cifre per i numeri di riga che hanno meno di quattro cifre significative.

Righe di istruzione

Una istruzione BASIC preceduta da un numero di istruzione è detta riga di istruzione. Esse vengono immesse tramite tastiera (una per ogni riga dello schermo) con un massimo di 64 caratteri. Le istruzioni non possono essere ripartite tra due righe dello schermo, Né vi può essere più di una istruzione in ciascuna riga dello schermo. Una tipica riga di istruzione è la seguente:

10 LET A = 2+2

Numero di Istruzione
Istruzione BASIC

In questo capitolo tutte le istruzioni BASIC vengono presentate in ordine alfabetico e con la stessa sintassi che è stata usata per i comandi BASIC, fatta eccezione per le istruzioni inerenti le operazioni sulle matrici. Queste istruzioni saranno discusse alla fine del capitolo sotto il titolo di Operazioni su Matrici.

ISTRUZIONI BASIC

Più oltre viene riportata la lista delle istruzioni usate nel linguaggio BASIC per il 5100 accludendo una breve descrizione per ciascuna di esse.

CHAIN — Termina un programma, indi carica ed inizia l'elaborazione di un altro programma.

CLOSE — Disattiva i files aperti.

DATA — Crea una tabella di dati interni di valori (forniti dall'esterno).

DEF — Definisce una funzione aritmetica che deve essere usata nel programma.

DIM — Specifica la grandezza e le dimensioni di un insieme.

END — Finisce un programma.

FNEND — Finisce una funzione aritmetica definita in una istruzione DEF.

FOR — Inizia una iterazione.

GET — Assegna dei valori a delle variabili prendendoli da un file.

GOSUB — Passa il controllo dell'esecuzione all'inizio di un sottoprogramma.

GOTO — Passa il controllo dell'esecuzione ad una determinata istruzione.

IF — Salta in funzione di determinate condizioni.

Image — Specifica in quale formato debbano essere stampati o visualizzati i dati.

INPUT — Assegna dei valori, da tastiera, a delle variabili durante l'elaborazione di un programma.

LET — Assegna dei valori a delle variabili.

MAT di Assegnazione — Assegna dei valori a tutti gli elementi di un insieme.

MAT GET — Assegna dei valori agli elementi di un insieme prelevandoli da un file.

MAT INPUT — Assegna dei valori tramite tastiera agli elementi di un insieme.

MAT PRINT (FLP) — Visualizza (o stampa) i valori di tutti gli elementi di un insieme.

MAT PRINT USING (FLP) — Visualizza (o stampa) i valori di tutti gli elementi di un insieme nel formato definito in una istruzione che da la maschera di stampa (Image)

MAT PUT — Scrive i valori di tutti gli elementi di un insieme in un file su nastro.

MAT READ — Assegna valori, prelevandoli da una tabella di dati interni (vedi DATA) agli elementi di un insieme.

NEXT — Fine di una iterazione (vedi FOR e NEXT).

OPEN — Attiva i files per immissioni od emissioni.

PAUSE — Interrompe l'elaborazione del programma.

PRINT (FLP) — Visualizza (o stampa) i valori delle variabili specificate, espressioni o costanti.

PRINT USING (FLP) — Visualizza (o stampa) i valori delle specificate variabili, espressioni o costanti in un formato definito in una istruzione maschera di stampa (Image).

PUT — Scrive i valori di variabili specificate in un file su nastro.

READ — Assegna valori, prelevandoli dalla tabella interna (vedi DATA), a variabili od ad elementi di insieme.

REM — Inserisce dei commenti o delle annotazioni in un programma.
RESET — Reimposta un file su nastro al suo inizio od alla sua fine.
RESTORE — Fa in modo che i valori che si trovano nella tabella di dati interni (vedi DATA) vengano riassegnati ad iniziare dal primo valore della tabella.
RETURN — Fine del corrente sottoprogramma.
STOP — Finisce un programma.
USE — Memorizza le variabili che devono essere usate nei successivi programmi.

CHAIN 'indirizzo dell'unità', espressione aritmetica

CHAIN

L'istruzione CHAIN elabora la seguente sequenza:

1. Finisce il programma attualmente in elaborazione.
2. Carica un nuovo programma.
3. Inizia l'elaborazione del nuovo programma.

La sintassi di CHAIN è la seguente:

CHAIN 'indirizzo dell'unità', espressione aritmetica

dove:

'indirizzo dell'unità' è l'indirizzo dell'unità che contiene il successivo programma che deve essere caricato ed elaborato. L'indirizzo dell'unità può essere 'E80' (unità nastro incorporata) o 'E40' (unità nastro ausiliaria).

espressione aritmetica è un'espressione unica o una costante

Prima dell'elaborazione dell'istruzione CHAIN, l'espressione viene valutata e troncata ad un numero intero. Questo numero intero identifica il file che contiene il nuovo programma nell'unità nastro specificata. Può essere usata una semplice costante. Tutti i files aperti nell'attuale programma vengono chiusi. Il programma che è nel file (determinato dalla espressione o dalla costante) dell'unità nastro specificata viene allora caricato ed elaborato ad iniziare dal più piccolo numero di istruzione.

Ad esempio:

0110 CHAIN 'E80', 14

In questa specifica il programma presente nel file numero 14 nell'unità 'E80' (unità nastro incorporata) viene caricato ed elaborato una volta terminato il corrente programma.

Nota relativa a CHAIN

Quando viene usata in congiunzione alla istruzione USE, la CHAIN permette di gestire i valori delle variabili da un programma al successivo (vedi istruzione USE).

CLOSE rifer. di file [,rifer. di file] . . .
--

CLOSE

L'istruzione CLOSE specifica il file che deve essere disattivato. Una CLOSE implicita viene automaticamente elaborata per ciascun file attivo alla fine dell'elaborazione del programma. La sintassi della CLOSE è la seguente:

CLOSE riferimento di file [, riferimento di file]...

dove:

riferimento di file va da FL0 a FL9 e rappresenta lo stesso file specificato nella istruzione OPEN. E' necessario solamente un riferimento al file.

Note relative a CLOSE

- Se un file viene utilizzato sia per operazioni di immissione che per operazioni di emissione durante l'elaborazione di un unico programma, il file deve essere dapprima chiuso e poi riaperto tra l'istruzione di immissione e quella di emissione.
- Se un file non viene chiuso (CLOSE o fine di programma), il file può diventare inutilizzabile.
- Se un file specificato in una CLOSE non è attivo allorchè viene elaborata l'istruzione CLOSE, questa viene ignorata.
- I riferimenti ai files debbono essere gli stessi di quelli già specificati per gli stessi files nella istruzione OPEN.

Esempio

Una istruzione CLOSE è del tipo:

0020 CLOSE FL2, FL9

DATA	$\left\{ \begin{array}{l} \text{costante arit.} \\ \text{costante carat.} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{costante arit.} \\ \text{costante carat.} \end{array} \right\} \right] \dots$
------	--

DATA

L'istruzione DATA è una istruzione non eseguibile mediante la quale è possibile creare una tabella di dati interni. Le costanti della tabella di dati vengono assegnate alle variabili ed agli elementi dell'insieme specificato nelle corrispondenti istruzioni READ o MAT READ.

La sintassi della istruzione DATA è del tipo:

$$\text{DATA} \left\{ \begin{array}{l} \text{costante aritmetica} \\ \text{costante carattere} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{costante aritmetica} \\ \text{costante carattere} \end{array} \right\} \right] \dots$$

dove:

costante è una costante aritmetica o di carattere. E' necessaria almeno una costante.

All'inizio dell'elaborazione del programma (prima che venga elaborata una qualsiasi istruzione elaborabile) viene costruita una tabella che contenga tutte le costanti che si trovano in tutte le istruzioni DATA (nel loro ordine di apparizione secondo il numero di riga). Allo stesso tempo viene impostato un indicatore sulla prima costante che è nella tabella. L'indicatore viene fatto avanzare attraverso la tabella, costante dopo costante, a mano a mano che i dati vengono assegnati alle variabili delle istruzioni READ o MAT READ. 9L'indicatore lo si può cambiare facendolo tornare ancora alla prima costante mediante l'istruzione RESTORE).

Nelle istruzioni DATA le costanti carattere che contengono meno di 18 caratteri vengono riempite alla loro destra con spazi vuoti fino alla lunghezza di 18 caratteri prima di essere assegnate alle variabili carattere. Le costanti carattere che contengono più di 18 caratteri vengono troncate alla loro destra ad una lunghezza di 18 caratteri prima di essere assegnate alle variabili carattere.

Note relative a DATA

- Ciascuna costante in una istruzione DATA deve essere dello stesso tipo di quella specificata per una variabile a cui si deve assegnare la costante nella corrispondente istruzione READ. Così se la terza costante che è in una DATA è una costante carattere, allora la variabile della READ, a cui questa viene assegnata, deve essere anch'essa una variabile carattere.
- Le istruzioni DATA possono venir inserite in ogni punto di un programma, sia prima che dopo la READ a cui forniscono i dati.
- Avverrà un errore se le istruzioni DATA non contengono sufficienti costanti per le READ emesse.
- Le istruzioni DATA non possono essere usate in un programma assegnato ad una delle Funzioni-tasto.

Esempio

un esempio di istruzione DATA è il seguente:

```
110 DATA 'BILL',21.60,'CHARGE',15.40
```

In questo esempio, le costanti carattere (BILL e CHARGE) e le costanti aritmetiche (21.60 e 15.40) vengono inserite nella tabella di dati interni.

DEF FNnome di funz. [(variab. arit. [,variab. arit.] . .)] [=espres. arit.]
FNEND [commento]
RETURN [espressione aritmetica]

DEF, FNEND, RETURN

L'istruzione DEF è non elaborabile, ma solo per l'informazione. Questa specifica permette all'utente di definire una funzione aritmetica per essere usata più oltre nel programma. L'istruzione FNEND indica la fine della funzione e la RETURN specifica il valore finale della funzione. La sintassi dell'istruzione DEF può essere costituita sia da una singola riga sia da una funzione multiriga. (La sintassi delle istruzioni FNEND e RETURN saranno spiegate subito dopo).

Funzione di Riga Singola

DEF FNnome di funz. (variab. arit. [,variab. arit.] . .) [=espres. arit.]

dove:

nome di funzione è un qualsiasi carattere di tutto l'alfabeto BASIC. Questo carattere assieme ad FN rappresenta il nome della funzione definita.

(variabile aritmetica) è una o più variabili aritmetiche semplici alle quali verrà assegnato un valore quando sarà richiamata la funzione (questa dovrà essere racchiusa tra parentesi).

espressione aritmetica è una espressione aritmetica che può essere chiamata come riferimento in un'altra istruzione durante l'elaborazione di un programma. Un esempio di una istruzione DEF di funzione di riga singola è il seguente:

120 DEF FNA(R)=2*R +100

Funzione Multiriga

DEF FNnome di funz. (variab. arit. [,variab. arit.] . .) [=espres. arit.]

dove:

nome di funzione è un qualsiasi carattere di tutto l'alfabeto BASIC. Questo carattere assieme ad FN rappresenta il nome della funzione definita.

(variabile aritmetica) è una o più variabili aritmetiche semplici che ricevono un valore quando la funzione compare in un riferimento. Queste variabili facoltative devono essere racchiuse tra parentesi.

RETURN [espressione aritmetica]

dove:

espressione aritmetica è una espressione aritmetica che specifica il valore della funzione definita dall'utente alla funzione di riferimento.

FNEND [commento]

L'istruzione **FNEND** è non eseguibile ed indica semplicemente la fine di una funzione multiriga. Il valore della funzione viene specificato in una espressione aritmetica di una istruzione **RETURN**. Il commento è facoltativo.

Allorchè viene incontrato, durante l'elaborazione di un programma, un riferimento ad una funzione definita dall'utente, il valore di ciascun parametro che è nell'espressione viene utilizzato per inizializzare la corrispondente variabile. Le variabili facoltative devono essere aritmetiche e devono corrispondere al numero specificato nel riferimento della funzione. Se l'espressione è presente, la funzione viene definita sulla stessa riga ed il suo valore è il valore di quella espressione. Questa è una funzione di riga singola. Se l'espressione non viene specificata, la specifica **DEF** diventa l'inizio di una funzione multiriga. In questo caso, l'istruzione **FNEND** indica la fine della funzione ed il valore della funzione viene specificato in una espressione aritmetica di una specifica **RETURN**.

Note relative a DEF

- Una funzione la si può definire in ogni punto di un programma BASIC, sia prima che dopo il suo riferimento.
- Una funzione con un certo nome può essere definita una sola volta nello stesso programma.
- Una funzione non può contenere riferimenti a se stessa o ad altre funzioni che le si riferiscono nelle loro definizioni.
- L'espressione nella **RETURN** è necessaria per le funzioni multiriga (vedi **GOSUB** e **RETURN**).
- Un riferimento di funzione ad una funzione definita dall'utente può comparire in ogni punto di una espressione BASIC ove può apparire una costante, variabile, riferimento di elementi d'insieme sottoscritto o riferimento di funzioni di sistema.
- Le variabili aritmetiche possiedono, nella istruzione **DEF**, un significato speciale. Conseguentemente è possibile avere una variabile con lo stesso nome di un'unica variabile aritmetica usata altrove nel programma. Ciascuna viene riconosciuta come unica e non viene fuori, da questo doppio uso, alcun conflitto di nomi o di valori.
- Il numero massimo di funzioni definite dall'utente in un programma è 29, e il numero massimo di riferimento a funzioni varia in relazione alla complessità delle istruzioni che le richiamano.
- Le funzioni definite dall'utente cui ci si riferisce durante una operazione di immissione od emissione non possono svolgere a loro volta una qualsiasi immissione od emissione di se stesse.

- Una volta che il comando è stato trasferito alla istruzione DEF senza riferimento alla funzione, il comando viene assegnato alla prima istruzione eseguibile che segue la definizione di funzione (seguente l'istruzione DEF in caso di funzioni di riga singola o seguente la FNEND nel caso di funzioni multiriga).
- L'ultima istruzione elaborabile che precede la FNEND dovrebbe essere una istruzione RETURN, STOP, CHAIN, od un incondizionato GOTO per impedire che il comando passi alla istruzione FNEND.
- Se una definizione di funzione cambia il valore di una variabile che possiede un riferimento nella stessa istruzione che richiama la funzione, si possono ottenere dei risultati inattendibili.

Esempio

Gli esempi che seguono illustreranno l'elaborazione di istruzioni DEF:

```
10 DEF FNA (X) = X↑3/2
20 Y = 10
30 Z = FNA (Y)
```

Dopo l'elaborazione dell'istruzione 30, alla variabile Z sarà assegnato il valore intero 500.

Nel prossimo esempio, la variabile R avrà il valore intero 72 dopo l'elaborazione della istruzione 80. Quando l'80 sarà elaborata, il valore attuale di Y, che è 2, verrà sostituita, per ciascuna ricorrenza della variabile di comodo X, nella espressione aritmetica della istruzione 100. Poichè la funzione FNC, definita nella 100, fa uso nella propria definizione della funzione FNB, viene sostituito il valore 2, per ciascuna ricorrenza di X, nella espressione aritmetica della istruzione 90. Il valore che ne risulta, 47, sostituirà allora il riferimento di funzione FNB(X) nella 100. Il valore attuale di Y che è 2, verrà poi sommato a 47 ed il valore che ne risulterà, 49, sostituirà il riferimento di funzione FNC(Y) nella istruzione 80. Questo valore sarà sommato a 23 ed il valore risultante, pari a 72, assegnato alla variabile R.

```
70 LET Y = 2
80 LET R = FNC(Y) + 23
90 DEF FNB(X) = 5*X**2+ 27
100 DEF FNC(X) = FNB(X) + X
```

L'esempio che segue mostra una definizione di funzione multiriga. Una volta che le istruzioni saranno state elaborate, sia C che D avranno un valore uguale a 7.

```
10 A = 2
20 B = 2
30 DEF FNA (X, Y)
40 IF X>0 GOTO 60
50 RETURN X-Y
60 RETURN X+Y
80 C = FNA (A,B)
90 D = FNA (A,B)
```


DIM nome di insieme (righe [,colonne]) [,nome di insieme (righe [,colonne])] . . .
--

DIM

L'istruzione DIM permette di specificare esplicitamente la grandezza degli insiemi. La sintassi della istruzione DIM è la seguente:

DIM nome di insieme (righe [, colonne]) , nome di insieme (righe[,
colonne]) ...

dove:

nome di insieme è un insieme aritmetico o di caratteri che deve essere dimensionato.

Righe, colonne sono costanti intere, prive di segno e diverse da zero, che specificano le dimensioni dell'insieme. Insiemi monodimensionali hanno bisogno che vengano immesse solo le righe. Gli insiemi bidimensionali hanno bisogno che vengano immesse righe e colonne.

Un insieme monodimensionale il cui nome venga specificato in una istruzione DIM viene definito come se avesse il numero di elementi rappresentato dall'immissione di righe.

Un insieme bidimensionale il cui nome venga specificato in una istruzione DIM viene definito come se avesse il numero di righe e di colonne immesse nella istruzione.

Il valore iniziale di ciascun elemento di insieme aritmetico è zero; ciascun elemento di un insieme di caratteri ha 18 caratteri spazi vuoti.

Note relative a DIM

- Un nome di insieme non può comparire in una DIM se non è stato precedentemente definito, implicitamente od esplicitamente in una istruzione USE oppure in una precedente istruzione DIM.
- Un insieme aritmetico o di caratteri ad una o due dimensioni può essere definito in una DIM.
- La massima grandezza permessa per ciascuna dimensione di un insieme è 255.

Esempio

Un esempio di istruzione DIM è il seguente

20 DIM Z\$(5), A(4,2)

Il risultato della precedente è:

Z\$= cinque stringhe (elementi dell'insieme) di 18 caratteri vuoti ciascuna.

L'insieme A possiede 4 righe di due colonne ciascuna:

0	0
0	0
0	0
0	0

END

L'istruzione END permette di specificare la fine logica di un programma BASIC e finire l'elaborazione di un programma. La sintassi della END è la seguente.

END [commento]

dove:

commento è facoltativo.

L'istruzione END sta a dire che il programma si intenderebbe finito. Può essere immessa in ogni punto di un programma BASIC. Allorchè, durante l'elaborazione di un programma BASIC, si incontra una END, avviene che tutti i file aperti vengano chiusi e finisca l'elaborazione. Ciò che fa l'istruzione END è identico a ciò che fa l'istruzione STOP.

Note relative a END

L'istruzione END è facoltativa. Se viene omessa, il sistema assume che ci sia una END dopo l'istruzione di programma con il numero di riga più grande.

Esempio

Un esempio di istruzione END è il seguente:

0910 END PG4

FNEND [commento]

FNEND

Per una descrizione completa della istruzione FNEND, vedi DEF, FNEND, RETURN.

FOR variab. di comando=espres. arit. TO espres. arit. [STEP espres. arit.]
NEXT variabile di comando

FOR E NEXT

Quando si trovano assieme, una istruzione FOR e la sua collegata NEXT, delimitano una iterazione. Un gruppo di istruzioni BASIC che possono essere elaborate un certo numero di volte. L'istruzione FOR definisce l'inizio della iterazione e specifica le condizioni della sua elaborazione e della sua conclusione. L'istruzione NEXT definisce la fine della iterazione.

La sintassi delle istruzioni FOR e NEXT è la seguente:

FOR variabile di comando=espressione aritmetica TO espressione aritmetica [STEP espressione aritmetica]

NEXT variabile di comando

dove:

variabile di comando è una variabile aritmetica semplice:

espressione aritmetica sono le espressioni che specificano il valore iniziale della variabile di comando, il valore finale della variabile di comando (a cui finirà l'elaborazione della iterazione) e l'incremento che la variabile di comando subirà dopo ciascuna elaborazione della iterazione. Se vengono omessi STEP e l'ultima espressione aritmetica, si assume un incremento uguale a 1.

Tutte le espressioni vengono valutate prima dell'elaborazione di queste istruzioni. Viene confrontato il valore iniziale della variabile di comando con il valore finale della stessa. Se il valore iniziale è maggiore (minore per incrementi negativi) del valore finale, la iterazione non viene elaborata. Invece, se il valore della variabile di comando resta immutato, allora il comando passa alla istruzione che segue la NEXT.

Se la iterazione viene elaborata, la variabile di comando viene impostata uguale al valore iniziale, e vengono elaborate le istruzioni che fanno parte dell'iterazione. Quando poi verrà elaborata la NEXT, l'incremento specificato verrà sommato alla variabile di comando, che sarà poi confrontata con il valore finale specificato. Se la variabile di comando è minore od uguale (maggiore od uguale, per incrementi negativi) del valore finale, l'iterazione viene elaborata di nuovo ed il ciclo continuerà fino a quando si avrà un incremento che renda la variabile di comando maggiore (o minore nel caso di incrementi negativi) del valore finale specificato. A questo punto, la variabile di comando viene impostata ad un valore superiore al suo ultimo valore ed il comando prosegue oltre fino alla prima istruzione da elaborare che segue la NEXT.

Note relative a FOR e NEXT

- Il valore della variabile di comando può essere modificato da istruzioni inserite all'interno dell'iterazione FOR, purtuttavia il suo valore iniziale, il valore finale e l'incremento vengono stabiliti durante la prima elaborazione della istruzione FOR e non vengono influenzati da alcuna istruzione posta all'interno della iterazione FOR.

- Se viene omessa nella istruzione FOR la espressione facoltativa aritmetica STEP, il valore dell'incremento viene automaticamente impostato a + 1.
- Se, quando viene eseguita l'istruzione FOR, il valore iniziale che bisogna assegnare alla variabile di comando è maggiore (o minore nel caso di incrementi negativi) del valore finale, l'iterazione non viene elaborata, non viene assegnato alcun valore alla variabile di comando e l'elaborazione procede con la prima istruzione elaborabile che segue l'istruzione NEXT collegata.
- Se il valore della espressione aritmetica STEP è zero, l'iterazione FOR viene elaborata un numero infinito di volte, o fino a che il valore della variabile di comando venga deliberatamente impostato al di là del valore finale specificato.
- Viene permesso un trasferimento del comando dentro o fuori di una iterazione FOR; tuttavia una istruzione NEXT non può essere elaborata a meno che sia stata precedentemente elaborata la sua corrispondente istruzione FOR.
- Iterazioni FOR possono essere incuneate all'interno di altre, purchè la iterazione FOR interna cada completamente dentro la iterazione FOR esterna (vedi l'esempio seguente). Le iterazioni FOR Incuneate non dovranno far uso della stessa variabile di comando.
- Il massimo numero di iterazioni FOR incuneate è 15.

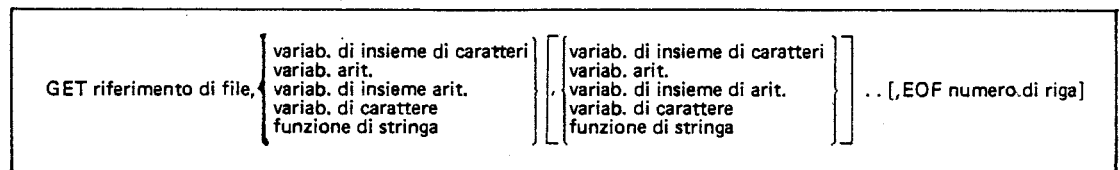
Esempio

L'esempio seguente mostra un campione di iterazione FOR che incrementa la variabile di comando di 2 fino a che non venga superato il valore 25.

```
20 FOR I = 1 TO 25 STEP 2
.
.
.
90 NEXT I
```

Il prossimo esempio mostrerà la tecnica corretta per l'incuneamento di iterazioni FOR. La iterazione interna viene elaborata 100 volte per ciascuna elaborazione della iterazione esterna.

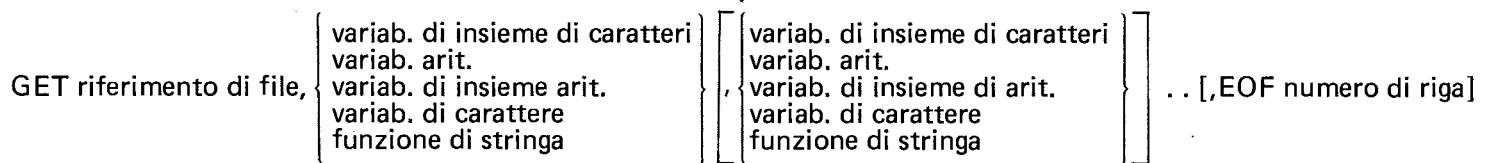
```
10 FOR J = A TO B STEP C
.
.
.
150 FOR K = 1 TO 100
.
.
.
250 NEXT K
.
.
.
300 NEXT J
```



GET

L'istruzione GET permette di assegnare dei valori, prelevandoli da un file specificato, alle variabili di riferimento. il file che contiene i valori da assegnare deve essere già stato aperto per l'immissione con una istruzione OPEN.

Il formato della istruzione GET è il seguente:



dove:

riferimento di file che va da FLO a FL9, serve per identificare il file che contiene i valori che devono essere assegnati. Questa istruzione deve essere uguale a quella specificata nella OPEN corrispondente.

variabile è una unica variabile, un riferimento di insieme indirizzato od un riferimento sottostringa. E' necessaria solamente una variabile. Se viene immessa più di una variabile, queste dovranno tra loro essere separate da virgole.

EOF numero di riga è il numero della istruzione a cui salterà il programma quando si sarà raggiunta la fine del file. Se viene fatto questo salto, il file viene chiuso e prima di un ulteriore accesso dovrà essere riaperto.

Quando verrà elaborata l'istruzione GET verranno assegnati i valori alle variabili di riferimento prelevandoli dal file specificato (FLO-FL9). Specifiche GET successive alla prima e relative allo stesso file determineranno l'assegnazione dei valori del file iniziando il prelevamento dalla posizione a quel punto occupata nel file.

Gli indici nei riferimenti di variabile vengono valutati quando si incontrano, da sinistra a destra. Pertanto, una variabile assegnata in una GET può essere usata come indice di un'altra variabile della stesa istruzione.

Note relative a GET

- Un file che è stato attivato come file di emissione non può essere specificato in una istruzione GET. Bisogna prima chiuderlo e poi riaprirlo (in una OPEN, come file di immissione).
- Ciascun valore assegnato in una istruzione GET deve essere dello stesso tipo di dati (aritmetici o di caratteri) della corrispondente variabile.
- Una istruzione GET che si riferisca ad un file attualmente chiuso causa la fine dell'elaborazione del programma.

Esempio

Un esempio di istruzione GET è il seguente:

```
0090 GET FL4, X, Y, Z, A(4), A(5), D$, EOF 620
```

In questo esempio i valori del file FL4 vengono assegnati rispettivamente alle variabili X, Y, Z ed al quarto ed al quinto elemento dell'insieme A e alla variabile carattere D\$. Alla fine del file, il programma salta alla istruzione numero 620.

GOSUB numero di riga [[,numero di riga] ... ON espressione aritmetica]
RETURN

GOSUB E RETURN

Le istruzioni GOSUB e RETURN vengono usate assieme per creare dei sottoprogrammi. L'istruzione GOSUB trasferisce il comando, condizionatamente od incondizionatamente, ad una istruzione. L'istruzione RETURN trasferisce il comando alla prima istruzione eseguibile che segue l'ultima istruzione GOSUB attiva che è stata elaborata.

La sintassi della GOSUB può essere di due tipi. La sintassi di GOSUB incondizionata è:

GOSUB numero di riga

dove:

numero di riga è il numero della istruzione a cui deve essere trasferito il comando.

La sintassi di GOSUB condizionata è:

GOSUB numero di riga [, numero di riga]... ON espressione aritmetica

dove:

numero di riga è un gruppo di numeri di riga separati tra loro da virgole. E' necessario almeno un numero di riga.

espressione aritmetica determina l'istruzione alla quale viene trasferito il comando.

Il formato della istruzione RETURN è semplicemente RETURN.

L'elaborazione di una istruzione GOSUB del primo tipo determina un trasferimento incondizionato del comando alla riga di cui è specificato il numero.

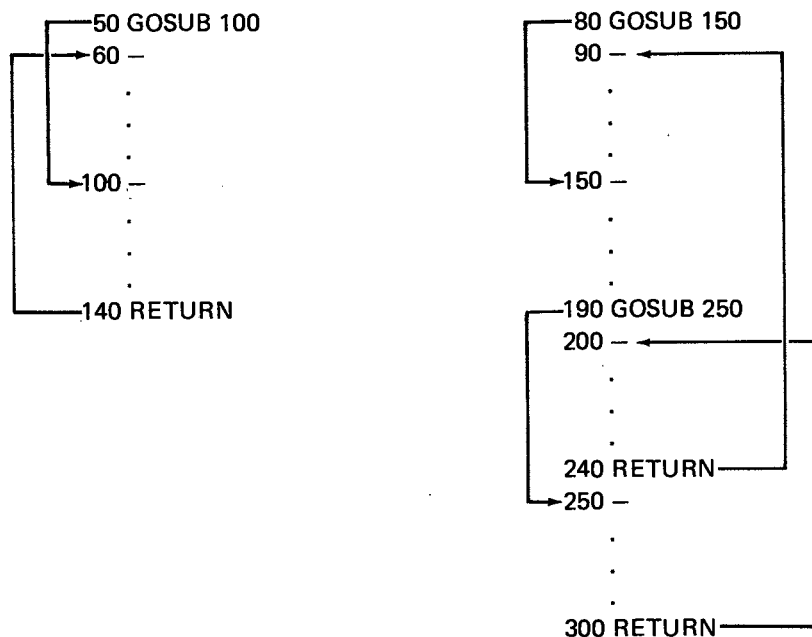
L'elaborazione di una istruzione GOSUB del secondo tipo determina l'elaborazione dell'espressione aritmetica ed il trasferimento del comando alla riga la cui posizione numerica nell'elenco dei numeri di riga (leggendo da sinistra verso destra) sia uguale al valore del numero intero (troncato) della espressione. In questo modo una espressione con un valore di 2.75 determinerà il trasferimento del comando al secondo numero dell'elenco. Se l'espressione ha un valore minore di 1 o maggiore del numero d'ordine più alto tra tutti i numeri di riga elencati, il comando viene allora trasferito alla prima istruzione eseguibile che segue l'istruzione GOSUB stessa.

Quando una istruzione GOSUB trasferisce il comando ad un'istruzione non eseguibile, il comando viene trasferito alla prima istruzione eseguibile che segue quella non eseguibile specificata.

L'elaborazione di una istruzione RETURN determina il trasferimento incondizionato del comando alla prima istruzione eseguibile seguente l'ultima GOSUB attiva che è stata elaborata.

Esempio

Gli esempi seguenti mostrano l'elaborazione dell'istruzioni GOSUB e RETURN:



L'esempio seguente mostra un GOSUB ad una funzione chiave:

```

100 GOSUB 9994
.
.
.
.
9994 PRINT 'IN MAIN PROGRAM'
9995 RETURN
  
```

LOAD0,KEY4
 9994 REM
 0010 PRINT 'IN KEYGROUP 9994'
 0020 RETURN

In questo esempio la 100 determina un salto del programma alla funzione REM definita dal tasto 4. Se il tasto 4 non è definito oppure è definito con qualcos'altro di diverso da REM, il programma ricerca la riga 9994 all'interno del programma principale.

GOTO numero di riga [[,numero di riga] ... ON espressione aritmetica]

GOTO

L'istruzione GOTO trasferisce il comando ad una determinata riga condizionatamente od incondizionatamente.

La sintassi della istruzione GOTO può essere quindi di due tipi. La sintassi della istruzione del primo tipo è:

GOTO numero di riga

dove:

numero di riga è il numero della istruzione a cui si trasferisce il comando.

La sintassi di GOTO condizionata (secondo tipo) è:

GOTO numero di riga [,numero di riga] ... ON espressione aritmetica

dove:

numero di riga è un gruppo di numeri di riga tra loro separati da virgole. E' necessario almeno un numero di riga.

espressione aritmetica determina il numero della istruzione cui va trasferito il comando.

L'elaborazione di una istruzione GOTO del primo tipo determina un trasferimento incondizionato del comando al numero di riga specificato.

L'elaborazione di un'istruzione GOTO del secondo tipo determina l'elaborazione dell'espressione aritmetica ed il trasferimento del comando alla riga la cui posizione numerica nell'elenco dei numeri di riga (leggendo da sinistra verso destra) sia uguale al valore del numero intero (troncato) della espressione. In questo modo una espressione con un valore di 2.75 determinerà il trasferimento del comando alla riga nella seconda posizione dell'elenco. Se l'espressione ha un valor minore di 1 o maggiore del numero d'ordine più alto tra i numeri di riga elencati, il comando verrà allora trasferito alla prima istruzione eseguibile che segue l'istruzione GOTO stessa.

Quando una istruzione GOTO trasferisce il comando ad una istruzione non eseguibile, il comando viene allora trasferito alla prima istruzione eseguibile che segue quella non eseguibile specificata.

La seguente istruzione trasferirà incondizionatamente il comando al numero di riga 20:

100 GOTO 20

Se $X=4$, la seguente istruzione trasferirà il comando al numero di riga 60:

50 GOTO 40, 60, 15, 100 ON $(X + 4)/4$

IF	$\left\{ \begin{array}{l} \text{espres. arit. operat. logico} \\ \text{espres. di carat. operat. logico} \end{array} \right.$	$\left\{ \begin{array}{l} \& \\ \end{array} \right.$	$\left\{ \begin{array}{l} \text{espres. arit. operat. logico} \\ \text{espres. di carat. operat. logico} \end{array} \right.$	$\left\{ \begin{array}{l} \text{THEN} \\ \text{GOTO} \end{array} \right.$	numero di riga
----	---	--	---	---	----------------

IF

L'istruzione IF permette di trasferire il comando del programma, in conformità con il risultato della valutazione di una espressione. La sintassi della IF è la seguente:

IF $\left\{ \begin{array}{l} \text{espres. arit. operat. logico} \\ \text{espres. di carat. operat. logico} \end{array} \right.$ $\left\{ \begin{array}{l} \& \\ | \end{array} \right.$ $\left\{ \begin{array}{l} \text{espres. arit. operat. logico} \\ \text{espres. di carat. operat. logico} \end{array} \right.$ $\left\{ \begin{array}{l} \text{THEN} \\ \text{GOTO} \end{array} \right.$ numero di riga

dove:

espressione aritmetica od espressione di caratteri sono espressioni aritmetiche o di caratteri oppure una funzione di stringa. E' necessaria solo una coppia di espressioni.

operatore logico è un operatore logico. E' necessario solamente un operatore.

& è il simbolo per AND. | è il simbolo per OR.

THEN e GOTO specificano che si dovrà trasferire il comando. Se si immette THEN, ogni volta che si fa la lista del programma sullo schermo video oppure lo si stampa, il sistema sostituisce GOTO a THEN.

numero di riga rappresenta il numero della istruzione a cui viene trasferito il comando se la espressione logica è vera.

Quando viene elaborata una istruzione IF, le espressioni vengono confrontate così come sono state specificate dall'operatore logico. Se la relazione è vera il comando viene trasferito al numero di riga che è stato specificato. Se la relazione non è vera, il comando viene trasferito alla prima istruzione eseguibile che segue la IF.

Prima del confronto, una costante carattere che contenga meno di 18 caratteri viene riempita con dei caratteri di spazi vuoti alla sua destra fino a raggiungere una lunghezza di 18 caratteri.

Una costante carattere che contenga più di 18 caratteri, prima del confronto viene troncata a destra ad una lunghezza di 18 caratteri. Una costante carattere che non contenga alcun carattere viene confrontata come se contenesse 18 caratteri spazi vuoti.

Se viene specificata &, ambedue le espressioni logiche devono essere vere affinché il comando passi al numero di riga specificata. Se viene specificata |, il comando passerà al numero di riga specificata se una delle due espressioni logiche è vera.

Se la relazione specificata è vera, o se l'istruzione che è stata specificata non è eseguibile, il comando viene passato all'prima istruzione eseguibile che segue quella non eseguibile specificata.

Note relative a IF

- Le espressioni che vengono confrontate nelle espressioni logiche devono contenere dati dello stesso tipo (di caratteri od aritmetici).
- Nella istruzione IF, THEN e GOTO sono intercambiabili. Può essere usato solo uno dei due, non entrambi. GOTO è memorizzato dal sistema.
- Un confronto di uguaglianza deve coincidere esattamente fino alla 13ma cifra.

Esempio

I seguenti esempi mostrano una varietà di istruzioni IF:

```
30 IF A(3) ≠ X+2/Z GOTO 85
```

```
40 IF R$ = 'CAT' GOTO 70
```

```
50 IF S2 = 37.222 GOTO 120
```

```
60 IF X>Y GOTO 90
```

```
70 IF A>B| C>D GOTO 110
```

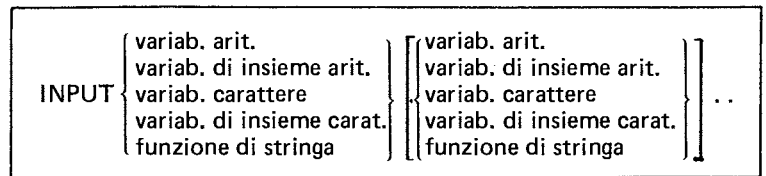
```
80 IF A$ = 'JOB' & B$ = 'DATE' GOTO 100
```

Nella riga 40, ad esempio, se la variabile carattere R\$ contiene la parola CAT, il comando del programma passa alla riga 70. Nella 70, se l'una o l'altra di A>B o C>D è verificata, il comando passa alla riga 110.

$\begin{Bmatrix} \text{stringa di caratteri} \\ \text{maschera di stampa} \end{Bmatrix} \dots \text{maschera di stampa} \begin{Bmatrix} \text{stringa di caratteri} \\ \text{maschera di stampa} \end{Bmatrix} \dots$

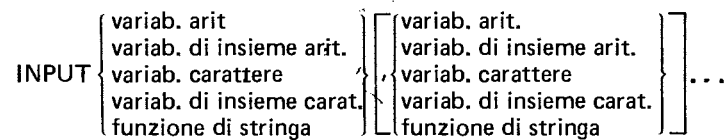
IMAGE

L'istruzione Image si usa per comandare il modo in cui vanno preparati i dati da stampare o visualizzare. Per una descrizione completa dell'istruzione Image, vedi PRINT USING e Image oppure MAT PRINT USING.



INPUT

L'istruzione INPUT permette di assegnare valori alle variabili tramite tastiera durante l'elaborazione del programma. La sintassi della istruzione INPUT è la seguente:



dove:

variabile sono variabili semplici aritmetiche o di carattere, riferimenti d'indice ad un elemento di insieme, o riferimenti di sottostringa. I riferimenti di righe o colonne ad un elemento di insieme debbono essere chiusi tra parentesi. In una istruzione INPUT è necessario solamente una variabile, quantunque possano venir specificate più variabili.

Quando viene elaborata una istruzione INPUT, viene visualizzato sullo schermo un punto interrogativo e l'elaborazione del programma si ferma. Bisogna allora immettere l'elenco dei valori che saranno assegnati, nello stesso ordine in cui vengono immessi, alle variabili elencate nella istruzione INPUT. Quando sarà stato immesso tutto l'elenco, battete EXECUTE per riprendere l'elaborazione del programma.

Gli indici delle variabili degli insiemi che compaiono nella istruzione INPUT vengono valutati quando si incontrano. Così, una variabile assegnata in una istruzione INPUT può essere usata successivamente dall'indice di un'altra variabile nella stessa istruzione.

Costanti carattere che contengano meno di 18 caratteri vengono riempite con dei caratteri di spazi vuoti alla loro destra fino a raggiungere una lunghezza di 18 caratteri, prima di venir assegnate alle variabili carattere. Costanti carattere che contengono più di 18 caratteri, vengono troncate a destra ad una lunghezza di 18 caratteri prima di venir assegnate. Costanti carattere che non contengono alcun carattere vengono assegnate come se contenessero 18 caratteri spazi vuoti.

Note relative a INPUT

- Ciascun valore immesso deve essere dello stesso tipo di dati (di caratteri od aritmetici) del corrispondente riferimento di variabile che è nella istruzione INPUT. I tipi di dati possono essere mischiati nella stessa istruzione.
- Ogni valore immesso deve essere separato dal successivo mediante una virgola. Due virgole poste successivamente all'altra vengono ignorate. Per finire la serie battere EXECUTE.
- Una costante carattere deve essere chiusa tra apici.
- Il numero di valori immessi all'atto dell'elaborazione deve essere uguale al numero di variabili di riferimento specificate nella istruzione INPUT. Se non vengono immessi abbastanza valori, il programma farà richiesta di altra immissione. Se si immettono troppe variabili, le immissioni che sono in eccesso saranno ignorate dal programma.

Esempio

Un campo di istruzione INPUT è il seguente:

```
60 INPUT A$, B, X, Y(X)
```

```
'YES', 18.6, 8, 1.3597E-6
```

Quando verrà elaborata la riga 60 si potrà immettere l'elenco dei valori mostrati. I valori verranno allora assegnati alle relative variabili corrispondenti.

$$[LET] \left\{ \begin{array}{l} \text{variab. arit.} \\ \text{variab. di insieme arit.} \end{array} \right\} \left[\begin{array}{l} \text{variab. arit.} \\ \text{variab. di insieme arit.} \end{array} \right] \dots = \text{espres. arit.} \left. \begin{array}{l} \text{variab. carattere} \\ \text{variab. di insieme carat.} \\ \text{funzione di stringa} \end{array} \right\} \left[\begin{array}{l} \text{variab. carattere} \\ \text{variab. di insieme carat.} \\ \text{funzione di stringa} \end{array} \right] \dots = \left\{ \begin{array}{l} \text{espres. carat.} \\ \text{costant. carat.} \end{array} \right\}$$

LET

L'istruzione **LET** permette di assegnare il valore di una espressione ad uno o più variabili. La sintassi della istruzione **LET** è la seguente:

$$[LET] \left\{ \begin{array}{l} \text{variab. arit.} \\ \text{variab. di insieme arit.} \end{array} \right\} \left[\begin{array}{l} \text{variab. arit.} \\ \text{variab. di insieme arit.} \end{array} \right] \dots = \text{espres. arit.} \left. \begin{array}{l} \text{variab. carattere} \\ \text{variab. di insieme carat.} \\ \text{funzione di stringa} \end{array} \right\} \left[\begin{array}{l} \text{variab. carattere} \\ \text{variab. di insieme carat.} \\ \text{funzione di stringa} \end{array} \right] \dots = \left\{ \begin{array}{l} \text{espres. carat.} \\ \text{costant. caratt.} \end{array} \right\}$$

dove:

variabili sono nomi di variabili scalari, riferimenti d'indice ad elementi d'insieme o riferimenti di sottostringa. E' necessaria una sola variabile, benchè, ne possano venir specificate diverse. I riferimenti di riga e colonna ad un elemento di insieme devono essere chiusi tra parentesi.

espressione è una espressione aritmetica o di carattere, una funzione di stringa, od una costante carattere.

Quando viene elaborata l'istruzione **LET**, viene valutata l'espressione, ed il valore che ne risulta viene assegnato alle variabili specificate, procedendo da sinistra a destra.

Costanti carattere che contengono meno di 18 caratteri vengono riempite con dei caratteri spazi vuoti alla loro destra fino a raggiungere una lunghezza di 18 caratteri prima di venir assegnate alle variabili carattere. Costanti carattere che contengono più di 18 caratteri, vengono troncate a destra ad una lunghezza di 18 caratteri prima di venir assegnate. Costanti carattere che non contengano alcun carattere vengono assegnate come se contenessero 18 caratteri spazi vuoti.

Note relative a LET

- I valori dei dati a destra del segno di uguale devono essere dello stesso tipo (aritmetici o di carattere) delle variabili alle quali vengono assegnati.
- Il tasto-parola LET è facoltativo.
- Costanti esadecimali si possono anche assegnare alle variabili carattere in istruzioni LET. La costante deve iniziare con X' seguito da un numero pari di caratteri da 0 a 9 e da A ad F. La fine della costante deve essere indicata da un altro apice ('). Vedere, in Appendice, la tavola con le 256 rappresentazioni esadecimali consentite in BASIC.
- Nelle istruzioni di assegnazione vengono permessi riferimenti d'indice ad elementi di insieme.
- Il massimo numero di variabili a sinistra del segno di uguale in una istruzione di assegnazione multipla viene limitato unicamente dalla lunghezza della riga (64 caratteri).

Esempio

Alcuni campioni di istruzione LET sono i seguenti:

```
10 LET Z$ = 'CAT'
```

```
20 LET X = 9
```

```
30 LET Y(X) = 2
```

```
40 Y(X), X = X/Y(X)
```

```
50 LET A$ = X'0201130903'
```

Dopo l'elaborazione della riga 10, la variabile carattere Z\$ conterrà la parola CAT seguita da 15 caratteri di spazi vuoti.

Dopo l'elaborazione della riga 30, il nono numero dell'insieme aritmetico monodimensionale (Y) avrà il valore intero 2.

Dopo l'elaborazione della riga 40, la variabile aritmetica X avrà il valore decimale 4.5. Il nono elemento dell'insieme aritmetico monodimensionale Y avrà il valore decimale 4.5. L'azione della riga 40 è di valutare per prima cosa l'espressione a destra in conformità con i valori correnti delle variabili Y(X) e X, 2, e 9, rispettivamente. Il valore che ne risulta, 4.5 viene per prima cosa assegnato a Y(9), poi alla variabile X.

Dopo l'elaborazione della riga 50, la variabile A\$ conterrà la costante esadecimale elencata.

NEXT variabile di comando

NEXT

Per una esauriente descrizione della istruzione NEXT vedere FOR e NEXT.

OPEN rifer. di file, 'indiriz. dell'unità' [,numero di file] [, { 'utente ID' }] { IN }
 { variabile carat. } { 'OUT' }

OPEN

L'istruzione OPEN permette di:

- Identificare un'unità collegata mediante un codice di riferimento di file.
- Assegnare spazio al file.
- Specificare il numero di file.
- Specificare l'identificazione che deve essere usata con il file.
- Specificare l'utilizzo del file (immissione od emissione).

La sintassi della istruzione OPEN è la seguente:

OPEN rifer. di file, 'indiriz. dell'unità' [,numero di file] [, { 'utente ID' }] { IN }
 { variabile carat. } { 'OUT' }

dove:

riferimento di file, che va da FL0 a FL9, serve per specificare il file logico che si deve associare al file materiale identificato dai restanti parametri che ci sono nella istruzione OPEN. A questa specificazione di file fanno riferimento le istruzioni GET, PUT, RESET, CLOSE, PRINT, PRINT USING, MAT PRINT, MAT PUT, MAT GET, e MAT PRINT USING.

'indirizzo dell'unità' è l'indirizzo dell'unità fisica su cui è disponibile il file stesso. Indirizzi accettati sono: 'E80' per l'unità nastro incorporata, 'E40' per l'unità nastro ausiliaria e '500' per la stampatrice.

numero di file è una costante od una variabile numerica. Il valore del numero di file viene allora usato per accedere al file corrispondente sull'unità specificata. I valori decimali delle variabili vengono troncati ad un numero intero. Vedi Numeri di File Negativi.

variabile carattere è una variabile carattere oppure una costante carattere chiusa tra apici. I primi 17 caratteri forniscono il campo di identificazione nel record di testata di un file che è stato aperto per l'emissione. Se viene aperto un file per l'immissione e viene specificata una variabile carattere per l'identificazione, (utente ID) quando l'istruzione OPEN verrà elaborata, verrà posizionato dal file nella variabile carattere.

IN oppure OUT servono ad indicare se il file deve essere usato per immissione (IN) o per emissione (OUT).

Un campione di istruzione OPEN è il seguente:

OPEN FL1, 'E80',3,IN

In questo esempio, il codice di riferimento di file FL1 verrà assegnato al file numero 3 che si trova nell'unità nastro incorporata ('E80') per l'immissione (IN).

Numeri di File Negativi

Numeri di file negativi possono essere usati per creare un file di interscambio generale (origine o dati) tipo 2 che possa essere elaborato sia da BASIC o APL. Le seguenti regole governano l'uso dei numeri di file negativi.

- Il numero negativo deve inizialmente essere assegnato ad una variabile. Costanti negative non possono essere immesse direttamente nella OPEN con numeri di file.
Ad esempio:

```
10 F = -1
20 OPEN FL2, 'E80', F, OUT
```

- Ciascuna istruzione nel file creato (origine o dati) deve finire con un carattere esadecimale X'E3'. Questo carattere è il codice interno che viene interpretato con fine di riga dal 5100.

L'esempio seguente mostra la creazione di un file singolo mediante un numero di file negativo.

```
10 F = -1
20 OPEN FL2, 'E80', F, OUT
30 A$ = X'E3'
40 PRINT FL2, '0090 PRINT &PI, &E'; A$;
```

In questo esempio, il riferimento di file FL2 conterrà il numero di file -1, che consiste della specifica singola per visualizzare &PI e &E. La costante esadecimale X'E3' viene assegnata a A\$, che finisce l'istruzione origine (40).

Note relative a OPEN

- Una OPEN deve venir emessa prima di una GET, MAT GET, PRINT, MAT PRINT, PRINT USING, MAT PRINT USING, PUT o MAT PUT che utilizzano il riferimento di file (FL0 - FL9) specificato nella OPEN stessa.
- Se un file è già aperto, la OPEN determina un messaggio di errore.
- Una volta che è stato aperto il file, non bisogna estrarre la bobina a nastro fino a che il file non venga chiuso.
- Un apice (') non può venir inserito in una costante carattere utilizzata per un utente ID.

P PAUSE [commento]

PAUSE

L'istruzione PAUSE permette di interrompere l'elaborazione del programma per svolgere operazioni con l'elaboratore. La sintassi della specifica PAUSE è la seguente:

PAUSE [commento]

dove:

commento è facoltativo.

Quando, durante l'elaborazione di un programma si incontra una PAUSE, l'elaborazione viene interrotta e viene mostrato il messaggio:

PAUSE s

sulla riga zero dello schermo video; s è il numero di riga della istruzione PAUSE eseguita. Per riprendere l'elaborazione del programma, bisogna immettere un comando GO. Tener presente che non si possono rinumerare le istruzioni (vedi RENUM, Capitolo 2) oppure modificare una istruzione richiamata (vedi DEF, FNEND, RETURN) mentre il programma è interrotto.

La seguente istruzione farà visualizzare il messaggio PAUSE 0080, e farà sospendere l'elaborazione fino a quando verrà emesso il comando GO:

80 PAUSE

$$\text{PRINT [rifer. di file]} \left[\begin{array}{l} \text{espres. arit.} \\ \text{espres. di carat.} \\ \text{TAB (espres.)} \end{array} \right] \left[\begin{array}{l} \text{costante carat.} \\ \text{[costante carat.]} \end{array} \right] \left[\begin{array}{l} \{ \{ ; \} \} \\ \{ \{ ; \} \} \end{array} \right] \left[\begin{array}{l} \text{espres. arit.} \\ \text{espres. di carat.} \\ \text{TAB (espres.)} \end{array} \right] \dots \left[\begin{array}{l} \text{costante di carat.} \\ ; \end{array} \right]$$

PRINT

L'istruzione PRINT determina la visualizzazione sullo schermo video dei valori delle espressioni scalari specificate, oppure stampate dalla stampatrice, oppure scritte in un file su nastro. Quando si fa uso della istruzione PRINT, il formato di tutti i valori che vengono visualizzati è standardizzato, ma si può agire sulla spaziatura tra i valori posti sulla stessa riga.

La sintassi della istruzione PRINT è la seguente:

$$\text{PRINT [rifer. di file]} \left[\begin{array}{l} \text{espres. arit.} \\ \text{espres. di carat.} \\ \text{TAB (espres.)} \end{array} \right] \left[\begin{array}{l} \text{costante carat.} \\ \text{[costante carat.]} \end{array} \right] \left[\begin{array}{l} \{ \{ ; \} \} \\ \{ \{ ; \} \} \end{array} \right] \left[\begin{array}{l} \text{espres. arit.} \\ \text{espres. di carat.} \\ \text{TAB (espres.)} \end{array} \right] \dots \left[\begin{array}{l} \text{costante di carat.} \\ ; \end{array} \right]$$

dove:

riferimento di file è FLP (stampatrice) oppure da FL0 a FL9 (file logica). Questa immissione è facoltativa. I file da FL0 a FL9 devono precedentemente essere stati aperti con una istruzione OPEN.

espressioni sono le espressioni aritmetiche o di carattere che si devono visualizzare o stampare. Le espressioni possono venir separate mediante virgole o punti e virgola.

E' possibile agire sulla spaziatura delle espressioni visualizzate o stampate, inserendo delle virgole o dei punti e virgola tra le espressioni (vedi funzione TAB). Le espressioni possono essere aritmetiche o di carattere (le costanti di carattere devono essere chiuse tra apici). Senza alcuna specifica di espressioni, la istruzione PRINT si può usare per il completamento della visualizzazione o stampa di una riga in sospeso oppure l'inserimento di una riga di spazi vuoti.

Quando viene elaborata una istruzione PRINT, il valore di ciascuna espressione specificata viene convertito nell'appropriato formato di emissione così come si trova descritto nel paragrafo "Zone di Stampa", e viene visualizzato o stampato da sinistra a destra nell'ordine in cui appare nella PRINT.

Zone di Stampa

Generalmente ciascuna istruzione PRINT determina l'inizio di una nuova riga visualizzata o stampata; ciascuna riga viene inoltre divisa in zone di stampa che possono essere o zone di stampa completa o zone di stampa compatta oppure una combinazione delle due.

Zone di stampa completa possiedono 18 posizioni di caratteri e vengono specificate mediante una virgola delimitatrice. Ad esempio la notazione:

200 PRINT A, B, C

determinerà la visualizzazione del valore della variabile A ad iniziare dalla prima posizione della riga, la visualizzazione del valore di B ad iniziare dalla 19, posizione e quella del valore di C ad iniziare dalla 37 posizione.

Zone di stampa compatta sono variabili in lunghezza e normalmente producono delle righe di emissione più dense delle zone di stampa completa. I delimitatori punto e virgola vengono usati per specificare le zone di stampa compatte. La lunghezza delle zone di stampa compatte, per i diversi tipi di dati verrà spiegato nel paragrafo "Spaziatura dei Valori Visualizzati o Stampati".

Spaziatura dei Valori Visualizzati o Stampati

Il valore convertito di ciascuna espressione viene stampato o visualizzato nella propria zona di stampa.

Zone di stampa Completa (Virgola come Delimitatore): Per le espressioni aritmetiche ogni voce dei dati convertiti ha necessità di una zona di stampa completa di 18 posizioni di caratteri. Per dati di carattere, l'area di stampa o di visualizzazione è il valore più piccolo delle zone di stampa completa (il più piccolo multiplo di 18) e sufficientemente grande per contenere il dato.

Poichè quasi tutti i valori stampati o visualizzati sono più corti di 18 caratteri, una riga di zone di stampa completa produce usualmente colonne di emissione molto spaziate.

Zone di Stampa Compatta (Spazio o Punto e Virgola come Delimitatore);

Per le espressioni aritmetiche, la lunghezza delle zone compatte viene determinata dalla lunghezza del valore convertito, inclusi il segno, le cifre, il punto decimale e l'esponente, come mostrato in Figura 9.

Per le costanti carattere, la lunghezza della zona di stampa compatta è uguale alla lunghezza dei caratteri racchiusi tra virgolette singole, inclusi gli spazi vuoti, ma escludendo gli apici che precedono un apostrofo.

Per le variabili carattere, la lunghezza della zona di stampa compatta è uguale alla lunghezza della stringa di caratteri, meno gli spazi vuoti di coda

Le zone di stampa compatta producono di norma delle righe di stampa più dense di quelle delle zone di stampa completa.

Lunghezza del gruppo di dati Convertito (Caratteri)	Lunghezza della zona di stampa Compatta (Caratteri)	Esempio (b rappresenta uno spazio vuoto)
2 - 4	6	b7.3bb
5 - 7	9	b17.357bb
8 - 10	12	-45.63927bbbb
11 - 13	15	b1.73579E-23bbbb
14 - 16	18	-8.92270493115bbbbbb
17 - 19	21	-1.234567890123E-22
Il numero di cifre visualizzate o stampate è comandato dall'attuale valore assegnato a RD= (vedi Comando RD=, Capitolo 2).		

Figura 9. Lunghezza delle zone di stampa compatte per espressioni aritmetiche

Formati di emissione Standard per la Stampa o la Visualizzazione**Costanti carattere**

I caratteri effettivi chiusi in apici (inclusi gli spazi vuoti di coda ma escluso l'apice che precede un apostrofo) vengono stampati o visualizzati.

Variabili Carattere

I caratteri effettivi (esclusi gli spazi vuoti di coda) vengono stampati o visualizzati.

Espressioni aritmetiche

Formato I: Viene stampato o visualizzato il numero intero, che consiste di un segno (spazio vuoto o segno negativo) e di non più di 13 cifre decimali significative per i numeri interi il cui valore assoluto sia minore di $1E+14$.

Formato E: Viene stampato o visualizzato il numero in virgola mobile, che consista di un segno (spazio vuoto o segno negativo), di non più di 13 cifre decimali significative, un punto decimale posto successivamente alla prima cifra, la lettera E, ed un esponente provvisto di segno e che consiste di una o due cifre. Il formato E viene usato per stampare o visualizzare numeri il cui valore assoluto sia minore di $1E-1$ ovvero maggiore od uguale a $1E+14$. I valori stampati o visualizzati vengono arrotondati, non troncati.

Formato F: Viene stampato o visualizzato il numero in virgola fissa, il quale consiste di un segno (spazio vuoto o segno negativo), di non più di 13 cifre significative, e di un punto decimale posto nella posizione appropriata. Il formato F viene usato per stampare o visualizzare i valori che non rientrano nelle descrizioni dei precedenti formati I od E.

Visualizzazione dell'Operazione di Riga

Allorchè i valori delle espressioni vengono trasferiti allo schermo video, viene gestito un indicatore interno di posizione di carattere per mantenere traccia della successiva posizione disponibile ove può essere posizionato un carattere (od una cifra). Il trasferimento di questo indicatore di posizione di carattere prima, durante e dopo la visualizzazione di una espressione dipende dal tipo di espressione e dal delimitatore che segue nella istruzione PRINT. La figura 10 mostra come agisce l'indicatore nei casi possibili.

La posizione dell'indicatore della posizione di riga dello schermo può essere spostata in avanti (ad una posizione di carattere con un numero più grande) usando la funzione TAB.

Tipo di dati	Delimitatore	Posizione dell'indicatore prima della visualizzazione o della stampa	Posizione dell'indicatore dopo la visualizzazione o la stampa
Espressione aritmetica	Virgola	Se il record contiene spazio sufficiente per l'espressione, i dati saranno scritti ad iniziare dall'attuale posizione dell'indicatore. Se non vi è spazio sufficiente, viene scritto il contenuto del primo record e l'espressione comincerà all'inizio del secondo record.	L'indicatore viene trasferito al di là di tutte le posizioni restanti di tutta la zona di stampa. Se si giunge alla fine del record, il record viene stampato o visualizzato e l'indicatore viene impostato all'inizio del record successivo.
	Punto e virgola		L'indicatore viene trasferito al di là delle posizioni restanti della zona di stampa compatta. Se si giunge alla fine del record il record viene stampato o visualizzato e l'indicatore viene impostato all'inizio del record successivo.
	Nulla (non alla fine della istruzione)		Il record viene stampato o visualizzato e l'indicatore viene impostato all'inizio del record successivo.
	Nulla (alla fine della istruzione)		

Figura 10. Parte 1 ^ Posizioni dell'indicatore di riga per visualizzare o stampa.

Tipo di dati	Delimitatore	Posizione dell'indicatore prima della visualizzazione o della stampa	Posizione dell'indicatore dopo la visualizzazione o la stampa
Nulla	Virgola	Nessun dato viene visualizzato	L'indicatore viene trasferito oltre di 18 posizioni di caratteri. Se si giunge alla fine del record, il record viene stampato o visualizzato e l'indicatore viene impostato all'inizio del record successivo.
	Punto e virgola		L'indicatore viene trasferito oltre di 3 posizioni di caratteri. Se si giunge alla fine del record, il record viene stampato o visualizzato e l'indicatore viene impostato all'inizio del record successivo.
	Nulla (non alla fine della istruzione)		
	Nulla (alla fine della istruzione)		
Costante carattere	Virgola	Nessun dato viene visualizzato	L'indicatore viene trasferito al di là di tutti gli spazi restanti nella zona di stampa completa. Se si giunge alla fine del record, il record viene stampato o visualizzato e l'indicatore viene impostato all'inizio del record successivo.
	Punto e virgola		
	Nulla (non alla fine della istruzione)	La variabile viene scritta cominciando dall'attuale posizione dell'indicatore. Se si raggiunge la fine del primo record prima che tutti i dati siano scritti, il record viene visualizzato o stampato ed i dati restanti cominceranno all'inizio del record successivo.	L'indicatore viene lasciato nella posizione immediatamente seguente l'ultimo carattere.
	Nulla (alla fine della istruzione)		Il record viene scritto e l'indicatore viene impostato all'inizio del record successivo.

Figura 10. Parte 3^ Posizioni dell'indicatore di riga per visualizzazione o stampa

Tipo di dati	Delimitatore	Posizione dell'indicatore prima della visualizzazione o della stampa	Posizione dell'indicatore dopo la visualizzazione o la stampa
Variabile Carattere	Virgola	Se rimangono nel record almeno 18 posizioni di carattere i dati verranno scritti ad iniziare dall'attuale posizione dell'indicatore. Se ne rimangono meno di 18, il record viene stampato o visualizzato e la costante verrà scritta cominciando dall'inizio del record successivo. Se si raggiunge la fine del record prima che vengano scritti i caratteri, il record viene stampato o visualizzato ed i caratteri rimanenti saranno scritti a cominciare dall'inizio del record successivo.	L'indicatore viene trasferito al di là di tutte le posizioni restanti di tutta la zona di stampa. Se si giunge alla fine del record, il record viene stampato o visualizzato e l'indicatore viene impostato all'inizio del record successivo.
	Punto e Virgola	La variabile viene scritta cominciando dall'attuale posizione dell'indicatore. Se si raggiunge la fine del primo record prima che tutti i dati siano scritti, il record viene visualizzato o stampato ed i dati restanti cominceranno all'inizio del record successivo.	L'indicatore viene lasciato nella posizione immediatamente seguente l'ultimo carattere.
	Nulla (non alla fine della istruzione)		Il record viene scritto e l'indicatore viene impostato all'inizio del record successivo.
	Nulla (alla fine della istruzione)		

Figura 10. Parte 2^ Posizioni dell'indicatore di riga per visualizzazione o stampa

Funzione TAB

La funzione TAB (espressione) permette di allineare colonne di dati. Quando la espressione è un numero non intero, esso viene troncato. Così TAB(n) inizia la successiva emissione sulla colonna n della riga. Se l'attuale posizione sulla riga è superiore ad n, i dati vengono posti sulla riga successiva nella posizione n. TAB(n) può essere seguito da una virgola o da un punto e virgola con lo stesso risultato. Se n supera la lunghezza della riga, il dato viene posto nella posizione X dopo l'elaborazione della seguente espressione in cui n rappresenta la lunghezza della riga.

$$X = n - L * \text{INT}(n-1)/L$$

Ad esempio se la lunghezza della riga è 64 e $\text{TAB}(n) = 70$, il valore intero $((70-1)/64) = 1$ viene moltiplicato per (70-64). Così, l'emissione inizia nella posizione 6 della riga successiva. Questa formula è applicabile solo quando $L = 1-9999$.

Memoria di transito per le righe di stampa

La memoria di transito della riga di stampa è un luogo di memorizzazione temporanea dove vengono collocati i caratteri per ciascuna riga di stampa. Dopo che sono stati collocati nella memoria di transito tutti i dati per una riga di stampa, la riga di dati viene stampata.

Viene usato un indicatore di posizione di memoria di transito per mantenere traccia della successiva posizione disponibile ove può essere posizionato nella memoria di transito, un carattere (od una cifra). Il trasferimento dell'indicatore di posizione della memoria di transito prima, durante e dopo il trasferimento di una espressione nella memoria di transito, dipende dal tipo di espressione che si deve stampare e dal delimitatore che la segue nella istruzione PRINT.

La posizione dell'indicatore della posizione della memoria di transito può essere avanzata con l'uso della funzione TAB. Vedi più oltre gli esempi di istruzioni PRINT.

Gli esempi seguenti mostreranno come diversi valori aritmetici vengono stampati o visualizzati. Il simbolo $\emptyset$ rappresenta un carattere di spazio vuoto che compare sempre nella posizione occupata dal segno di un numero positivo.

Valore fornito	Valore stampato
0123	$\emptyset$ 123
00123	$\emptyset$ 123
-1234.5	-1234.5
135999999999	$\emptyset$ 135999999999

Note relative a PRINT

Se non c'è una stampatrice collegata al 5100, l'elaborazione della istruzione PRINT FLP interrompe il programma a meno che la opzione P = D (stampatrice = schermo video) sia stata specificata nel comando RUN.

Esempio

Gli esempi seguenti mostreranno le emissioni che si ottengono da diverse istruzioni PRINT:

Istruzioni	Emissione
10 PRINT FLP, 'A', 'B'	A - 17 spazi vuoti — B
20 PRINT 'A'; 'B'	AB
30 LET A\$ = 'B'	
40 PRINT FLP, 'A' A\$	AB BA - 17 spazi vuoti — BB
50 PRINT FLP, A\$ 'A' ,A\$;A\$	BA
60 PRINT A\$; 'A'	
70 LET A\$=""	
80 PRINT FPL, 'A';A\$;'A'	AA
90 PRINT 'A'; ';'A'	A&A
100 PRINT FLP	
110 PRINT 'NAME' TAB (30) 'ADDRESS'	NAME — 25 spazi vuoti — ADDRESS
120 END	

Conversione dei Valori di Riferimento dei Dati

Quando il dato cui ci si riferisce è un valore di carattere, i caratteri in esso contenuti vengono stampati nella riga, sostituendo i propri caratteri nelle specifiche di formato, ai segni (+, -), al segno di libbra (α), o al punto decimale e | | | |.

Se il valore di un carattere stampato è più corto della specifica del suo formato, vengono sistemati alla sua destra degli spazi vuoti riempitivi. Se il valore di un carattere stampato è più lungo della specifica del suo formato, viene troncato a destra. Una costante di carattere che non contenga caratteri (spazio) determina il riempimento di tutta la specifica di formato con tutti spazi vuoti.

Una espressione aritmetica viene convertita in conformità con la sua specifica di formato così come viene ora spiegato:

- Se la specifica di formato contiene un segno positivo ed il valore dell'espressione è positivo, viene stampato nella riga un segno positivo.
- Se la specifica di formato contiene un segno positivo ed il valore dell'espressione è negativa, viene stampato nella riga un segno negativo.
- Se la specifica di formato contiene un segno negativo ed il valore dell'espressione è positivo, viene stampato nella riga uno spazio vuoto.
- Se la specifica di formato contiene un segno negativo ed il valore dell'espressione è negativo, viene stampato nella riga un segno negativo.
- Se la specifica di formato non contiene un segno ed il valore dell'espressione è negativo, verranno stampati il segno negativo ed il numero negativo a patto che la specifica di formato sia lunga abbastanza per contenere sia il numero che il segno. Se la specifica di formato non è lunga abbastanza, vengono stampati nella riga alcuni asterischi al posto del numero negativo. Vengono inoltre stampati nella riga degli asterischi al posto di valori positivi che siano troppo lunghi.
- Il valore dell'espressione viene convertito in conformità con il tipo della sua specifica di formato così come spiegato:

Formato I: Il valore della espressione viene convertito in un numero intero, arrotondando ogni frazione.

Formato F: Il valore della espressione viene convertito in un numero in virgola mobile, sia arrotondando il valore oppure allungandolo con degli zeri e regolando l'esponente in conformità con la specifica di formato.

Formato E: Il valore della espressione viene convertito in un numero in virgola mobile, arrotondando il valore oppure allungandolo con degli spazi in conformità con la specifica di formato.

Se la lunghezza della parte intera del valore dell'espressione aritmetica è minore od uguale alla lunghezza della parte intera della specifica di formato, il valore della espressione viene stampato, aggiustato a destra, e riempito con degli spazi vuoti nella riga. Se la lunghezza della parte intera della specifica di formato è minore della lunghezza della parte intera del valore dell'espressione, vengono stampati nella riga degli asterischi al posto del valore dell'espressione.

Specifiche di Formato

Per ogni ricorrenza del segno di libbra (#) in una istruzione "maschera di stampa", viene riservato uno spazio singolo nella riga di video o di stampa per un carattere dell'espressione corrispondente nelle istruzioni PRINT USING o MAT PRINT USING collegate. Il segno di libbra rappresenta un dato aritmetico od un dato di carattere. Per dati aritmetici, punti decimali e segni di più e meno come i caratteri nella descrizione di formato generale, vengono stampati così come vengono immessi a patto che i valori siano adatti ai segni specificati. (Vedi Conversazione dei Valori di Riferimento dei Dati per una discussione della Visualizzazione/stampa dei segni in una specifica "maschera di stampa"). Per i dati di carattere, la stringa di caratteri trascurerà qualsiasi descrittore di formato.

I diversi tipi di specifiche di formato sono:

- Formato carattere — Uno o più caratteri, del tipo ;

####

- Formato I: (formato intero) — Un segno facoltativo, seguito da uno o più caratteri #, del tipo:

[+ -] # [#]

- Formato F (formato in virgola fissa) — Un segno facoltativo seguito da:

1. Nessun carattere #, un punto decimale, uno o più caratteri #, del tipo:

.# [###]

2. Uno o più caratteri #, un punto decimale seguito da nessun carattere #, del tipo:

[###].

3. Uno o più caratteri #, un punto decimale ed uno o più caratteri #, del tipo:

[+ -] [#]# [#] [#] . . .

- Formato E (formato esponenziale) — Il formato I oppure il formato F (precedentemente descritto) seguito da quattro caratteri (||||).

Le seguenti regole definiscono l'inizio di una specifica di formato:

1. Si incontra un carattere # ed il carattere precedente non è un carattere #, un punto decimale, un segno positivo o negativo.
2. Si incontra un segno positivo o negativo, seguito da:
 - a. Un carattere #, oppure
 - b. Un punto decimale che precede un carattere #.
3. Si incontra un punto decimale seguito da un carattere # e:
 - a. Il carattere precedente non è un carattere #, un segno positivo o un segno negativo, oppure
 - b. La serie di caratteri precedenti è una specifica di formato F.

Le seguenti regole definiscono la fine di una specifica di formato che è stata iniziata:

1. Si incontra un carattere # e:
 - a. Il carattere seguente non è un carattere # oppure
 - b. Il carattere seguente non è un punto decimale, oppure
 - c. Il carattere seguente è un punto decimale ed un punto decimale è stato già incontrato, oppure
 - d. I seguenti quattro caratteri consecutivi non sono caratteri (||||).
2. Si incontra un punto decimale e:
 - a. Il carattere seguente non è un carattere #, oppure
 - b. Il carattere seguente è un altro punto decimale, oppure
 - c. I seguenti quattro caratteri consecutivi non sono caratteri (||||).

Saranno mostrati qui oltre alcuni esempi di espressioni e il modo in cui essi vengono visualizzati o stampati sotto diverse specifiche di formato.

Specifica di Formato	Valore della Espressione	Formato Visualizzato o stampato
#####	'APPLES'	APPLE
####	-123	-123
###	123	123
###	12	12
##.##	1.23	1.23
##.##	123	*****
##.##	1.23	1.23
##.##	1.23456	1.23
##.##	.123	1.23

###	12.345	12.35
###	123	123E+00
###	12.3	123E-01
##.##	.1234	123E-03
##.##	123	12.30E+01
##.##	1.23	12.30E-01
##.##	.1234	12.34E-02
##.##	1234	12.34E+02

Durante l'emissione, le espressioni specificate vengono visualizzate o stampate iniziando dal punto della riga in cui le precedenti istruzioni PRINT o PRINT USING sono finite. Se questa è la prima istruzione PRINT o PRINT USING, l'emissione inizia su una nuova riga. Se l'istruzione PRINT USING contiene espressioni che superano, nella "maschera di stampa", le specifiche di formato, l'emissione viene controllata dal delimitatore che segue le espressioni. Se il delimitatore è una virgola, l'attuale riga viene visualizzata o stampata e le espressioni rimanenti vengono preparate per la stampa in conformità con l'inizio della maschera di stampa, e l'emissione inizia su una nuova riga. Se il delimitatore è un punto e virgola, le espressioni che restano dopo la fine della specifica "maschera di stampa" vengono preparate per la stampa in conformità con l'inizio della "maschera di stampa" stessa ma l'emissione prosegue sulla stessa riga.

Il delimitatore che segue l'ultimo valore dell'espressione comando la stampa alla fine delle espressioni. Se il delimitatore è una virgola od uno spazio vuoto, la corrente riga viene visualizzata o stampata e l'emissione successiva inizierà su una nuova riga. Se il delimitatore è un punto e virgola, la corrente riga non viene visualizzata o stampata e la successiva emissione verrà aggiunta alla corrente riga.

Se si raggiunge la fine della specifica "maschera di stampa" dopo che si è trovato l'ultimo valore dell'espressione o la prima specifica di formato non usata, la preparazione per la stampa viene arrestata.

Note Relative a PRINT USING

- Il numero delle "maschere di stampa" permesse in un programma BASIC è limitato unicamente dalla memoria disponibile.
- Se l'elenco di emissione della istruzione PRINT USING contiene almeno un elemento, dovrà esserci almeno una specifica di formato nella corrispondente "maschera di stampa".
- Se non vi è una stampatrice collegata al 5100, l'elaborazione della istruzione PRINT USING FLP arresta il programma, a meno che sia stato specificato nel comando RUN l'opzione P=D (stampatrice = schermo video).
- Le istruzioni "maschera di stampa" sono non eseguibili e possono venir sistemate in ogni punto del programma BASIC, sia prima che dopo le istruzioni PRINT USING che ne fanno riferimento.

Esempio

Il seguente esempio mostra l'elaborazione di una istruzione PRINT USING
in cui A=342 e B=42.02

```
100: RATE OF LOSS ##### EQUALS #####.## POUNDS  
110 PRINT USING 100, A, B
```

L'emissione è:

```
RATE OF LOSS 342 EQUALS 42.02 POUNDS
```

$$\text{PUT rifer. di file, } \left\{ \begin{array}{l} \text{espres. arit.} \\ \text{espres. di carat.} \end{array} \right\} \left[, \left\{ \begin{array}{l} \text{espres. arit} \\ \text{espres. di carat.} \end{array} \right\} \right] \dots$$

PUT

L'istruzione PUT permette di scrivere i valori delle variabili specificate in un particolare file. Il file deve precedentemente esser stato aperto con una OPEN. La sintassi dell'istruzione PUT è del tipo:

$$\text{PUT rifer. di file, } \left\{ \begin{array}{l} \text{espres. arit.} \\ \text{espres. di carat.} \end{array} \right\} \left[, \left\{ \begin{array}{l} \text{espres. arit} \\ \text{espres. di carat.} \end{array} \right\} \right] \dots$$

dove:

riferimento di file va da FL0 a FL9 e serve per identificare il file nel quale verranno scritti i valori delle variabili.

espr. sono riferimenti di funzione o riferimenti di insiemi indicizzati che rappresentano i valori che devono essere scritti. E' necessario un solo valore. Se ne viene specificato più di uno, i valori vanno tra loro separati con virgole.

Quando viene elaborata una istruzione PUT, il valore di riferimento scalare specificato viene immesso da sinistra verso destra nella memoria di transito del file specificato, cominciando dall'attuale posizione. Il file viene scritto sequenzialmente, in maniera tale che il primo valore immesso con l'istruzione PUT sia il primo valore che il file assegna quando al file stesso si fa riferimento in una istruzione GET. Quando la memoria di transito è piena, il contenuto viene trasferito come un record. I record del nastro sono lunghi 512 bytes. In ciascun record vengono usati il maggior numero di bytes possibile.

Un file può essere attivato unicamente con una OPEN e viene disattivato con una CLOSE oppure alla fine dell'elaborazione di un programma.

Note Relative a PUT

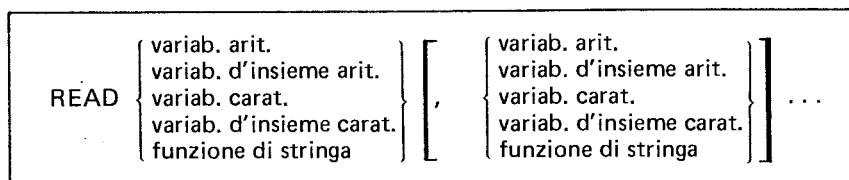
- Un file correntemente attivato come un file di immissione non può essere specificato in una PUT. Deve essere prima chiuso, poi riaperto per l'emissione, oppure si deve emettere una istruzione RESET END.
- Se lo spazio di memoria che è nel file di emissione viene esaurito prima che siano stati trasferiti nel file di emissione tutti i valori, finisce l'elaborazione del programma.
- Una istruzione PUT che si riferisce ad un file correntemente chiuso determina la fine dell'elaborazione del programma.

Esempio

Viene mostrato qui oltre un esempio di istruzione PUT:

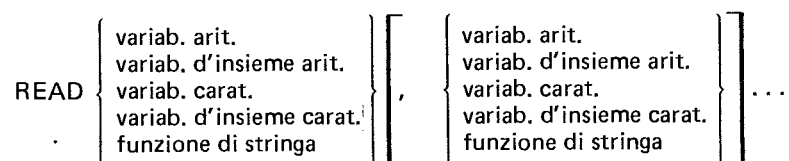
20 PUT FL1, Z3, 5*X-7,A,D\$,9.005

In questo esempio i valori specificati saranno scritti nel file FL1.



READ

L'istruzione READ permette di assegnare dei valori a variabili ed elementi di insiemi prelevandoli dalla tabella di dati interni creata mediante istruzioni DATA. La sintassi della istruzione READ è la seguente:



dove:

variabile sono variabili semplici aritmetiche o di carattere, sottostringhe, oppure riferimenti d'indice ad un unico elemento di insieme. E' necessaria almeno una variabile.

Quando viene elaborata una istruzione, i valori vengono progressivamente assegnati dalla tabella dei dati alle variabili che sono nella READ da sinistra verso destra, iniziando dall'attuale posizione dell'indicatore della tabella dei dati. L'indicatore della tabella dei dati lo si può far tornare indietro all'inizio della tabella mediante una istruzione RESTORE.

Gli indici delle variabili degli insiemi che si trovano nelle istruzioni READ vengono valutati allorchè li si incontra; così, una variabile assegnata in una istruzione READ la si può usare in seguito come indice di un'altra variabile nella stessa istruzione.

Note Relative a READ

- Ciascun valore letto dalla tabella dei dati deve essere dello stesso tipo (di carattere o aritmetico) della variabile alla quale viene assegnato.
- Se nel programma non ci sono istruzioni DATA, la READ determina un errore.
- Se la tabella dei dati è ormai vuota, ma restano ancora nella istruzione READ delle variabili da assegnare, si determina un messaggio di errore.
- Una istruzione READ che in un programma viene assegnata ad un tasto funzionale, causerà un errore.

Esempio

Gli esempi che seguono mostreranno l'elaborazione di istruzioni READ:

```
10 DATA 'JONES', 15.00, 'SMITH', 20.50
```

```
20 READ A$,A1,B$,B1
```

```
30 DATA 1,2,3,4,5,6
```

```
40 READ A,B,C,X(A), X(B), X(C)
```

Dopo l'elaborazione di queste istruzioni, le variabili carattere A\$ e B\$ conterranno le stringhe di carattere JONES e SMITH, rispettivamente, ciascuna riempita alla propria destra di spazi vuoti fino ad una lunghezza di 18 caratteri. Le variabili aritmetiche A1 e B1 conterranno rispettivamente i valori decimali 15.00 e 20.50. Le variabili aritmetiche A, B e C conterranno rispettivamente i valori interi 1, 2 e 3; inoltre i primi tre elementi dell'insieme monodimensionale X conterranno i valori interi 4, 5 e 6 rispettivamente.

REM [commento]

REM

L'istruzione REM permette di inserire notazioni o commenti in una lista di programma BASIC. La sintassi dell'istruzione REM è del tipo:

REM [commento]

dove:

commento è costituito da uno o più caratteri, ed è facoltativo.

L'istruzione REM è non elaborabile. Compare nella lista di programma, ma non ha alcun effetto sull'elaborazione del programma.

Note relative a REM

- Una istruzione REM la si può usare in ogni punto di un programma BASIC.
- Vedi anche tasti funzionali, Capitolo 2.

Esempio

Qui oltre è mostrato un esempio di istruzione REM:

10 REM THIS PROGRAM DETERMINES THE COST PER UNIT

RESET

Una istruzione RESET permette di reimpostare un file di immissione al suo inizio oppure un file di emissione al suo inizio od alla sua fine. La sintassi della istruzione RESET è la seguente:

RESET riferim. di file [END] , riferim. di file [END] ...

dove:

riferimento di file va da FL0 a FL9 per identificare il file che si deve reimpostare. Il file deve essere stato precedentemente aperto mediante una OPEN. Una sola istruzione RESET è necessaria per un solo file. La immissione di END è facoltativa e verrà descritta più volte.

Allorchè viene elaborata un'istruzione RESET, il file specificato viene reimpostato in modo che i successivi riferimenti ad esso fatti da istruzioni GET o PUT, si riferiranno al primo articolo del file. Quando un file viene aperto da una istruzione OPEN, il primo articolo del file diventa automaticamente accessibile.

Allorchè viene specificato RESET END, il file viene chiuso e riaperto per l'emissione. Il file viene reimpostato onde poter scrivere nuovi dati incominciando dalla fine dei dati già esistenti nel file.

Se durante l'elaborazione dello stesso programma un file lo si deve usare per l'immissione mentre si trova aperto per l'emissione, lo si deve chiudere e riaprire con delle istruzioni CLOSE e OPEN. La stessa cosa vale per files che, nello stesso programma, devono essere usati per emissione mentre si trovano aperti per l'emissione.

Note relative a RESET

Se un file specificato in una istruzione RESET non è correntemente attivo, il riferimento fatto dalla istruzione RESET viene ignorato.

Esempio

Nell'esempio che segue l'istruzione RESET (numero 100) reimposta il file FL6 al suo inizio. La GET (numero 110) Legge allora i primi tre valori del file FL6 come variabili A, B e C rispettivamente.

90 GET FL6, X,Y,Z

100 RESET FL6

110 GET FL6, A,B,C

RESTORE [commento]

RESTORE

L'istruzione RESTORE permette di ricominciare l'assegnazione di valori a partire dal primo articolo che si trova nella prima istruzione DATA (vedi DATA) del programma non appena verrà elaborata la successiva istruzione READ. La sintassi della istruzione RESTORE è la seguente:

RESTORE [commento]

dove:

commento è costituito da uno o più caratteri.

L'istruzione RESTORE riporta l'indicatore della tabella di dati interni dalla sua attuale posizione all'inizio della tabella. Il commento, facoltativo, è costituito da una stringa di caratteri che non influisce sull'elaborazione della istruzione stessa.

Note relative a RESTORE

- Una istruzione RESTORE in un programma che non contiene istruzioni DATA viene ignorata.
- Una istruzione RESTORE per un indicatore di tabella di dati già ripristinato alla sua posizione iniziale viene ignorata.

Esempio

Dopo l'elaborazione delle seguenti istruzioni, le variabili A e C avranno ciascuna un valore di 1 e B e D avranno ciascuna un valore di 2.

10 DATA 1,2

20 READ A,B

30 RESTORE

40 READ C,D

RETURN

Per l'uso della istruzione RETURN nella creazione di sottoprogrammi vedi le istruzioni GOSUB e RETURN. Vedi inoltre i Tasti funzionali.

Per l'uso della istruzione RETURN con una definizione di funzione multiriga, vedi le istruzioni DEF, FNEND, RETURN.

STOP [commento]

STOP

L'istruzione STOP permette di finire l'elaborazione del programma. La sintassi della istruzione STOP è la seguente:

STOP [commento]

dove:

commento è costituito da uno o più caratteri.

Allorchè si incontra una istruzione STOP durante l'elaborazione di un programma, questa determina la chiusura di tutti i files aperti e determina la fine dell'elaborazione. Il sistema di agire della STOP è identico a quello della END.

Note relative a STOP

Una istruzione STOP può comparire in ogni punto di un programma BASIC.

Esempio

Il seguente è un esempio di istruzione STOP:

110 STOP

USE	$\left\{ \begin{array}{l} \text{variab. arit.} \\ \text{variab. di carat.} \\ \text{insieme arit. (righe [,colonne])} \\ \text{insieme di carat. (righe [,colonne])} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{variab. arit.} \\ \text{variab. di carat.} \\ \text{insieme arit. (righe [,colonne])} \\ \text{insieme di carat. (righe [,colonne])} \end{array} \right\} \dots \right]$
-----	--

USE

L'istruzione USE permette di specificare le variabili che si devono assegnare all'area di memoria comune. Quest'area di memoria comune conserva quelle variabili specificate e le trasferisce da un programma BASIC al successivo. I valori di queste variabili non vengono modificati quando un programma esegue l'istruzione CHAIN (vedi). La sintassi della istruzione USE è la seguente:

$$\text{USE} \left\{ \begin{array}{l} \text{variab. arit.} \\ \text{variab. di carat.} \\ \text{insieme arit. (righe [,colonne])} \\ \text{insieme di carat. (righe [,colonne])} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{variab. arit.} \\ \text{variab. di carat.} \\ \text{insieme arit. (righe [,colonne])} \\ \text{insieme di carat. (righe [,colonne])} \end{array} \right\} \dots \right]$$

dove:

variabile od insieme è una variabile numerica o di carattere oppure un insieme che si deve assegnare all'area di memoria comune.

righe e colonne sono necessarie solamente per gli insiemi. Queste costanti definite da numeri interi privi di segno e diversi da zero specificano il numero di righe e colonne che forniscono le dimensioni dell'insieme assegnato all'area di memoria comune.

Note relative a USE

- Se una variabile, in una istruzione USE viene definita come un insieme, non deve venir inclusa in una istruzione DIM (vedi DIM).
- USE deve essere specificata in tutti i programmi concatenati sia in quelli di partenza che in quelli di arrivo.
- Se per una variabile vengono specificate una o due dimensioni, la variabile viene assegnata all'area di memoria comune come un insieme fornito di quelle dimensioni.
- In un programma si può immettere solamente una istruzione USE e deve essere la prima istruzione elaborata che contenga un riferimento di variabile. Così, ad esempio, solo le istruzioni DEF, GOSUB incondizionato e GOTO possono essere elaborate prima della USE.
- Dati variabili vengono assegnati all'area di memoria comune nella stessa sequenza in cui sono stati immessi nella istruzione USE. Così, i valori dei dati diventano inattendibili se i tipi di dati differiscono da un programma concatenato al successivo.

Esempio

Così, in questo esempio:

0290 USE A(9,4), T, C

un insieme (A) con nove righe e quattro colonne è assegnato all'area di memoria comune. Inoltre anche le variabili scalari T e C vengono assegnate all'area di memoria comune.

OPERAZIONI DI MATRICI

Nel 5100 un insieme può contenere sia dati numerici che di caratteri, mentre una matrice può contenere solamente dati numerici. prima di venir usati in istruzioni MAT, insiemi e matrici devono essere stati precedentemente definiti, implicitamente oppure esplicitamente in un'istruzione DIM o USE.

ISTRUZIONI DI ASSEGNAZIONE MAT

Le istruzioni di assegnazione MAT permettono di assegnare valori agli elementi di un insieme. Il valore assegnato può provenire da una qualsiasi delle seguenti espressioni di insieme. Ciascuna espressione di insieme sarà discussa in dettaglio più oltre.

Espressione di insieme	Significato
(e)	Valore scalare
A	Insieme semplice
A + B	Somma di matrici
A - B	Sottrazione di matrici
A * B	Moltiplicazione di matrici
(e) * A	Moltiplicazione scalare
IDN	Funzione d'identità
INV (A)	Funzione inversa
TRN (A)	Funzione trasposta

MAT nome di insieme [(righe [,colonne])] = (espressione scalare)

ASSEGNAZIONE MAT (VALORE SCALARE)

Questa istruzione permette di assegnare un determinato valore scalare a ciascun elemento di un insieme. La sintassi di questa specifica è la seguente:

MAT nome di insieme [(righe [,colonne])] = (espressione scalare)

dove:

nome di insieme è il nome dell'insieme che riceve i valori.

righe, colonne sono le specifiche delle dimensioni dell'insieme (vedi Ridimensionamento degli Insiemi).

espressione scalare è il valore che deve essere assegnato.

Quando questa istruzione viene elaborata, l'espressione scalare fra parentesi viene valutata e ciascun elemento dell'insieme specificato nella istruzione viene impostato a quel valore.

Se vengono incluse le specifiche delle dimensioni, la parte intera troncata di ciascuna espressione che rappresenta righe e colonne viene usata per ridimensionare l'insieme prima che il valore scalare venga valutato ed assegnato a ciascun elemento dell'insieme.

Per insiemi di caratteri, se l'espressione è inferiore a 18 caratteri, questa viene riempita a destra con degli spazi vuoti fino alla lunghezza di 18 caratteri prima di venir assegnata. Se nell'espressione vi sono più di 18 caratteri, vengono assegnati agli elementi dell'insieme solamente i primi 18 caratteri.

Note relative a MAT (Scalare)

- L'espressione scalare a destra del segno di uguale deve essere dello stesso tipo (aritmetico o di carattere) dell'insieme al quale viene assegnata .
- Se le specifiche di ridimensionamento seguono il nome dell'insieme, bisogna seguire le regole così come sono state enunciate nel Capitolo 3 sotto il titolo Ridimensionamento degli Insiemi.

L'esempio che segue mostra l'elaborazione di una istruzione di assegnazione (scalare) MAT:

40 DIM Y (3,3)

50 MAT Y(2,2) = (1)

I valori risultanti sono:

Diagram illustrating the resulting matrix Y (3x3) after the assignment operation. The matrix is shown with its dimensions and the values assigned (1) at specific positions (2,2).

1		1
1		1

Labels: Colonne (Columns), Righe (Rows), Y risulta (Y results).

MAT nome dell'insieme [(righe, colonne)] = nome dell'insieme

ASSEGNAZIONE MAT (SEMPLICE)

L'istruzione Assegnazione MAT semplice permette di assegnare gli _____ elementi di un insieme ad un altro insieme. La sintassi di questa istruzione è:

MAT nome dell'insieme (righe, colonne) = nome dell'insieme

dove:

nome dell'insieme è il nome dell'insieme.

righe, colonne sono le specifiche di ridimensionamento per il primo insieme (vedi Ridimensionamento degli Insiemi, Capitolo 3).

Ciascun elemento dell'insieme specificato a destra del segno di uguale viene assegnato all'elemento corrispondente dell'insieme specificato alla sinistra del segno di uguale.

Se le specifiche di ridimensionamento seguono il nome alla sinistra del segno di uguale, la parte intera troncata di ciascun valore dell'espressione nelle righe, colonne viene usato per ridimensionare l'insieme prima che gli vengano assegnati i valori.

Note relative a MAT (Semplice)

- Ambedue gli insiemi specificati debbono essere dello stesso tipo (aritmetici o di caratteri).
- Ambedue gli insiemi, specificati nella istruzione di assegnazione di insieme, debbono possedere identiche dimensioni (dopo il ridimensionamento, se c'è).
- Se vengono incluse le specifiche di ridimensionamento, bisogna seguire le regole descritte sotto la voce Ridimensionamento degli Insiemi.

Esempio

L'esempio seguente mostra l'elaborazione di una istruzione di assegnazione (semplice) MAT:

20 DIM A(2,2),B(2,2)

.
.
.

100 MAT A = B

Qui oltre sono rappresentati i valori risultanti:

$$\text{Se } B = \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array} \quad \text{e } A = \begin{array}{|c|c|} \hline e & f \\ \hline g & h \\ \hline \end{array}$$

dopo la riga 100 si avrà:

$$B = \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array} \quad A = \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array}$$

MAT nome di matrice [(righe [,colonne])] = nome di matrice $\left\{ \begin{smallmatrix} + \\ - \end{smallmatrix} \right\}$ nome di matrice

ASSEGNAZIONE MAT (ADDIZIONE E SOTTRAZIONE)

L'istruzione di Assegnazione MAT permette di sommare e sottrarre il contenuto di due matrici ed assegnare il risultato ad una terza matrice. La sintassi della istruzione è la seguente:

MAT nome di matrice [(righe [,colonne])] = nome di matrice $\left\{ \begin{smallmatrix} + \\ - \end{smallmatrix} \right\}$ nome di matrice

dove:

tutti i parametri della istruzione sono gli stessi di quelli già visti per altre istruzioni di assegnazione MAT.

Gli elementi corrispondenti delle matrici specificate a destra del segno di uguale vengono sommate o sottratte, così come indicato, ed il risultato della operazione viene assegnato ai corrispondenti elementi nella matrice specificata a sinistra del segno di uguale.

Se le specifiche di ridimensionamento seguono il nome della matrice alla sinistra del segno di uguale, la parte intera troncata, di ciascun valore di espressione nelle righe, colonne viene usato per ridimensionare la matrice prima che le vengano assegnati valori.

Note relative a MAT (Addizione e Sottrazione)

- Tutte e tre le matrici debbono essere numeriche.
- Tutte e tre le matrici specificate nella istruzione debbono possedere identiche dimensioni (dopo il ridimensionamento, se c'è).
- Se vengono incluse le specifiche di ridimensionamento, bisogna seguire le regole descritte sotto la voce Ridimensionamento degli Insiemi, Capitolo 3.

Esempio

L'esempio seguente mostra l'elaborazione di questa istruzione:

```
10 DIM X(3,2), Y(2,2), Z(2,2)
```

```
.
```

```
.
```

```
100 MAT X (2,2) = Y + Z
```

I valori che ne risultano sono:

Se $Y =$

a	b
c	d

 e $Z =$

e	f
g	h

, allora $X =$

a+e	b+f
c+g	d+h

M AT nome della matrice [(righe [,colonne])] = nome della matrice * nome della matrice

ASSEGNAZIONE MAT (MOLTIPLICAZIONE DI MATRICI)

Questa istruzione permette di svolgere la moltiplicazione matriciale tra due matrici numeriche ed assegnare il prodotto ad una terza. La sintassi di questa istruzione è la seguente:

MAT nome della matrice (righe [,colonne]) = nome della
matrice*nome della matrice

dove:

tutti i parametri della istruzione sono gli stessi di altre istruzioni di assegnazione MAT già esaminate.

In una moltiplicazione di matrici, una matrice (A) con dimensioni (p, m) ed una matrice (B) con dimensioni (m, n) forniscono una matrice prodotto (C) di dimensioni (p, n) così che per $i = 1, 2, \dots, p$ e per $j = 1, 2, \dots, n$:

$$C(i,j) = \sum_{k=1}^m A(i,k) * B(k,j)$$

Se le specifiche di ridimensionamento seguono il nome di matrice alla sinistra del segno di uguale, la parte intera troncata di ciascun valore di espressione nelle righe, colonne viene usato per ridimensionare la matrice prima che le vengano assegnati i valori.

Note relative a MAT (Moltiplicazione di Matrici)

- Tutte e tre le matrici specificate debbono essere numeriche.
 - Se la matrice specificata alla sinistra del segno di uguale è uguale ad una delle due matrici a destra del segno di uguale, si otterranno risultati sbagliati.
 - Tutte le seguenti regole, in cui: $A=B*C$ debbono essere soddisfatte (dopo ridimensionamento, se c'è).
1. Tutte e tre le matrici debbono essere bidimensionali.
 2. Il numero di colonne della seconda matrice (B) deve essere uguale al numero di righe della terza matrice (C).
 3. Il numero di righe della prima matrice (matrice prodotto A) deve essere uguale al numero di righe della seconda matrice (B).
 4. Il numero di colonne della prima matrice (matrice prodotto A) deve essere uguale al numero di colonne della terza matrice (C).
- Se vengono accluse le specifiche di ridimensionamento, si devono seguire le regole descritte sotto la voce Ridimensionamento degli Insiemi, Capitolo 3.

Esempio

L'esempio seguente mostra l'elaborazione dell'istruzione di Assegnazione MAT:

```
10 DIM X(2,2), Y(2,2), Z(2,2)
```

```
·  
·  
·
```

```
100 MAT Z = X * Y
```

I valori che ne risultano sono:

$$\text{Se } X = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \text{e } Y = \begin{bmatrix} e & f \\ g & h \end{bmatrix}, \text{ allora } Z = \begin{bmatrix} a*e+b*g & a*f+b*h \\ c*e+d*g & c*f+d*h \end{bmatrix}$$

MAT nome di matrice [(righe [,colonne])] = (espressione arit.) * nome di matrice

ASSEGNAZIONE MAT (MOLTIPLICAZIONE SCALARE)

Questa istruzione permette di moltiplicare gli elementi di una matrice numerica per il valore di una espressione aritmetica, ed assegnare i prodotti che ne risultano agli elementi di un'altra matrice numerica. La sintassi di questa istruzione è la seguente:

MAT nome di matrice [(righe [,colonne])] = (espressione aritm.)*nome di matrice

dove:

nome di matrice è il nome di una matrice numerica.

righe e colonne sono le specifiche di ridimensionamento (facoltative).

(espressione aritmetica) è una espressione aritmetica scalare, che deve essere chiusa in parentesi.

Viene valutata l'espressione scalare e ciascun elemento della matrice a destra del segno di uguale viene moltiplicato per il valore della espressione. Il risultato viene assegnato agli elementi corrispondenti della matrice che si trova a sinistra del segno di uguale.

Se le specifiche di ridimensionamento seguono il nome di matrice alla sinistra del segno di uguale, la parte intera troncata del valore di ogni espressione nelle righe, colonne viene usato per ridimensionare la matrice prima della moltiplicazione.

Note relative a MAT (Moltiplicazione Scalare)

- Ambedue le matrici specificate debbono essere numeriche.
- Ambedue le matrici specificate debbono possedere identiche dimensioni (dopo ridimensionamento, se c'è).
- Se vengono accluse le specifiche di ridimensionamento degli Insiemi, Capitolo 3.

Esempio

Il seguente esempio mostra l'elaborazione di una istruzione di assegnazione MAT:

```
20 DIM X(2,2), Y(2,2)
```

```
·  
·
```

```
100 MAT Y = (4) * X
```

I valori che ne risultano sono:

$$\text{se } X = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \text{ allora } Y = \begin{bmatrix} 4*a & 4*b \\ 4*c & 4*d \end{bmatrix}$$

ASSEGNAZIONE MAT (FUNZIONE D'IDENTITÀ)

Questa istruzione permette di far assumere ad una matrice numerica la forma di una matrice d'identità. La sintassi dell'istruzione è la seguente:

MAT nome di matrice [(righe,colonne)] = IDN

dove:

tutti i parametri sono gli stessi di quelli già descritti in altre funzioni di assegnazione, e IDN specifica la matrice identità.

A ciascun elemento della matrice specificata, per il quale i valori di ambedue gli indici sono uguali (elementi sulla diagonale maggiore), per esempio A(2,2) oppure A(3,3), viene assegnato il valore intero 1. A tutti gli altri elementi, per esempio A(2,3) o A(3,1) vengono assegnati dei valori 0.

Se le specifiche di ridimensionamento seguono il nome di matrice, la parte intera troncata di ciascun valore di espressione nelle righe, colonne viene usato per ridimensionare la matrice prima che vengano assegnati a ciascuno dei suoi elementi i valori 1 o 0.

Note relative a MAT (Funzione d'Identità)

- La matrice specificata deve essere numerica.
- La matrice numerica specificata deve essere una matrice quadrata; cioè il numero delle righe deve essere uguale al numero delle colonne (dopo ridimensionamento, se c'è).
- Se vengono accluse le specifiche di ridimensionamento, si devono seguire le regole descritte sotto la voce Ridimensionamento degli Insiemi, Capitolo 3.

Esempio

L'esempio seguente mostra l'elaborazione di una istruzione MAT (funzione d'identità):

50 DIM X(16)

.
.
.

60 MAT X(4,4) = IDN

I valori che ne risultano sono:

X risulta

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

$\text{MAT nome di matrice}[(\text{righe}, \text{colonne})] = \text{INV}(\text{nome di matrice})$

ASSEGNAZIONE MAT (FUNZIONE INVERSA)

Questa istruzione permette di assegnare l'inversa di una matrice ad un'altra. La sintassi della istruzione è la seguente:

$\text{MAT nome di matrice}[(\text{righe}, \text{colonne})] = \text{INV}(\text{nome di matrice})$

dove:

tutti i parametri sono gli stessi di quelli già visti per le precedenti istruzioni di assegnazione MAT; INV specifica la funzione inversa.

La matrice inversa della matrice specificata a destra del segno di uguale viene assegnata alla matrice specificata a sinistra del segno di uguale. Per la matrice quadrata A di dimensioni (n,n), la matrice inversa B, se esiste, è una matrice con identiche dimensioni così che:

$$A*B = B*A = I$$

in cui I è un amatrice d'identità.

Non tutte le matrici possiedono una loro inversa. La funzione del sistema DET (vedi funzioni del Sistema) può essere usata per determinare se una data matrice possiede una sua inversa. La matrice inversa della matrice A esiste se $\text{DET}(A) \neq 0$.

Se le specifiche di ridimensionamento seguono il nome della matrice posto alla sinistra del segno di uguale, la parte intera troncata del valore di ogni espressione nelle righe, colonne viene usato per ridimensionare la matrice prima che le vengano assegnati i valori.

Note relative a MAT (Funzione Inversa)

- Ambedue le matrici specificate debbono essere numeriche.
- Ambedue le matrici specificate debbono essere quadrate, ed ambedue debbono possedere dimensioni identiche (dopo ridimensionamento, se c'è).
- Il determinante lo si considera zero se il risultato è minore di $1\text{E-}20$.
- Se vengono accluse le specifiche di ridimensionamento, si devono seguire le regole descritte sotto la voce ridimensionamento degli Insiemi, Capitolo 3.

Esempio

L'esempio seguente mostra l'elaborazione di una istruzione MAT (inversa);

```
20 DIM X(2,2), Y(2,2)
.
.
.
80 IF DET (Y) = 0 GOTO 300
90 MAT X = INV (Y)
.
.
.
295 GOTO 310
200 PRINT 'SINGULAR MATRIX'
310 END
```

I valori che ne risultano sono:

Se Y è

1	1
1	2

, allora X è

2	-1
-1	1

MAT nome di insieme [(righe, colonne)] = TRN (nome di insieme)

ASSEGNAZIONE MAT (FUNZIONE TRANSPOSTA)

Questa istruzione permette di sostituire gli elementi di un insieme con la matrice transposta di un altro insieme. La sintassi della istruzione fa seguente:

MAT nome di insieme [(righe,colonne)] =TRN(nome di insieme)

dove:

tutti i parametri della istruzione sono gli stessi già visti per le precedenti istruzioni di assegnazione MAT e TRN specifica la funzione transposta.

La funzione transposta dell'insieme specificato a destra del segno di uguale viene assegnata all'insieme specificato a sinistra del segno di uguale. I valori presenti nella colonna Y di un insieme divengono i valori della riga Y dell'altro insieme.

Se le specifiche di ridimensionamento seguono il nome della matrice posto alla sinistra del segno di uguale, la parte intera troncata di ciascun valore dell'espressione nelle righe, colonne viene usato per ridimensionare la matrice prima che le vengano assegnati i valori.

Note relative a MAT (Funzione Transposta)

- Ambedue gli insiemi specificati debbono essere bidimensionali ed il numero delle righe di ciascun insieme deve essere uguale al numero delle colonne dell'altro (dopo ridimensionamento, se c'è); devono inoltre essere dello stesso tipo (numerico o di carattere).
- Lo stesso insieme non può essere usato da entrambe le parti del segno di uguale. Ciò determinerebbe un risultato errato.
- Se vengono accluse le specifiche di ridimensionamento, si devono seguire le regole descritte sotto la voce Ridimensionamento degli Insiemi, Capitolo 3.

Esempio

L'esempio seguente mostra l'elaborazione di una istruzione MAT (funzione transposta):

```
40 DIM A(3,2), B(2,3)
```

```
·  
·  
·
```

```
80 MAT B = TRN(A)
```

I valori che ne risultano sono:

Se A è

a	d
b	e
c	f

, allora B è

a	b	c
d	e	f

MAT rifer. di file, nome di insieme [(righe [,colonne])] [,nome di insieme [(righe [,colonne])]] ... [,EOF numero di riga]
--

MAT GET

L'istruzione MAT GET permette di assegnare un valore a ciascun elemento di un insieme prelevandolo da un determinato file senza doversi riferire a ciascun elemento individuale dell'insieme. Il file specificato deve essere stato già precedentemente aperto mediante una OPEN (vedi). E' inoltre possibile usare l'istruzione MAT GET per ridimensionare gli insiemi. La sintassi della istruzione è la seguente:

MAT rifer. di file, nome di insieme [(righe [,colonne])] [,nome di insieme [(righe [,colonne])]] ... [,EOF numero di riga]

dove:

riferimento di file va da FLO a FL9 e fa riferimento al file specificato nella OPEN.

nomi di insieme sono nomi di insiemi numerici o di caratteri.

righe, colonne sono le dimensioni degli insiemi nominati.

EOF numero di riga è il numero della istruzione alla quale il programma salterà quando e se sarà raggiunta la fine del file.

Quando viene elaborata una istruzione MAT GET, i valori del file vengono assegnati agli insiemi specificati riga per riga. Il file del riferimento deve essere stato precedentemente aperto per l'immissione mediante una OPEN. Il file viene posizionato al suo inizio alla prima istruzione MAT GET a meno che una o più operazioni GET siano state precedentemente elaborate. Successive istruzioni MAT GET relative allo stesso file determineranno l'assegnazione dei valori a partire dalla corrente posizione sul file.

Se le immissioni (facoltative) relative a righe e colonne seguono nella istruzione MAT GET i nomi dell'insieme, le parti intere troncate dei valori dell'espressione verranno usati per ridimensionare gli insiemi prima che i valori dei dati vengano assegnati prelevandoli dal file.

Un file può essere attivato unicamente mediante una OPEN e viene disattivato mediante una CLOSE oppure alla fine dell'elaborazione del programma.

Note Relative a MAT GET

- Un file attualmente attivato come un file di emissione non può venir specificato in una istruzione MAT GET. Deve prima venir chiuso e poi riaperto.
- Se il file di immissione viene vuotato prima che un determinato insieme sia stato riempito, l'elaborazione del programma ha termine a meno che venga specificato un numero di riga EOF.
- Se vengono accluse le specifiche di ridimensionamento, si devono seguire le regole descritte sotto la voce Ridimensionamento degli Insiemi, Capitolo 3.

Esempio

L'esempio seguente mostra una istruzione MAT GET:

```
0090 MAT GET FL2, A, B (5,10), Z (4,5), EOF 210
```

In questo esempio, l'insieme A, l'insieme B (ridimensionato a 5 righe e 10 colonne) e l'insieme Z (ridimensionato a 4 righe e 5 colonne) riceveranno i valori al file FL2. Quando sarà raggiunta la fine del file, il comando verrà trasferito alla riga 210.

MAT INPUT nome di insieme [(righe [,colonne])] [,nome di insieme [(righe [,colonne])]]

MAT INPUT

L'istruzione **MAT INPUT** permette di assegnare valori agli elementi di insiemi tramite tastiera senza dover specificare individualmente ciascun elemento dell'insieme. L'istruzione **MAT INPUT** può inoltre essere usata per ridimensionare gli insiemi. La sintassi della istruzione è la seguente:

MAT INPUT nome di insieme [(righe [,colonne])] [,nome di insieme [(righe [,colonne])]]

dove:

nomi di insieme sono insiemi numerici o di caratteri.

righe, colonne sono le dimensioni degli insiemi che vengono nominati. E' necessario unicamente il nome di un insieme; righe e colonne sono facoltative.

Quando viene elaborata una istruzione **MAT INPUT**, questa determina la visualizzazione di un punto interrogativo lampeggiante sulla riga 1 dello schermo video e l'elaborazione del programma viene interrotta. Bisogna allora immettere un elenco di valori che saranno assegnati riga per riga agli elementi degli insiemi specificati.

Dopo aver battuto alla tastiera ciascuna riga di immissione (al massimo 64 caratteri) la si deve elaborare battendo **EXECUTE**. Se non è stato ancora riempito l'insieme specificato viene ulteriormente visualizzato il segno di punto interrogativo per indicare che è necessaria altra immissione.

Costanti carattere le quali contengano meno di 18 caratteri vengono riempite con dei caratteri di spazi vuoti alla loro destra fino a raggiungere una lunghezza di 18 caratteri prima di venir assegnate all'elemento di insieme. Le costanti caratteri che contengono più di 18 caratteri, prima di venir assegnate, vengono troncate a destra a 18 caratteri. Costanti carattere che non contengono alcun carattere (nulla) vengono assegnate come se contenessero 18 caratteri spazi vuoti.

Note relative a MAT INPUT

- Ciascun valore immesso deve essere dello stesso tipo (numerico o di carattere) dell'elemento dell'insieme corrispondente.
- Ciascun valore immesso deve essere separato dal successivo mediante una virgola.
- L'ultimo valore immesso su una riga può essere seguito da una virgola o da uno spazio vuoto.
- Se il numero di elementi di dati immessi supera il numero richiesto degli elementi dell'insieme a cui si riferiscono, gli elementi di dati in eccesso vengono ignorati.
- Spazi vuoti all'interno degli articoli di dati numerici vengono ignorati.

- Dati di caratteri devono essere posti tra apici.
- Se vengono accluse le specifiche di ridimensionamento, si devono seguire le regole descritte sotto la voce Ridimensionamento degli Insiemi, Capitolo 3.

Esempio

Il seguente esempio mostra una istruzione MAT INPUT:

```
0100 MAT INPUT A, A$,B(3,6)
```

Prima dell'elaborazione di questa istruzione, bisogna immettere i valori degli insiemi A, A\$ e B (ridimensionato a 3 righe, 6 colonne).

$$\text{MAT PRINT [rifer. di file,] nome di insieme } \left\{ \begin{matrix} ; \\ , \end{matrix} \right\} \text{ nome di insieme } \dots \left\{ \begin{matrix} ; \\ , \end{matrix} \right\}$$

MAT PRINT

L'istruzione MAT PRINT determina la visualizzazione o la stampa degli elementi di un insieme numerico o di caratteri specificato, senza riferirsi individualmente a ciascun elemento dell'insieme. Quando si fa uso della istruzione MAT PRINT, il formato di tutti i valori visualizzati o stampati è standardizzato; purtuttavia si può agire sulla spaziatura tra i valori appartenenti alla stessa riga. La sintassi della istruzione è la seguente:

MAT PRINT [riferimento di file,] nome di insieme $\left\{ \begin{matrix} ; \\ , \end{matrix} \right\}$ nome di insieme $\dots \left\{ \begin{matrix} ; \\ , \end{matrix} \right\}$

dove:

riferimento di file è FLP (per la stampatrice) oppure da FLO a FL9 per il file logico. Ai file da FLO a FL9 deve aver fatto riferimento in precedenza con una istruzione OPEN.

nomi di insieme sono insiemi numerici o di caratteri, e la virgola o il punto e virgola sono delimitatori che specificano la spaziatura tra gli elementi dell'insieme. Si deve specificare almeno un nome di insieme.

Quando viene elaborata una istruzione MAT PRINT, ciascun elemento, di insieme viene convertito nel formato di emissione specificato e visualizzato. Vedi "Conversione dei Valori di Riferimento dei Dati" nel paragrafo della istruzione PRINT.

Ciascun insieme viene visualizzato per righe; la prima riga di ciascun insieme comincia nella posizione iniziale di una nuova riga ed è separata dalla precedente riga da due righe di spazi vuoti. Le restanti righe di ciascun insieme cominciano all'altezza di una nuova riga e sono separate dalla riga precedente da una riga di spazi vuoti. Dopo che è stato visualizzato o stampato ciascun elemento dell'insieme, la posizione iniziale dell'elemento successivo viene determinata dal carattere delimitatore, come descritto nel paragrafo "Spaziatura dei Valori Visualizzati".

Spaziatura dei Valori Visualizzati

Il valore convertito di ciascun elemento dell'insieme viene visualizzato nella propria zona di stampa. Una zona di stampa può essere sia completa (18 posizioni), sia compatta (lunghezza variabile) così come è stato specificato dal delimitatore che segue il riferimento di insieme.

Zone di Stampa Complete (Virgole come Delimitatore): Ciascun elemento di insieme sfrutta una zona di stampa completa di 18 posizioni di caratteri.

Zone di Stampa Compatte (Punto e Virgola come Delimitatore): Per gli insiemi numerici la lunghezza della zona di stampa compatta viene determinata dalla lunghezza di ciascun valore convertito, inclusi il segno, le cifre, il punto decimale e l'esponente così come mostrato in Figura 11.

Lunghezza dell'articolo di dati Convertito (Carattere)	Lunghezza della Zona di Stampa Compatta (Carattere)	Esempio (b rappresenta uno spazio vuoto)
2 - 4	6	b7.3bb
5 - 7	9	b17.357bb
8 - 10	12	-45.63927bbb
11 - 13	15	b1.73579E-23bbb
14 - 17	18	-8.922704931156bbb

Figura 11. Lunghezza delle zone di stampa compatta per elementi di insiemi autentici.

Operazioni di Visualizzazione/Stampa di Righe

Quando i valori degli elementi di insieme vengono trasferiti allo schermo video ad alla memoria di transito della stampatrice, viene gestito un indicatore interno di posizione di riga per conservare traccia della successiva posizione disponibile in cui si possa collocare un carattere (o cifra).

Se il delimitatore di insieme è una virgola, l'indicatore di posizione di riga viene spostato al di là di qualsiasi restante posizione della zona di stampa completa dopo il trasferimento del valore dell'elemento allo schermo video. Vedi "Operazione di Stampa della Memoria di Transito di Riga", nel paragrafo PRINT.

Note Relative a MAT PRINT

- In una istruzione MAT PRINT non sono permessi delimitatori vuoti fatta eccezione come delimitatori finali.
- Se il 5100 non possiede allacciata una stampatrice, la specifica MAT PRINT FLP arresta il programma a meno che l'opzione P = D (stampatrice = schermo) sia stata specificata nel comando RUN.

Esempio

L'esempio seguente mostra l'elaborazione di una istruzione MAT PRINT:

```
10 DIM A(3,4), A$(12), C(2,2)
```

```
20 MAT READ A,A$
```

```
30 DATA 1,2,3,4,5,6,7,8,9,10,11,12
```

```
40 DATA 'A','B','C','D','E','F','G','H','I','J','K','L',
```

```
50 MAT C = (1)
```

```
60 MAT PRINT A
```

```
70 MAT PRINT A$;
```

```
80 MAT PRINT C
```

L'emissione output è:

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

ABCDEFGHIJKL

1	1
1	1

MAT PRINT USING [rifer. di file,] numero di riga, nome di insieme $\left[\begin{Bmatrix} , \\ ; \end{Bmatrix} \right]$ [nome di insieme]] ... $\left[\begin{Bmatrix} , \\ ; \end{Bmatrix} \right]$
: $\left[\begin{Bmatrix} \text{stringa di caratteri} \\ \text{maschera di stampa} \end{Bmatrix} \right]$... maschera di stampa $\left[\begin{Bmatrix} \text{stringa di caratteri} \\ \text{maschera di stampa} \end{Bmatrix} \right]$...

MAT PRINT USING

L'istruzione MAT PRINT USING e la sua collegata, la "maschera di stampa", permettono di visualizzare o stampare tutti gli elementi di un insieme specificato, in un formato di nostra scelta, senza indicare ciascun elemento individuale dell'insieme. Nella istruzione MAT PRINT USING si possono specificare i valori da visualizzare o da stampare ed il numero di riga della "maschera di stampa" che si deve usare per il formato di emissione dei dati. La sintassi della istruzione MAT PRINT USING è la seguente:

MAT PRINT USING [riferim. di file,] numero di riga, nome di insieme $\left[\begin{Bmatrix} , \\ ; \end{Bmatrix} \right]$ [nome di insieme]] ... $\left[\begin{Bmatrix} , \\ ; \end{Bmatrix} \right]$

dove:

riferimento di file è FLP (per la stampatrice) oppure va da FL0 a FL9 per il file logico.

numero di riga è il numero della specifica "maschera di stampa".

nomi di insieme sono insiemi numerici o di caratteri i cui elementi si devono stampare nel formato specificato nella "maschera di stampa". E' necessario almeno un nome di insieme. I nomi di insiemi possono tra loro essere separati da virgole o punti e virgola. L'ultimo nome di insieme può anche essere seguito da una virgola o da un punto e virgola.

La sintassi della "maschera di Stampa" è:

: $\left[\begin{Bmatrix} \text{stringa di caratteri} \\ \text{maschera di stampa} \end{Bmatrix} \right]$... maschera di stampa $\left[\begin{Bmatrix} \text{stringa di caratteri} \\ \text{maschera di stampa} \end{Bmatrix} \right]$...

dove:

stringa di caratteri è una stringa di caratteri che non contiene apici.

maschera di stampa è la specifica di formato (vedi "Specifiche di Formato nel paragrafo PRINT USING e IMAGE"). E' necessario almeno una maschera di stampa nella istruzione.

Quando viene elaborata una "maschera di stampa", vengono valutati i riferimenti agli insiemi specificati e vengono stampati i suoi elementi, nell'ordine in cui compaiono nella istruzione MAT PRINT USING, nel formato corrispondente alla "maschera di stampa" specificata. Una stringa di costanti di caratteri verrà visualizzata o stampata esattamente come è stata immessa nella "maschera di stampa".

Ciascun elemento di insieme viene visualizzato per riga in conformità con le specifiche di formato definite dalla "maschera di stampa" collegata.

Quando viene visualizzata, la prima riga di ciascun insieme inizia nella posizione iniziale di una nuova riga ed è separata dalla precedente riga da due righe di spazi vuoti. Ogni successiva riga di insieme comincia all'altezza di una nuova riga ed è separata dalla riga precedente mediante una riga di spazi vuoti.

Se il numero di elementi in una riga di un insieme in una istruzione MAT PRINT USING supera le specifiche di formato della "maschera di stampa", il delimitatore tra i riferimenti di insieme comanda l'emissione:

- Se il delimitatore è una virgola od uno spazio vuoto, e viene raggiunta la fine delle specifiche di formato della "maschera di stampa", viene visualizzata o stampata attuale riga, e l'emissione prosegue su una nuova riga usandodall'inizio la "maschera di stampa".
- Se il delimitatore è un punto e virgola e viene raggiunta la fine delle specifiche di formato della "maschera di stampa", l'emissione prosegue sulla stessa riga continuando ad usare dall'inizio la stessa "maschera di stampa".

Dopo che è stata visualizzata o stampata l'ultima riga dell'ultimo insieme, il delimitatore che segue quel riferimento nella istruzione MAT PRINT USING determina quanto segue:

- Se il delimitatore è una virgola od uno spazio vuoto, l'attuale riga viene stampata o visualizzata e l'emissione successiva inizierà su una nuova riga.
- Se alla fine dell'elenco di riferimento di insieme si trova un punto e virgola, non viene impostata una nuova riga dopo la visualizzazione o la stampa dell'ultimo dato. Pertanto i dati successivamente visualizzati o stampati si troveranno sulla stessa riga.

Note relative a PAT PRINT USING

- Se il 5100 non possiede allacciata una stampatrice, l'istruzione MAT PRINT USING FLP arresta il programma a meno che sia stata specificata l'opzione P = D nel comando RUN.
- La "maschera di stampa" deve contenere almeno una specifica di conversione.
- Se il numero di elementi in una riga di insieme supera il numero delle specifiche di conversione nella "maschera di stampa", la "maschera di stampa" viene riutilizzata per gli elementi restanti di quella riga, e se il delimitatore che segue l'insieme è una virgola, viene iniziata una nuova riga.

Esempio

Il seguente esempio mostra l'elaborazione di una istruzione MAT PRINT USING:

```
0010  DIM A(4,3)
0020  :### ##.## ##.## |||
0030  MAT A=(1)
0040  MAT PRINT USING FLP,0020,A
```

L'emissione che apparirà è:

```
1  .  1.00  10.00E-01
1    1.00  10.00E-01
1    1.00  10.00E-01
1    1.00  10.00E-01
```

MAT PUT riferimento di file, nome di insieme [,nome di insieme] . . .

MAT PUT

L'istruzione MAT PUT permette di collocare il valore di ciascun elemento di un insieme in un file specificato, senza aver bisogno di riferirsi a ciascun elemento individuale dell'insieme. Il file deve essere stato precedentemente aperto mediante una istruzione OPEN. La sintassi della istruzione MAT PUT è la seguente:

MAT PUT riferimento di file, nome di insieme [,nome di insieme] . . .

dove:

riferimento di file va da FL0 a FL9, per identificare il file definito nella OPEN.

nome di insieme è il nome di un insieme numerico o di caratteri. E' necessario almeno un insieme.

Quando viene elaborata una istruzione MAT PUT, i valori degli elementi dell'insieme specificato vengono immessi in una memoria di transito per il file specificato, ad iniziare dalla posizione dell'attuale file. Il file viene ordinato sequenzialmente affinché il primo valore immesso sia poi il primo valore assegnato dal file quando gli ci si riferisce con una istruzione MAT GET (oppure GET). Quando la memoria di transito diventa piena, il contenuto viene trasferito come un unico record.

Un file può essere attivato unicamente da una istruzione OPEN e disattivato da una CLOSE o dalla fine dell'esecuzione programma.

Note Relativa a MAT PUT

- Un file attivato come un file di immissione non può venir specificato in una MAT PUT. Lo si deve prima chiudere e poi riaprire come file di emissione oppure bisogna emettere una istruzione RESET END (vedi RESET).
- Se al file di emissione si esaurisce lo spazio prima che siano stati collocati nel file tutti gli articoli dell'elenco di emissione, finisce l'elaborazione del programma.
- Una istruzione MAT PUT che si riferisca ad un file al momento chiuso, determina la fine dell'elaborazione del programma.

Esempio

Il seguente esempio mostra una istruzione MAT PUT:

0100 MAT PUT FL4, B, C\$

In questo esempio i valori degli elementi dell'insieme B e dell'insieme C\$ vengono immessi nella file FL4.

MAT READ nome di insieme [(righe [,colonne])] [,nome di insieme [(righe [,colonne])]]...

MAT READ

L'istruzione MAT READ permette di assegnare valori prelevandoli da una tabella di dati (definita da istruzioni DATA), agli elementi di un insieme senza doversi riferire a ciascun elemento individuale dell'insieme. Si può inoltre usare la istruzione MAT READ per il ridimensionamento degli insiemi. La sintassi della istruzione MAT READ è la seguente:

MAT READ nome di insieme [(righe [,colonne])] [,nome di insieme [(righe [,colonne])]]...

dove:

nome di insieme è il nome di un insieme numerico o di caratteri. E' necessario almeno un insieme.

righe, colonne sono le specifiche facoltative di ridimensionamento dell'insieme.

All'inizio dell'elaborazione di un programma, viene impostato un indicatore sul primo valore che è nella tabella dei dati. Allorchè si incontra una specifica READ o MAT READ, vengono assegnati valori in successione dalla tabella dei dati alle variabili od agli elementi di insieme specificati nella READ, oppure all'intero insieme specificato nella MAT READ. I valori vengono assegnati all'insieme, per righe, a partire dall'attuale posizione dell'indicatore della tabella dei dati. L'indicatore della tabella dei dati può essere ripristinato usando l'istruzione RESTORE.

Se una specifica di ridimensionamento segue il nome di insieme, la parte intera troncata di ciascun valore delle righe e colonne viene usato per ridimensionare l'insieme prima che gli siano assegnati i valori.

Note relative a MAT READ

- Prima di venir utilizzati in una MAT READ gli insiemi devono essere stati precedentemente definiti implicitamente od esplicitamente in una istruzione USE o DIM.
- Se viene esaurita la tabella dei dati e restano ancora nella istruzione MAT READ insiemi non assegnati o elementi di insiemi, si verificherà un errore.
- Se nel programma non vi sono istruzioni DATA, la MAT READ determinerà un errore.
- L'istruzione MAT READ non può venir usata in un programma assegnato ad uno dei tasti funzionali. Ciò determinerebbe un errore quando la funzione fosse elaborata.
- Se vengono immesse specifiche di ridimensionamento, si devono seguire le regole enunciate nel Capitolo 3, Ridimensionamento degli Insiemi.

Esempio

Il seguente esempio mostra una istruzione MAT READ:

```
0100 MAT READ A$ (2), A, B(12)
```

In questo esempio l'insieme A\$ viene ridimensionato a due elementi di 18 caratteri ciascuno; l'insieme B viene ridimensionato a 12 righe, e poi i valori vengono assegnati dalla tabella di dati agli insiemi A\$, A, e B.

COMPATIBILITA' DEL BASIC 5100 CON IL BASIC DELL'IBM 370/VS

Il linguaggio BASIC usato nel 5100, differisce dal BASIC dell'IBM 370/VS nei seguenti punti:

- Nel 5100 i caratteri alfabetici minuscoli non sono disponibili.
- Nel 5100 non è disponibile il doppio apice ("").
- Il 5100 fornisce unicamente un tipo di precisione (13 cifre significative).
- Il 5100 fornisce costanti esadecimali, che sono racchiuse tra apice e precedute da X.
- Il 5100 permette l'emissione di una TAB (espressione) in una istruzione PRINT.
- Il 5100 permette di inviare una emissione preparata per la stampa sia alla stampatrice, oppure allo schermo o ad una file su nastro. Questa capacità è fornita mediante un parametro addizionale (FLP) nelle istruzioni PRINT, MAT PRINT, PRINT USING, e MAT PRINT USING. Un parametro FLO-FL9 indica una emissione al file aperto con una istruzione OPEN.
- Il 5100 permette di specificare la descrizione del file che viene fornita dal sistema operativo (nel BASIC 370/VS).
- Il 5100 permette di applicare l'istruzione CHAIN ad un programma senza inizializzare l'area di dati riservata, per quelle variabili specificate nell'istruzione USE.
- Il 5100 permette di effettuare un elenco di variabili in una istruzione USE, incluse le dimensioni per gli insiemi. In congiunzione con una istruzione CHAIN, la USE permette di trasferire i dati da un programma ad un altro.
- Il 5100 fornisce la funzione TAB(x) ma non fornisce le seguenti funzioni BASIC VS: CEN(x), CLK, CNT, CPU, DAT(x), DOT(x,y), FAH(x), IDX x,y), IDY(x), KLN(x), KPS(x), LEN(x), MAX (x,u,z), MIN(x,y,z), NUM(x), PRD(x), RLN(x), SUM(x), e TIM.
- Il 5100 non fornisce le funzioni di riordino degli insiemi in modo crescente (ASORT) e decrescente (DSORT).
- Il 5100 ha bisogno di numeri di riga da 1 fino a 9999 e consente funzioni speciali create dall'utente ai numeri 9990-9999.
- Il 5100 non supporta le seguenti istruzioni BASIC: DELETE FILE, EXIT, FORM, READ FILE, REREAD FILE, REWRITE FILE, e WRITE FILE.
- Il 5100 non fornisce i parametri EXIT, CONV o IOERR nella istruzione MAT GET, il parametro ELSE nella IF, i parametri EXIT o IOERR nella MAT PUT, od il parametro ALL nella istruzione OPEN.
- Il 5100 non sa interpretare il simbolo di esclamazione (!) in una istruzione "maschera di stampa", benchè il simbolo possa essere immesso sovrapponendo l'apice al punto decimale.

USO E PRECAUZIONI CON LE BOBINE A NASTRO

- Preservare le cassette a nastro dalla polvere e dalla sporcizia. Le cassette che si usano di rado, dovrebbero essere riposte anche nei loro contenitori di plastica.
- Custodire le cassette a nastro lontano da campi magnetici e da materiali ferromagnetici che potrebbero venir magnetizzati. Una bobina qualsiasi esposta ad un campo magnetico potrebbe perdere informazioni.
- Non esporre le cassette a nastro a fonti di calore eccessive (superiori a 55 °C) od ai raggi diretti del sole.
- Non toccare nè pulire la superficie del nastro con le dita.
- Se una cassetta a nastro è stata esposta ad una temperatura anche di poco inferiore a 0 °C, dopo l'ultima volta che è stata usata, spostare il nastro dalla sua posizione prima di usarlo. La procedura per spostare il nastro dalla sua posizione è:
 1. Far uso del comando UTIL per spostare il nastro all'ultimo file marcato.
 2. Far uso del comando MARK per eseguire la marcatura del nastro a partire dall'ultimo file marcato fino alla fine del nastro. Per esempio:
MARK 200, 1, n
in cui n rappresenta il numero dell'ultimo file marcato, più uno. Ciò determinerà l'errore 151 (fine del nastro). Per proseguire battere ATTN.
 3. Usare il comando REWIND per riavvolgere il nastro.

SEGRETEZZA DEI DATI

Il 5100 fornisce un numero di funzioni che permette di proteggere dati importanti o delicati. La bobina a nastro magnetico può facilmente essere chiusa in una scrivania più di qualsiasi altro documento riservato. Bisogna ricordarsi che le bobine che contengono dati delicati sono sotto la vostra responsabilità. Si può riscrivere (OPEN per emissione e CLOSE) un file su nastro per far diventare i dati che si trovano sul nastro inaccessibili al 5100. Si può inoltre far uso delle seguenti istruzioni BASIC per cancellare i dati da un file su nastro.

```
10 DIM A(100)
20 OPEN FLO, 'E80',1,OUT
30 MAT PUT FLO,A
40 GOTO 30
```

Questo programma scriverà degli zeri in ogni posizione, e poi visualizzerà un messaggio di errore per indicare la fine del file. Oltre ai vantaggi di segretezza della bobina a nastro, si possono eliminare, tutti i dati che si trovano correntemente nell'area di lavoro del 5100, immettendo LOAD0, batterdo RESTART oppure spegnendo l'elaboratore.

Ricordarsi inoltre che la freccia di protezione da scrittura posta sulla bobina del nastro permette che una importante informazione presente sul nastro non possa venir distrutta da accidentali scritture (SAFE).

Appendice A. Caratteri BASIC

La seguente lista contiene l'elenco di tutti i caratteri disponibili sul 5100 BASIC e del loro relativo valore, procedendo dal più piccolo al più grande.

Carattere	Numero della Sequenza	Carattere	Numero della Sequenza
(blank)	65	=	127
'	76	↑	139
<	77	≥	141
(	78	[	174
+	79	≤	175
	80	]	190
&	81	≠	191
!	91	A	194
\$	92	B	195
*	93	C	196
)	94	D	197
;	95	E	198
-	97	F	199
/	98	G	200
,	108	H	201
>	111	I	202
?	112	J	210
:	123	K	211
#	124	L	212
@	125	M	213
,	126	N	214

Carattere	Numero della Sequenza	Carattere	Numero della Sequenza
O	215	Y	233
P	216	Z	234
Q	217	0	241
R	218	1	242
\	225	2	243
S	227	3	244
T	228	4	245
U	229	5	246
V	230	6	247
W	231	7	248
X	232	8	249
		9	250

Appendice B. Messaggi di Errore BASIC e Correzioni Suggerite all'Operatore

L'elenco che segue contiene tutti i codici di errore del 5100. Una qualsiasi individuazione di errore disattiverà i tasti della tastiera e determinerà il lampeggiamento dell'intero schermo. Per interrompere il lampeggiamento dello schermo, battere ATTN. I codici di errore inferiori a 100 sono errori di I/O e vengono visualizzati con 3 cifre numeriche, seguite da tre spazi vuoti ed uno dei seguenti indirizzi di unità: 500 per la stampatrice, E80 per l'unità incorporata, E40 per l'unità nastro ausiliaria. Codici di errore superiori a 100 vengono semplicemente visualizzati con tre cifre numeriche. Se non diversamente specificato, è possibile correggere ciascun errore di elaborazione mediante una delle seguenti procedure etichettate con (A), (B), (C), o (D) :

- (A) Immettere GO o GO END per finire il programma oppure immettere GOx per proseguire, in cui x è il numero di riga di una istruzione qualsiasi del programma.
- (B) Immettere GO END per finire il programma oppure GO per proseguire.
- (D) Immettere GO o GO END per finire il programma.
- (C) Correggere l'istruzione od il comando che si trova sulla riga di immissione poi battere EXECUTE.

Tener presente che se gli errori vengono individuati durante l'elaborazione di un programma che coinvolga operazioni su nastro, si dovrebbe immettere GO END per assicurarsi che i file su nastro siano esattamente chiusi.

Se avvengono su un file su nastro errori di I/O (01-99), il file viene automaticamente marcato non aperto, e non gli si può più accedere con le istruzioni GET o PUT. Un file di emissione che non è stato esattamente chiuso non può più essere accessibile. La istruzione GO END chiude tutti gli altri files.

Errore	Significato	Correzione
002	Comando di unità errato, o sequenza di comandi di unità errati.	Immettere GO END per finire il programma. Controllare se sono stati specificati in modo giusto gli indirizzi di unità nelle istruzioni OPEN e CLOSE. I giusti indirizzi delle unità sono 500, E40 e E80. L'errore può inoltre essere stato causato dal tentativo di aprire od applicare l'istruzione CHAIN al file 0, ad esempio: OPEN FL1, 'E80', 0, OUT. Valori di file tipo .5 possono inoltre essere la causa dell'errore. Se le istruzioni OPEN o CHAIN erano sbagliate, correggerle ed eseguire nuovamente il programma. Se invece sono giuste ritentare l'operazione. Se l'errore si ripresenta, chiamare il servizio manutenzione.
003	Errore di circuito interno oppure non è possibile inserire correttamente il nastro	Se si stà eseguendo un programma BASIC, immettere GO END per finire il programma quindi ritentare l'operazione che ha determinato l'errore. Se l'errore capita con un comando LOAD o SAVE, ritentare il comando. Se l'errore persiste, chiamare il servizio manutenzione.
004	Non si riescono a trovare dati sul nastro oppure una operazione di REWIND ha impiegato più di tre minuti. Questo errore avviene quando si tentano operazioni su una bobina a nastro che non è stata marcata (MARK).	Se l'errore avviene quando il nastro viene estratto durante una operazione REWIND, allora non c'è bisogno di alcuna correzione. Tuttavia se l'errore per eccesso di tempo, non è imputabile ad una precedente rimozione del nastro durante una operazione di REWIND, chiamare il servizio manutenzione. Se l'errore avviene quando non è in funzione una operazione di REWIND, bensì un programma BASIC è in elaborazione, immettere GO END per finire il programma, poi ritentare l'operazione che ha determinato l'errore. Se l'errore capita con un comando LOAD o SAVE, ritentare il comando. Questo errore può inoltre capitare se un comando REWIND viene emesso relativamente all'unità nastro ausiliaria e questa è spenta. Se l'errore ricapita, battere il comando REWIND (dopo aver immesso GO END se necessario) e ritentare l'operazione. Se l'errore ricapita, chiamare il servizio manutenzione.
005	Cassetta a nastro non inserita.	Inserire la cassetta a nastro. Se l'errore capita con una operazione di comando, reimmettere il comando. Se l'errore capita durante l'elaborazione di un programma: – Inserire la cartuccia a nastro ed immettere GO END per finire il programma. – Inserire la cartuccia a nastro ed immettere GOx per continuare, dove x è il numero di riga della istruzione OPEN. Se l'errore ricapita, chiamare il servizio manutenzione.
006	Il nastro è nello stato di protezione di file. (SAFE).	se si desidera scrivere sul nastro, ruotare l'indicatore SAFE, posto in alto sulla sinistra della cassetta, dalla sua posizione di protezione è: 1. Se l'errore avviene con una operazione di comando, ritentare il comando. 2. Se l'errore avviene all'apertura di una file durante l'elaborazione del programma: – Immettere GO END per finire il programma. – Immettere GOx per continuare, dove x è il numero di riga della istruzione OPEN. Se l'errore avviene nuovamente, chiamare il servizio manutenzione.

Errore	Significato	Correzione
007	E' avvenuto un errore CRC (controllo di ridondanza ciclica).	Se l'errore avviene durante l'elaborazione di un programma, immettere GO END per finire il programma. Ritentare poi l'operazione che ha determinato l'errore. Se l'errore ricapita e l'errore è sul nastro, il file su nastro è probabilmente inutilizzabile. Far uso del programma per la correzione dei nastri se il file è del tipo 1, 2, 3, 8. (v. PATCH).
008	Non si riesce a trovare il successivo record fisico che ci si aspetta	Se l'errore riavviene, chiamare il servizio manutenzione. Immettere GO END per finire il programma. Ritentare l'operazione. Se l'errore avviene ancora, copiare i files che seguono quello che contiene l'errore su un'altra cassetta a nastro; dopodichè rimarcare la prima cassetta a partire dal file che contiene l'errore, per tutti i files che si desidera.
009	Durante la lettura di dati da file su nastro si è incontrata la fine dei dati. (Tener presente che possono essere aggiunti al file un maggior numero di dati).	questo errore potrebbe essere avvenuto se il nastro è stato rimosso durante una precedente operazione di scrittura o un comando MARK. Immettere GO END per finire il programma. Si dovrebbe controllare la logica del programma onde correggerlo laddove effettua la richiesta di troppi dati. Opzionalmente si può indicare in una istruzione GET o MAT GET il parametro EOF. Esso definisce il numero della riga cui il programma salta allorchè si incontra la fine del file. Correggere pertanto la logica del programma e/oppure aggiungere alle istruzioni GET, MAT GET il parametro EOF.
010	Si è verificata la fine file durante una lettura da un nastro nel tentativo di leggere al di là dell'ultimo record, oppure durante una operazione di scrittura, si è tentato di scrivere al di là dell'ultimo record del file stesso.	Immettere GO END per finire il programma. Se l'errore avviene durante un'operazione SAVE, vuol dire che il file non è abbastanza grande. Si dovrà allora memorizzare i dati in un file più grande per contenere i dati. Poi svolgere l'operazione SAVE. Se l'errore avviene durante l'elaborazione di un programma, si dovrebbe controllare la logica del programma onde correggerlo nei punti in cui effettua la richiesta di troppi dati. Opzionalmente, si può indicare in una istruzione GET o MAT GET il parametro EOF. Esso definisce il numero della riga cui il programma salta allorchè si incontra la fine del file. Correggere pertanto la logica del programma e/oppure aggiungere alle istruzioni GET, MAT GET il parametro EOF. Se l'errore avviene durante l'uso di PUT, MAT PUT, vuol dire che il file non è sufficientemente grande per contenere i dati. Pertanto specificare un file più grande oppure marcare lo stesso file più grande, poi cambiare le appropriate istruzioni nel programma e quindi eseguirlo di nuovo.
011	Non viene trovato il numero di file specificato.	Esser sicuri che è stata caricata la cassetta giusta. Poi specificare l'esatto numero di file.
012	Si è giunti alla fine materiale del nastro. Tener presente che questo potrebbe capitare quando un comando UTIL viene emesso su un nastro nuovo che si trova in una posizione precedente ai due fori sul nastro.	Emettere un comando REWIND poi riprovare con il comando UTIL. Se ricapita l'errore, inserire un'altra cassetta a nastro e ritentare. Se l'errore avviene nuovamente, chiamare il servizio manutenzione.

Errore	Significato	Correzione
013	L'unità specificata non è allacciata, oppure è stato immesso un comando UTIL MODE COM e il dispositivo di comunicazione non è disponibile.	Probabilmente è stato specificato un errato indirizzo di unità oppure è stato specificato un errato parametro. Specificare un giusto indirizzo di unità dopo aver emesso un GO END per finire il programma.
014	Non si può scegliere una sottounità in un indirizzo di unità (come un nastro). Ciò potrebbe significare che è stato specificato un errato indirizzo o un errato numero di unità o che è avvenuto un errore nei circuiti od un errore di programmazione	Assicurarsi che è stato specificato un giusto indirizzo di unità, poi riprovare l'operazione, o immettere GO END per finire il programma.
050	La stampatrice ha individuato la fine del modulo di carta.	Inserire altri moduli. Se l'errore avviene con una operazione di comando, ritentare il comando. Se l'errore avviene durante l'elaborazione di un programma: Immettere GOx per continuare, dove x è il numero di riga visualizzato. Potrebbe però accadere che l'ultima riga venga stampata due volte.
051	La stampatrice non è pronta.	Accendere la stampatrice. Se l'errore avviene in una operazione di comando, riprovare il comando. Se l'errore avviene durante l'elaborazione di un programma, immettere GOx in cui x è il numero di riga visualizzato
052-059	Errori di stampatrice	Ⓐ Se l'errore ricapita, chiamare il servizio manutenzione
100	Errore di sintassi in una istruzione. Essa non è accettata.	Ⓓ
101	Costante aritmetica sbagliata-riga non accettata.	Ⓓ
102	Costante Carattere incompleta istruzione non accettata.	Ⓓ
103	Numero dispari di parentesi in una espressione-istruzione non accettata.	Ⓓ
104	E' stato dimenticato il delimitatore tra costanti carattere-istruzione non accettata.	Ⓓ
105	Errore di sintassi nella funzione STR-istruzione non accettata.	Ⓓ
106	L'istruzione che seguiva LOAD, KEYx non era una istruzione di intestazione-istruzione non accettata.	Reimmettere LOAD, KEYx poi immettere una nuova istruzione di intestazione. (REM, CMD, TXT o NUL)
107	Errato numero di riga per l'intestazione del gruppo funzionale specificato-istruzione non accettata.	Reimmettere LOAD, KEYx poi immettere una nuova istruzione di intestazione.

Errore	Significato	Correzione
108	Errore di sintassi nella intestazione di un gruppo funzionale specificato-istruzione non accettata.	Reimmettere LOAD0,KEYx poi immettere una nuova istruzione di intestazione.
109	Tipo di istruzione non permessa in un gruppo funzionale-riga non accettata.	Ⓓ
110	Errata sintassi di un comando o tentativo di rimarcare un file contenente un errore CRC fino e oltre l'errore CRC. Comando non accettato.	Ⓓ oppure reimmettere REWIND e rimarcare a partire da un file prima dell'errore CRC.
111	Errato uso di un comando: 1. Nessun posto ove andare in un comando GO 2. Nessuna istruzione per il RENUM 3. Nessun programma per il RUN 4. Nessuna istruzione per il LIST	Ⓓ
112	Incapacità a determinare un numero di riga od il successivo più piccolo numero di riga in LIST.	Ⓓ
113	Tasto funzionale non caricato o definito come NUL. Richiesta conclusa.	Definire il tasto funzionale prima di riprovare.
114	Riga creata dal comando AUTO oltre 9999. Comando concluso.	
115	RENUM determinerà una riga > 9989.	Poni un incremento od un numero di partenza più piccolo e riprova.
116	Numero di istruzione in una istruzione BASIC non trovato durante RENUM-il comando viene interrotto. Viene visualizzato il numero di riga della istruzione con il riferimento sbagliato.	Ⓓ
117	L'unità specificata nel comando non può accettare la funzione richiesta-comando non accettato.	Ⓓ
118	Fine di un file raggiunta prima della conclusione di SAVE-il comando viene concluso.	Vedi Comando SAVE nel Capitolo 2.
119	La riga di immissione ha superato 64 caratteri. La riga viene troncata.	Vedi Comando LOAD nel Capitolo 2.

Errore	Significato	Correzione
120	Le dimensioni dell'insieme non sono comprese nell'intervallo da 1 a 255.	
121	Il tipo del file su nastro non concorda con il tipo dell'area di lavoro oppure si è fatto il tentativo di accedere ad un file al di là dell'ultimo file marcato.	Montare il giusto nastro poi ①
122	Record di dati maggiore di 324 caratteri incontrato durante LOADx, DATA. L'area di lavoro è azzerata.	Far uso di un programma BASIC con istruzioni GET o PRINT per ispezionare i dati.
150	Comando MARK su un file già marcato.	Immettere GO unicamente nelle posizioni 1 e 2 della riga di immissione. (ATTENZIONE: Questa operazione è pericolosa per dati eventualmente presenti. Vedere comandi MARK).
151	Viene raggiunta la fine della bobina del nastro prima del completamento di un MARK. Vengono indicati K-bytes e l'ultimo record marcato.	
152	Area di lavoro insufficiente.	Opzione dell'utente.
153	L'istruzione DIM contiene un insieme già definito esplicitamente od implicitamente.	①
154	I valori per il ridimensionamento sono zero, negativi o maggiori di 255.	①
155	Nuove dimensioni di ridimensionamento in eccedenza rispetto a quelle definite in DIM o USE o da una assegnazione implicita.	①
156	Errate dimensioni per la funzione usata, e per l'insieme prodotto specificato.	①
157	Funzione IDN specificata per una matrice che non è quadrata.	①
158	Insieme referenziato di una istruzione MAT precedentemente non definito in istruzioni DIM o USE o da una definizione implicita.	①
159	Errato tipo di dati dal file dalla tastiera	①

Errore	Significato	Correzione
160	L'elemento di dati rilevato nell'istruzione DATA non è dello stesso tipo (numerico o di caratteri) della corrispondente variabile dell'elenco READ.	Ⓐ
161	Troppo pochi DATA in confronto alle istruzioni READ emesse.	Ⓐ
162	Errato formato dei dati numerici dal file o dalla tastiera.	Ⓐ
163	Istruzione OPEN specificata per un file attualmente già aperto.	Ⓐ
	Emissione di un comando per un file che è già aperto nel programma.	Ⓑ
164	Istruzione OPEN che specifica una immissione per una unità di emissione oppure viceversa specifica una emissione per una unità di immissione.	Ⓐ
165	File multipli aperti su una stessa unità.	Ⓐ
166	Tentativo di aprire APL-file protetto con una "parola d'ordine".	Ⓐ
167	Il tipo del file non è il giusto tipo per una immissione.	Ⓐ
168	Istruzione di emissione specificata per un file aperto per l'immissione, oppure istruzione di immissione specificata per un file aperto per l'emissione, oppure il file non è stato aperto, oppure è stato specificato PUT per un file per il quale è stata già emessa una istruzione PRINT, o viceversa.	Ⓐ
169	Operazione di I/O iniziata durante la valutazione della funzione ad esso riferita in una operazione di I/O.	Ⓐ
170	Istruzione PRINT USING che specifica elementi di dati ma non esistono nella relativa "maschera di stampa" specifiche di conversione.	Ⓐ
171	Numero di riga (visualizzato) che si riferisce ad una istruzione inesistente nel programma.	Ⓐ

Errore	Significato	Correzione
172	Istruzione FNEND non legata ad una istruzione DEF.	Ⓐ
173	Istruzione NEXT non corrispondente alla relativa FOR, oppure la variabile di comando dell'iterazione non corrisponde.	Ⓐ
174	Istruzione NEXT elaborata prima della corrispondente FOR.	Ⓐ
175	Istruzione RETURN, priva di un valore, incontrata quando però nessun GOSUB è attivo.	Ⓐ
176	Istruzione RETURN, con un valore, incontrata quando però nessuna funzione dell'utente è attiva.	Ⓐ
177	Istruzione USE incontrata non in prima posizione.	Ⓐ
178	Tentativo di riferimento oltre il diciottesimo carattere in una funzione STR (stringa).	Ⓒ
200	OVERFLOW: numero probabilmente infinito. E' stato assunto il valore di $\pm .9999999999999999E99$.	Ⓑ
201	UNDERFLOW: numero probabilmente infinitamente piccolo. E' stato assunto il valore di 0.	Ⓑ
220	Istruzione troppo complessa, oppure troppe funzioni incuneate tra loro vengono chiamate.	Ⓒ
221	Area di lavoro esaurita mentre si cerca di assegnare spazio per una nuova variabile.	Ⓒ
222	Area di lavoro insufficiente per calcolare un determinante.	Ⓒ
223	Troppi indici usati per un riferimento ad un insieme.	Ⓒ
224	Indice al di fuori dell'intervallo di dimensioni troppo grande o negativo.	Ⓒ
225	Matrice non quadrata per una funzione DET, o matrice non definita.	Ⓒ

Errore	Significato	Correzione
227	Errato numero di parametri specificato per un riferimento di funzione definita dall'utente.	Ⓒ
228	L'argomento di LOG è zero o negativo.	Ⓒ
229	L'argomento della radice quadrata è negativo.	Ⓒ
230	L'argomento dell'arcocoseno è maggiore di 1 o minore di -1.	Ⓒ
231	Elevazione a potenza: tentativo di elevare 0 alla potenza o, oppure 0 ad una potenza negativa.	Ⓒ
232	Elevazione a potenza: tentativo di elevare un numero negativo ad una potenza non intera.	Ⓒ

PIANO DELLA INSTALLAZIONE

Condizioni

L'elaboratore da tavolo IBM 5100 e le unità collegate sono progettate con i seguenti presupposti:

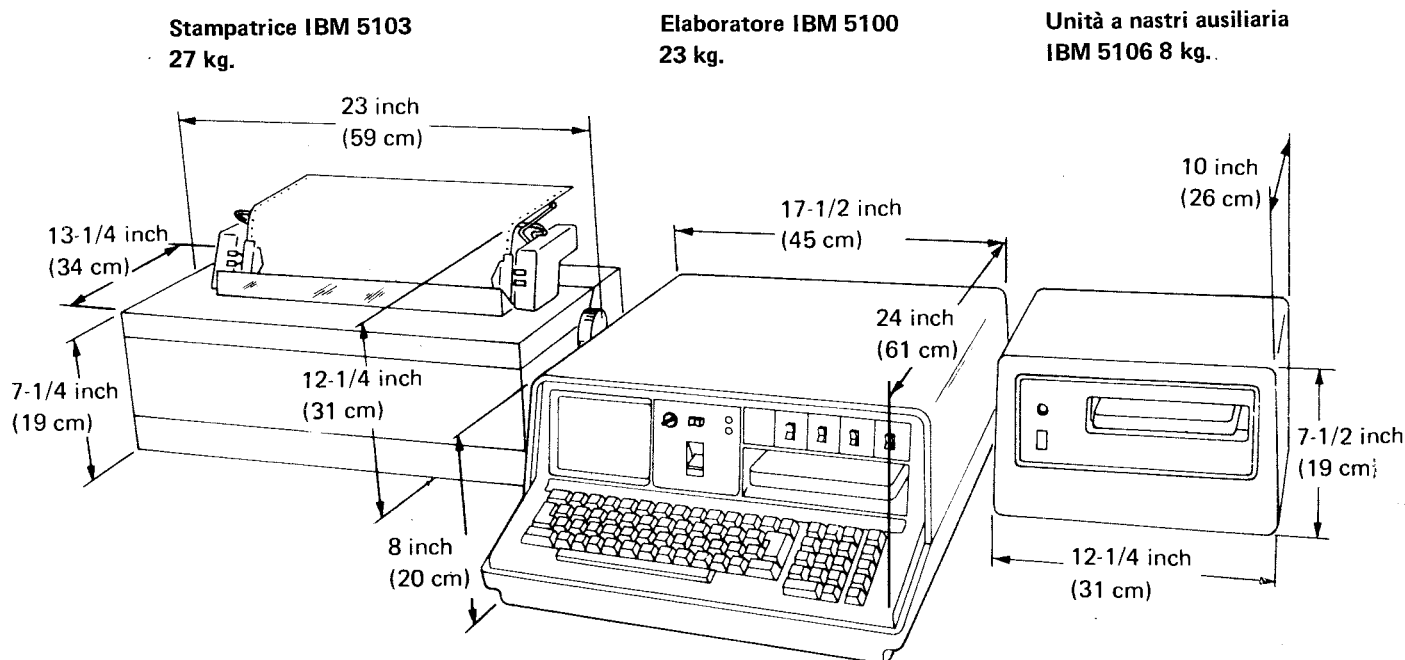
Condizioni Operative	Condizioni non Operative
Temperatura 60°-90° F (15°-32° C)	50°-105° F (10°- 43° C)
Umidità relativa 8 - 80	8 - 80
Temperatura massima a bulbo bagnato 73° F (23° C)	80° F (27° C)

Non si dovrebbe esporre l'elaboratore per troppo tempo ad intense temperature. Ad esempio, non bisogna riporre l'elaboratore nel baule dell'auto quando fa molto caldo o molto freddo. Se è proprio necessario esporre l'elaboratore a temperature estreme; questo dovrebbe essere acclimatato alle condizioni operative prima di accenderlo:

Se l'elaboratore è stato esposto a fonti di calore, si deve provvedere a raffreddarlo fino a che sia possibile posarci sopra una mano; a questo punto lo si può accendere.

Se l'elaboratore è stato esposto al freddo purchè non ci sia nè brina nè rugiada depositata sulle superficie esterne, nè sia visibili sulle parti interne, allora lo si può accendere.

Se l'elaboratore è stato esposto al freddo e c'è brina o rugiada sulle parti esterne oppure è visibile sulle parti interne, è necessario acclimatare l'elaboratore per 8 ore dopo che la brina o la rugiada siano scomparse. Ciò si fa per essere sicuri che tutta la rugiada depositata internamente sia evaporata prima di accendere l'elaboratore; viceversa l'elaboratore può rimanere danneggiato.



Specifiche elettriche

Per ciascuna unità è necessaria un'alimentazione a 220 volt, 15 ampere, monofase, 50 cicli con protezione di terra. Sono consentite tolleranze di ± 10 sul voltaggio e $\pm 1/2$ ciclo sulla frequenza.

Sarebbe consigliabile una presa doppia in modo da alimentare dalla stessa uscita sia l'elaboratore da tavolo IBM 5100 che una unità nastro ausiliaria o la stampatrice.

La spina di rete dell'unità nastro ausiliaria possiede una presa di rete in modo da poterci collegare con la stampatrice, il monitor TV od un modulatore/demodulatore di comunicazioni.

Tutti i cavi di collegamento sono lunghi 1,8 metri (6 feet).

La corrente necessaria è:

Elaboratore da tavolo 5100	1.5 A
Stampatrice 5103	0.5 A
Unità nastro ausiliaria 5106	0.25 A

L'emissione di calore è:

Elaboratore da tavolo 5100	780 BTU per ora
Stampatrice 5103	350 BTU per ora
Unità nastro ausiliaria 5106	130 BTU per ora

Collegamenti

L'elaboratore da tavolo 5100 è collegato alla stampatrice da un cavo per trasmissione di impulsi lungo 1.2 metri e con l'unità nastro ausiliaria da un cavo analogo lungo 0.6 metri. I cavi vengono forniti assieme ai dispositivi.

Quando sia la stampatrice che l'unità nastro ausiliaria si trovano installate contemporaneamente, la stampatrice deve essere collegata all'unità nastro ausiliaria invece che all'Elaboratore da Tavolo 5100.

Allorchè si trova installata solo l'unità nastro senza la stampatrice, il dispositivo di attacco dell'unità ausiliaria deve essere spostato dalla sua posizione.

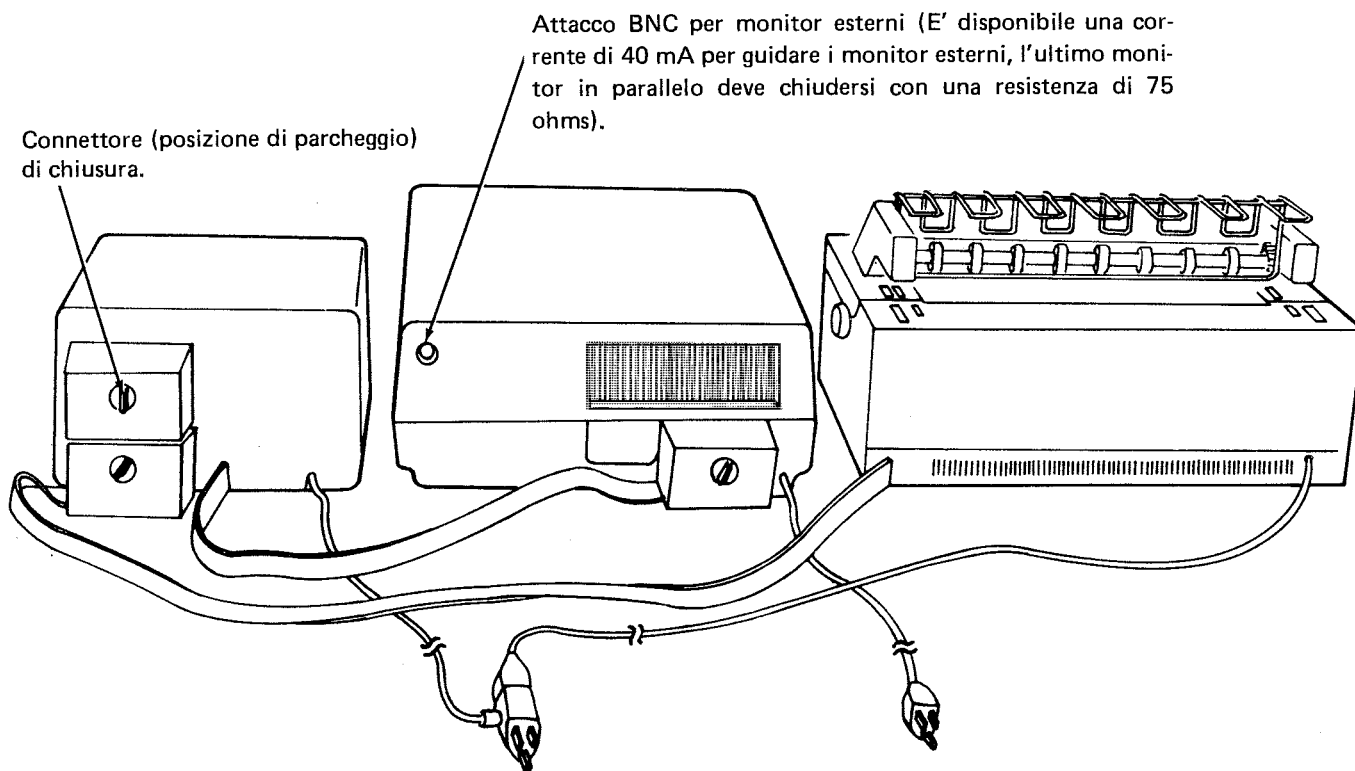
Comunicazioni

Interpellare il rappresentante commerciale IBM per avere le specifiche relative al "modem" (modulazione/demodulazione).

Interpellare immediatamente il rappresentante locale della società di comunicazioni per impiegare meno tempo possibile per ottenere il complesso degli allacciamenti. (linea commutata per trasmissione dati).

Il cavo di 1.8 metri necessario al collegamento 5100-modem viene fornito con il dispositivo.

I collegamenti del modem debbono essere compatibili con il connettore (a 25 posizioni) n° 205208 dell'AMP Corporation o ad un suo equivalente.



Tener presente che tutte le parti plastiche del 5100, delle 5103 e 5106 devono essere pulite con un detergente innocuo (non solventi) od alcool isopropilico. NON FAR MAI USO dei normali liquidi per pulire (solventi tipo benzina, etere etc.)

PROCEDURE DI INSTALLAZIONE

Procedura di installazione del 5100

Dopo aver collocato il 5100 nel luogo in cui si intende usarlo, controllare che l'interruttore ACCESO/SPENTO (POWER ON/POWER OFF), sistemato sul pannello frontale, si trovi nella posizione SPENTO. Infilare allora la spina nella presa di rete provvista di terra.

Nota: Un uso appropriato del 5100 richiede un'alimentazione provvista di una buona presa di terra.

Posizionare l'interruttore di accensione su ACCESO ed assicurarsi che sia funzionante il ventilatore interno:

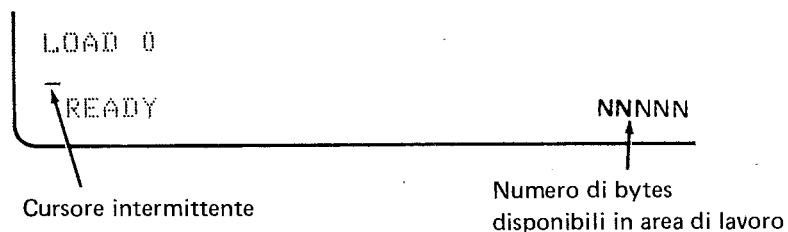
- Se il luogo dove si usa l'elaboratore non è troppo rumoroso, si dovrebbe udire il rumore del funzionamento del ventilatore.
- Se non si è certi del suo funzionamento, porre una striscietta di carta vicino alla feritoia di ingresso dell'aria sotto all'elaboratore. La punta della striscietta dovrà essere risucchiata dalla macchina.

Se pare che il funzionamento del ventilatore sia negativo o scarso, controllare l'allacciamento alla spina di rete è buono, porre l'interruttore di accensione su SPENTO e chiamare la manutenzione. Interrompere poi la lettura di queste istruzioni.

Se il ventilatore funziona, aspettare circa 20 secondi, perchè il 5100 diventi pronto per l'uso.

Procedure di controllo BASIC

- 1. Dopo che sono passati al massimo 20 secondi dal momento dell'accensione dovrebbe comparire sullo schermo:



Se lo schermo non presenta la precedente informazione, controllare gli interruttori seguenti posti nella parte alta del pannello frontale.

- Ruotare il comando LUMINOSITA' (BRIGHTNESS) per verificare l'ottimale definizione dei caratteri.
- Porre l'interruttore REGISTRI (DISPLAY REGISTERS) nella posizione NORMALE (NORMAL).
- Porre l'interruttore S32 64 D32 (L32 64 R32) nella posizione centrale (64).
- Porre l'interruttore BASIC/APL (presente solo nella macchina col doppio linguaggio) sulla posizione BASIC.
- Se l'informazione visualizzata non è uguale a quella su esposta premere l'interruttore RIPARTENZA (RESTART). Questa operazione fa riciclare una seconda volta una parte della sequenza di accensione. Se l'informazione continua ad essere diversa da quella su esposta (dopo circa 20 secondi da quando si è premuto RIPARTENZA) chiamare la manutenzione.

- 2. Se lo schermo non mostrerà gli esatti risultati illustrati nei paragrafi seguenti di questa procedura di installazione, premere ancora RIPARTENZA, tornare al passo 1 e poi riprovare ancora. Se l'esatto risultato continua a non comparire, chiamare la manutenzione.

Immettere i dati illustrati dai tasti disegnati qui sotto. I dati dovranno comparire allorché verranno battuti i tasti.

Se si dovesse commettere un errore di battitura, si deve allora battere il tasto di spazio indietro  (prima di EXECUTE) per spostare

indietro il cursore e poi ribattere il tasto giusto. Se si batte EXECUTE ma la riga era sbagliata, allora battere una volta. Itz 

(posto a destra di ATTN) e poi reimmettere la riga. Se l'errore causa il lampeggiamento dello schermo, battere ATTN prima di reimmettere la riga.

Battere i tasti sotto esposti seguendo lo stesso ordine, riga per riga:

1 0 A = 3 EXECUTE

2 0 B = 9 EXECUTE

3 0 C = A * B EXECUTE

4 0 P R I ~~N~~ T A , B , C EXECUTE

R U N EXECUTE

- 3. Al di sotto delle righe del programma di controllo che è stato ora immesso, deve comparire la risposta 27 accanto alle costanti 3 e 9 (il programma le moltiplica).

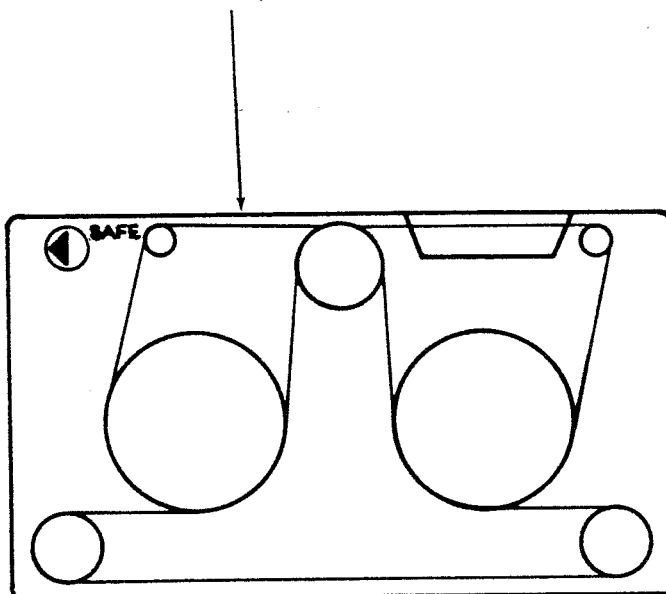
```
LOAD 0
10A=3
20B=9
30C=A*B
40PRINTA,B,C
RUN
    3                9                27
-
READY
```

- 4. Se si vuol saltare la parte relativa al controllo delle operazioni sul nastro perchè si vuole usare la procedura di controllo APL, bisogna inserire la cassetta a nastro nel 5100 e procedere al punto 6.

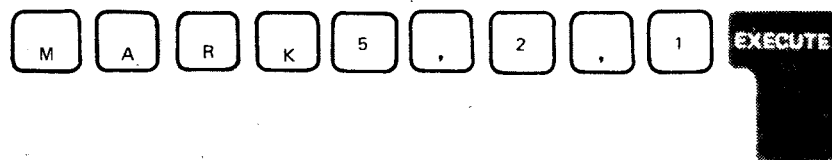
Si deve estrarre una nuova cassetta a nastro dal suo pacchetto. Controllare che la freccia sia puntata dalla parte opposta della parola SAFE come mostrato nella illustrazione. Per ruotare la freccia dalla parte opposta della parola SAFE, usare una moneta oppure un cacciavite.

Nota: NON USARE MAI una qualsiasi delle cassette a nastro già memorizzate che sono state inviate assieme alla macchina.

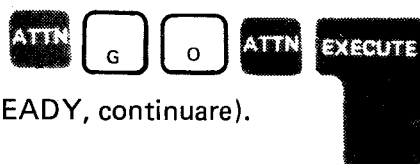
Questo lato deve essere quello di immissione nella macchina.



- ☐ 5A. Inserire la cassetta a nastro nel 5100 (la base metallica sotto) e schiacciarla con lieve pressione delle dita fino a che sia ben installata. Battere poi i seguenti tasti:



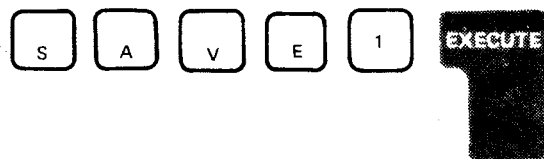
- 5B. Il precedente paragrafo imposta il nastro per conservare l'informazione. Allorchè lo schermo video visualizza nuovamente READY nell'angolo a sinistra in basso, procedere al punto 6. (Se viene visualizzato un codice di errore 150, vuol dire che il nastro è già marcato. Per rimarcare il nastro, battere i seguenti tasti:



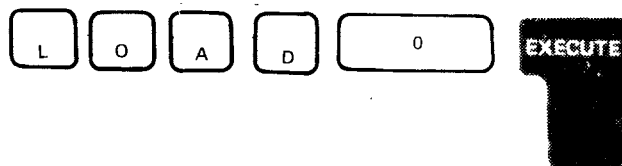
Quando sarà mostrato

READY, continuare).

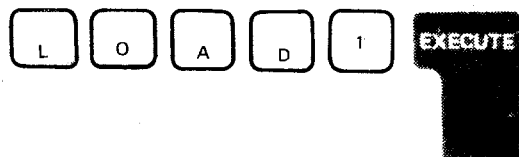
- ☐ 6. Battere i seguenti tasti:



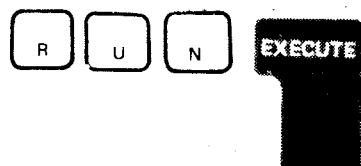
- ☐ 7. L'ultimo passaggio eseguito serve per scrivere il programma nel nastro, ma è ancora memorizzato nell'area di lavoro. Per provare il programma lo si deve leggere dal nastro, e pertanto deve prima essere cancellato dalla memoria. Per fare ciò, battere i seguenti tasti:



- ☐ 8. Per leggere il programma trasferendolo dal nastro al 5100, battere i seguenti tasti:



- ☐ 9. Il nastro si sposterà e quando verrà visualizzato nuovamente READY, battere i seguenti tasti:



Lo schermo video mosterà ancora:

```
RUN
3          9          27
-
READY
```

Con ciò è stata completata la procedura per il controllo del BASIC.

- ☐ 10. Controllare se sono stati ricevuti tutti i componenti come elencati lista di Controllo del Contenuto.

Se ne manca anche uno solo, farlo presente al rappresentante commerciale.

Se le parole al disopra della riga superiore dei tasti numerici sono etichettate sulla sinistra con:

APL	,l'elaboratore usa il linguaggio APL ed è privo del dispositivo di telecomunicazione
BASIC	, l'elaboratore usa il linguaggio BASIC ed è privo del dispositivo di telecomunicazione.
BASIC APL	, l'elaboratore usa sia il linguaggio APL che il BASIC
COMM BASIC APL	, l'elaboratore usa sia il linguaggio APL che il BASIC ed è fornito del dispositivo di telecomunicazioni.
COMM APL	, l'elaboratore usa il linguaggio APL ed è fornito del dispositivo di telecomunicazione.
COMM BASIC	, l'elaboratore usa il linguaggio BASIC ed è fornito del dispositivo di telecomunicazione.

Se non è stato eseguito il controllo APL su un elaboratore con il doppio linguaggio, porre l'interruttore BASIC/APL nella posizione APL, battere RESTART e proseguire con le procedure di controllo APL illustrate nell'appendice A del Manuale di Riferimento APL IBM 5100. Se sono già state eseguite le procedure di controllo APL, proseguire con il punto 11.

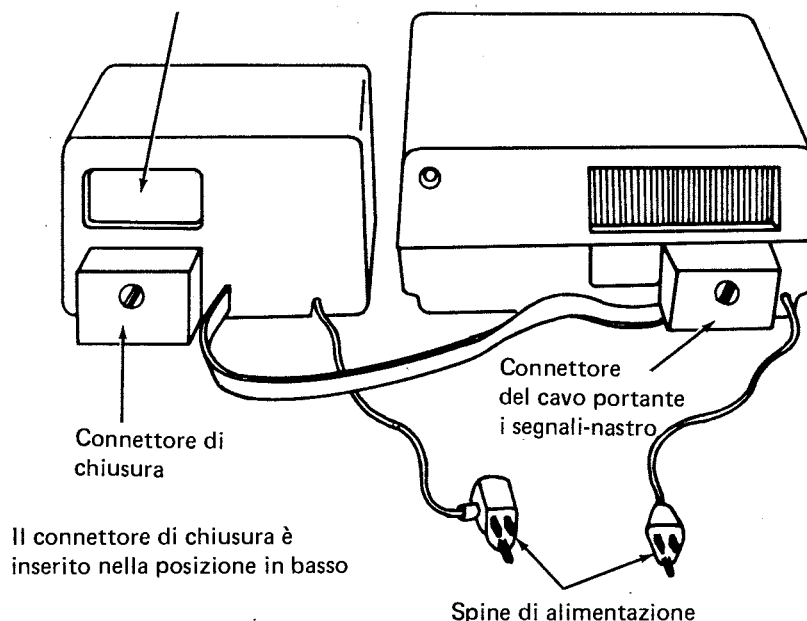
- ☐ 11. Se è stata installata l'unità nastro ausiliaria, aprire il pacco delle unità nastro e procedere con la Procedura di Installazione dell'unità nastro ausiliaria, esposta più oltre. Dopo avere eseguito l'installazione dell'unità nastro ausiliaria, procedere con il punto 12.
- ☐ 12. Se è stata installata la stampatrice, aprire il pacco della stampatrice e procedere con la Procedura di Installazione della stampatrice riportate più avanti in questa Appendice. Dopo aver installato la stampatrice, procedere al punto 13.
- ☐ 13. Se il 5100 è equipaggiato con un altro dispositivo qualsiasi far uso del manuale fornito assieme al dispositivo per installare e controllare il dispositivo, dopo di che tornare al punto 14 di questo manuale.

- ☐ 14. Quando le precedenti unità o dispositivi sono installati, oppure non lo sono affatto iniziare la lettura dell'IBM 5100 Introduzione (BASIC o APL) per imparare come usare il 5100.

Procedura di Installazione dell'unità nastro ausiliaria IBM 5106

- ☐ 1. Impostare gli interruttori del 5100 e dell'unità nastro ausiliaria su OFF.
- ☐ 2. Togliere il nastro adesivo per il trasporto dal cavo degli impulsi (cavo piatto) e connetterlo con il 5100. Esser certi che le connessioni siano effettuate efficacemente. Ruotare poi il pomello nel verso orario fino a che le connessioni siano saldamente fissate:

Posizione di parcheggio per il connettore di chiusura



- ☐ 3. Controllare che la spina del dispositivo sia a posto nel pannello posteriore come illustrato nel precedente diagramma.
- ☐ 4. Togliere il nastro adesivo per il trasporto del cavo di alimentazione ed inserire la spina in una presa di rete fornita di terra.
- ☐ 5. Portare l'interruttore di accensione dell'unità nastro ausiliaria su ACCESO (POWER ON) ed assicurarsi del buon funzionamento del ventilatore.
 - a. Se nella stanza in cui si lavora non c'è troppo rumore, si dovrebbe udire il rumore del ventilatore.
 - b. Se non si è certi del suo funzionamento, porre una striscia di carta vicino alla feritoia di ingresso dell'aria sul lato sinistro dell'unità nastro. La punta della striscetta dovrà essere risucchiata dall'unità nastro.

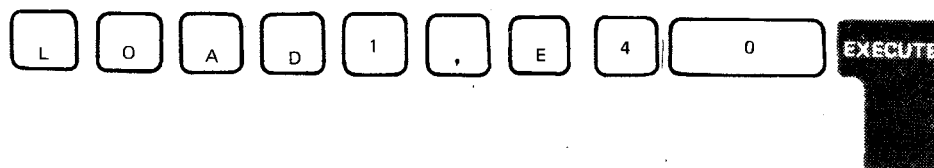
Se pare che il funzionamento del ventilatore sia negativo o scarso, controllare l'allacciamento della spina alla rete. Se il collegamento alla rete è buono, porre l'interruttore di accensione su SPENTO e chiamare la manutenzione. Interrompere poi la lettura di queste istruzioni.

- ☐ 6. Impostare l'interruttore di accensione del 5100 su ACCESO e proseguire con la procedura di controllo.

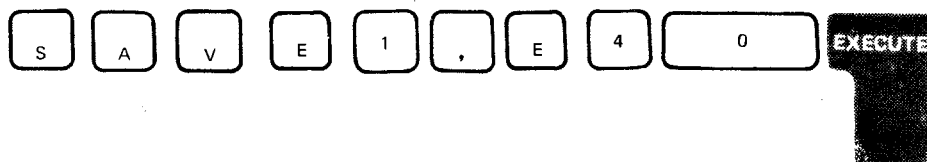
Procedura di controllo dell'Unità Nastro Ausiliaria

Nota: I seguenti passi presumono che si stia usando la stessa cassetta nastro usata per il controllo del 5100. Se non lo si fa, scrivere un qualsiasi programma nella cassetta nastro dell'unità nastro ausiliaria e leggerlo.

- ☐ 1. Inserire la cassetta a nastro dell'unità nastro ausiliaria, dopo aver verificato che la freccia di protezione si diretta dalla parte opposta della parola SAFE.
- ☐ 2. Battere i seguenti tasti per leggere il programma che è stato memorizzato su nastro durante la procedura di controllo del 5100:



- ☐ 3. Il nastro si sposterà, e quando verrà visualizzato nuovamente READY, battere i seguenti tasti:



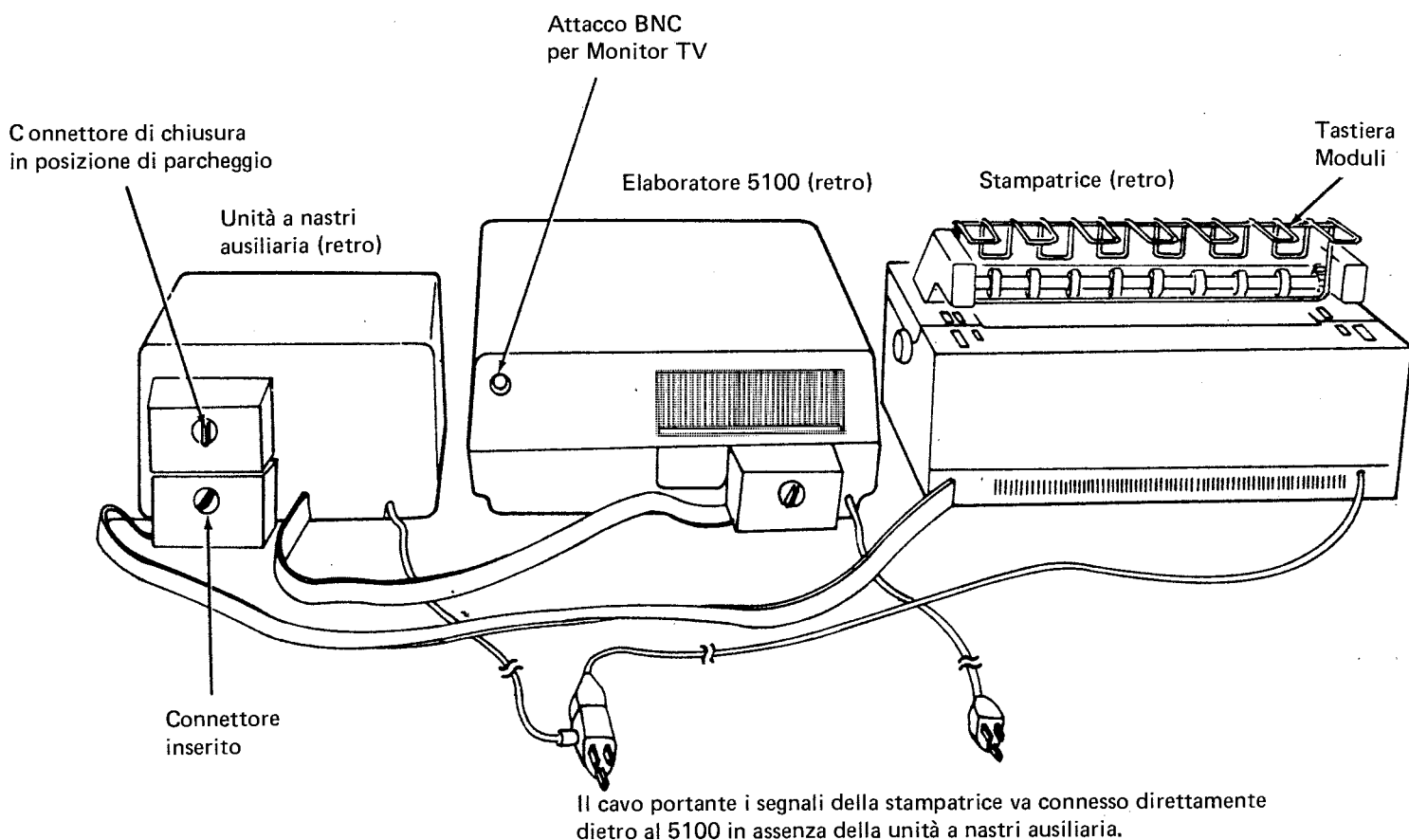
- ☐ 4. READY viene nuovamente visualizzato per verificare che il programma è stato riscritto sul nastro ed è stato controllato dal 5100.

Con ciò risulta completata la procedura di controllo dell'unità nastro ausiliaria.

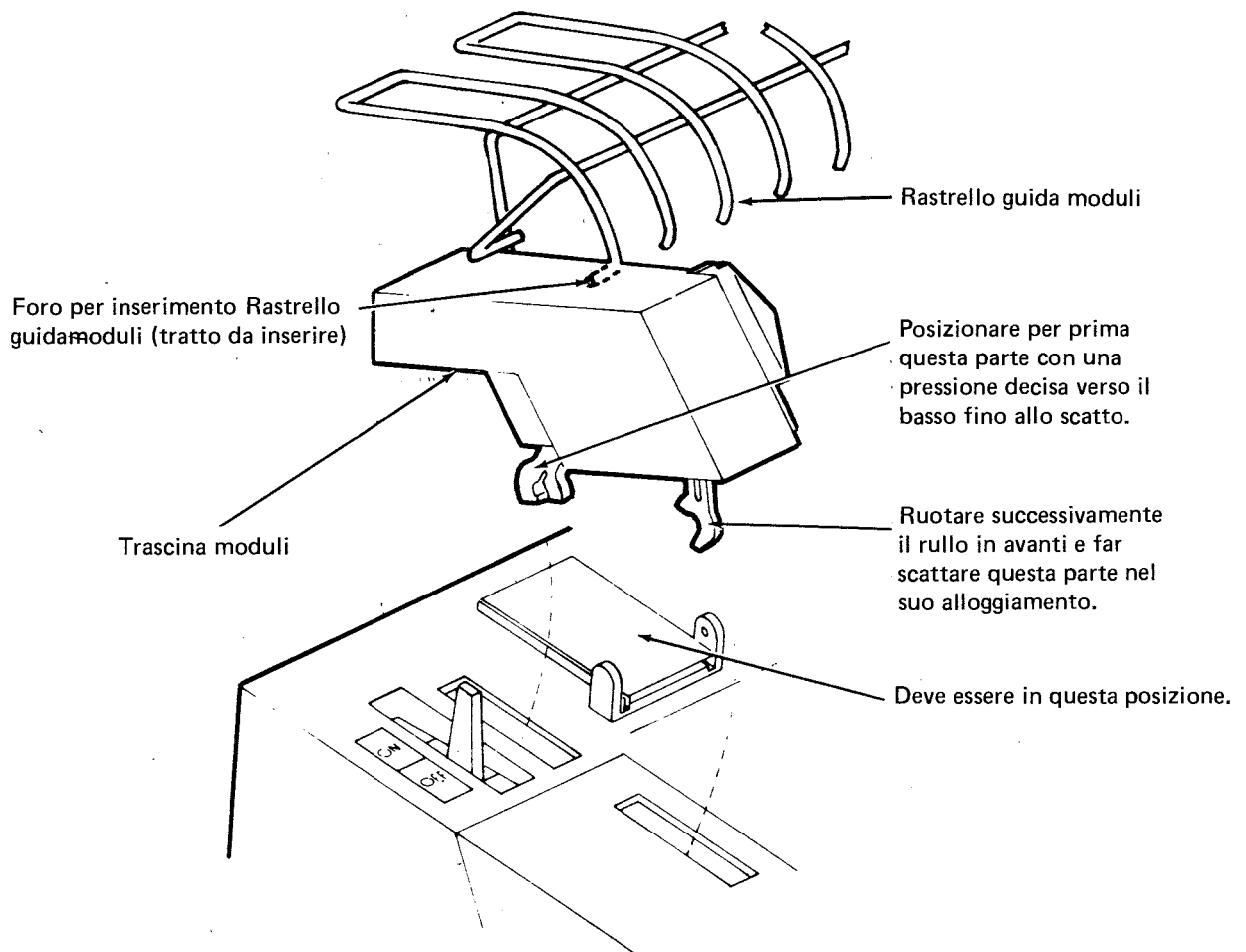
Ritornare al punto 12 della procedura di controllo del 5100.

Procedura di installazione della Stampatrice IBM 5103

- ☐ 1. Porre gli interruttori di accensione di tutte le unità su SPENTO o OFF.
- ☐ 2. Se si è già in possesso di una unità nastro ausiliaria togliere la presa del dispositivo dalla posizione inferiore ed inserirla nella posizione superiore (posizione di parcheggio).
- ☐ 3. Togliere il nastro adesivo di imballaggio dal cavo dei segnali della stampatrice (cavo piatto) e connettere questo cavo dei segnali alla parte bassa dell'unità nastro ausiliaria, se è collegata, oppure alla parte bassa del 5100. Esser certi che le connessioni siano fatte esattamente. Ruotare poi il pomello nel verso orario fino a che le connessioni siano saldamente fissate.
- ☐ 4. Togliere il nastro adesivo da imballo dal cavo di alimentazione della stampatrice ed inserire la spina nella spina di alimentazione dell'unità nastro ausiliaria, oppure inserirla autonomamente in una presa di rete provvista però di terra.



- ☐ 5. Disimballare trascina moduli e metterlo a posto nella parte superiore della stampatrice come illustrato nello schema. Inserire il rastrello guida dei moduli sul trascina moduli



- ☐ 6. Inserire la carta nella stampatrice. Utilizzare le informazioni sulla stampatrice che sono in questo manuale se si desidera un aiuto per inserire la carta (vedi Appendice D).
- ☐ 7. Spostare ambedue gli interruttori di accensione del 5100 e della stampatrice su ACCESO e poi continuare la procedura di controllo.

Procedura di Controllo della Stampatrice

Battere alcuni tasti alfanumerici per visualizzare qualche informazione. Poi, tenendo premuto il tasto CMD battere il tasto sotto COPIA

SCHERMO (COPY DISPLAY) posto sulla riga delle parole di comando. Allora la stampatrice fornirà una copia della informazione che si trova sullo schermo.

Ritornare al punto 13 della procedura di controllo del 5100.

Appendice D. Stampatrice 5103

La stampatrice 5106 possiede le seguenti caratteristiche:

Stampa bidirezionale (da sinistra a destra e da destra a sinistra). La testina scrivente si sposta dal margine sinistro e scrive una riga. Le righe successive saranno stampate in ambedue le direzioni in funzione di quale margine della riga si trova più vicino alla posizione corrente della testina scrivente.

Una riga di stampa può contenere un massimo di 132 caratteri.

Nota: Se vengono preparati per la stampa 132 caratteri per moduli che hanno una larghezza minore di 132 caratteri, si avranno dei danni (perdite) ai dati, allorchè la testina scrivente uscirà dai moduli.

Possibilità di utilizzare moduli individuali e continui. Al massimo sono consentite sei copie, però, onde non rischiare, si consiglia un massimo di quattro fogli per modulo.

Solo il trascina moduli di destra è spostabile per consentire l'uso di moduli di diverse larghezze. La larghezza per i moduli singoli può andare da 76.2 a 368.3, mentre per i moduli continui da 76.2 a 381 mm.

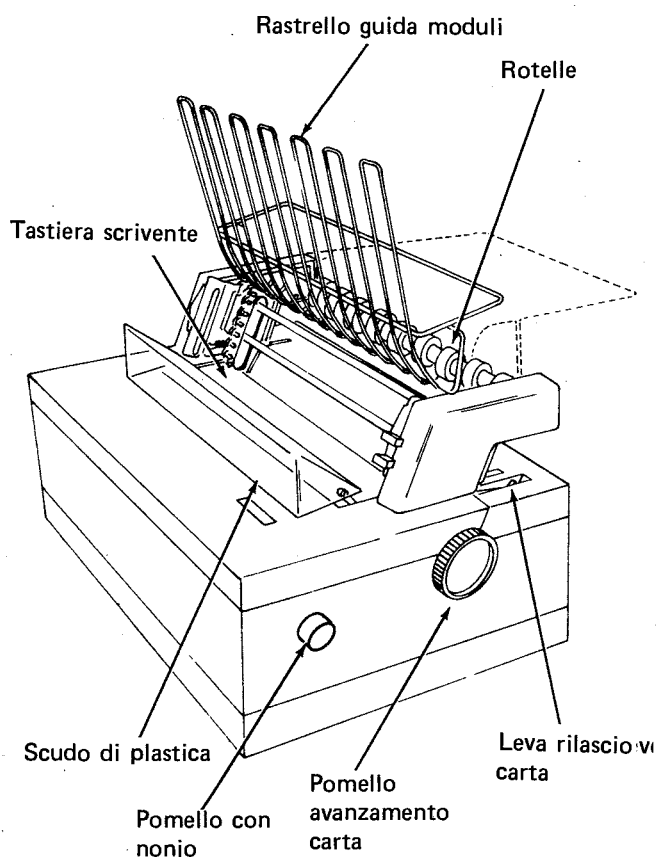
Spaziatura delle posizioni di stampa di 10 caratteri per pollice e spaziatura delle righe, di 6 righe per pollice.

Non si possono usare nè moduli tra loro cuciti, nè la scorta di normale carta continua.

La velocità di stampa è di 80 caratteri al secondo.

Un pomello con nonio (posto sul lato destro della stampatrice) che permette di regolare gli aggiustamenti fini della posizione di stampa. Questo pomello lo si dovrebbe usare solamente quando la testina scrivente si trova all'estrema posizione sinistra.

Come Inserire i Moduli

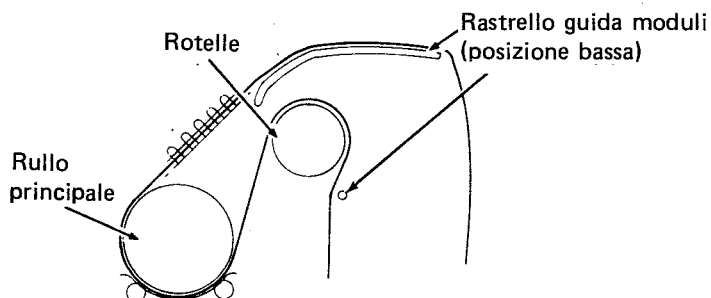


1. Ruotare in avanti lo scudo di plastica.
2. Spingere con un dito la testina scrivente nell'estrema posizione a sinistra.
3. Per moduli singoli, ruotare verso l'alto il braccio guida dei moduli e spingerlo in posizione verticale.
4. Spingere indietro la leva per il rilascio della carta onde attivare i rulli di alimentazione a frizione.
5. Sistemare i moduli in posizione dietro la stampatrice.

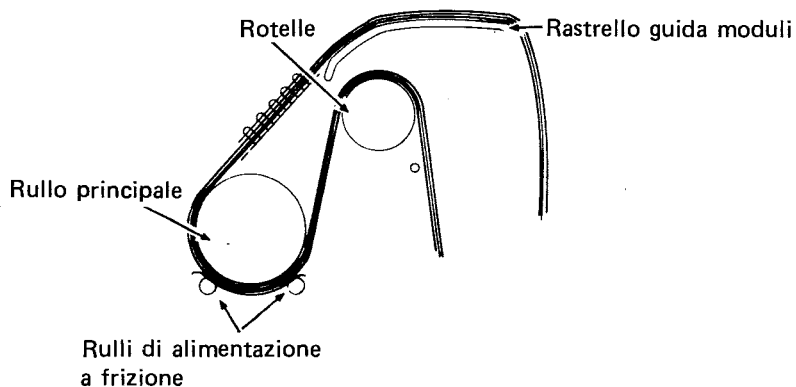
Nota: I moduli debbono essere sistemati dietro la stampatrice in maniera tale che i moduli vengano alimentati nella stampatrice ben adattati e allineati.

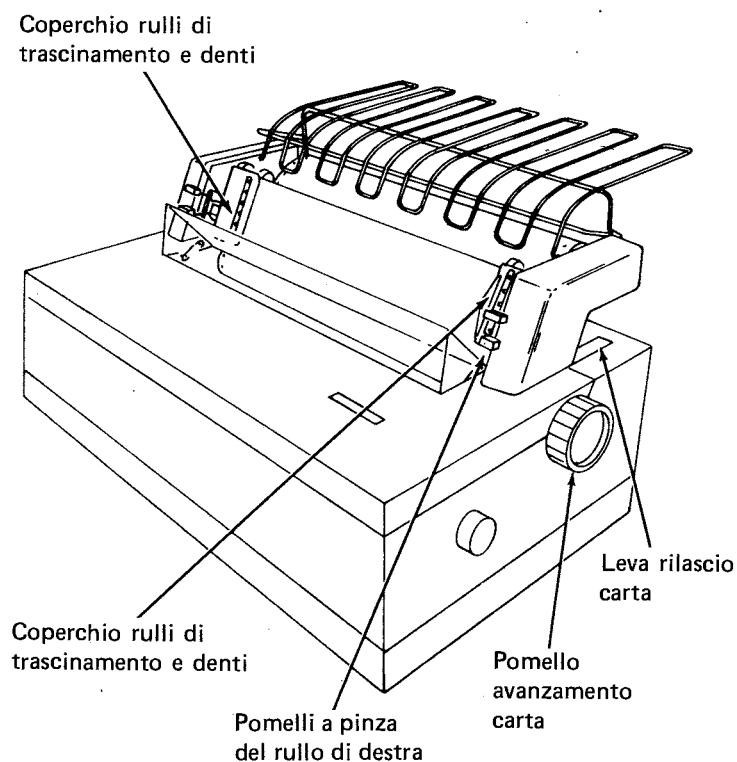
6. Spingere in basso la carta, passando però al disopra delle rotelle, dietro i rulli di trascinamento e dietro il rullo principale.
7. Ruotare il pomello di avanzamento della carta per spostare la carta intorno al rullo fino a che la si può afferrare con le dita.

Itinerario per moduli singoli



Itinerari per moduli multipli





8. Sollevare ambedue i coperchi metallici dei rulli di trascinamento.
9. Tirare in avanti la leva per il rilascio della carta per sganciare i rulli di alimentazione a frizione.
10. Tirare verso l'alto la carta e sistemare dapprima i fori del bordo sinistro sui denti del rullo di trascinamento. Assicurarsi che il rullo di trascinamento di sinistra sia nella sua posizione alla estremità sinistra.
11. Chiudere il coperchio del rullo di trascinamento di sinistra.
12. Stringere i due pomelli a pinza sul rullo di trascinamento di destra e far scorrere il rullo di trascinamento per allineare i denti del rullo con i fori del margine destro del modulo.
13. Sistemare quindi i fori del margine di destra del modulo sui denti del rullo di trascinamento.
14. Chiudere anche il coperchio metallico del rullo di trascinamento di destra.
15. Nel caso di moduli singoli ruotare nella posizione orizzontale il rastrello guida dei moduli.
16. Ruotare il pomello di avanzamento della carta fino a sistemare il modulo in modo che venga stampata la prima riga. La carta dovrebbe fuori uscire fino a sistemarsi sopra il rastrello guida dei moduli.

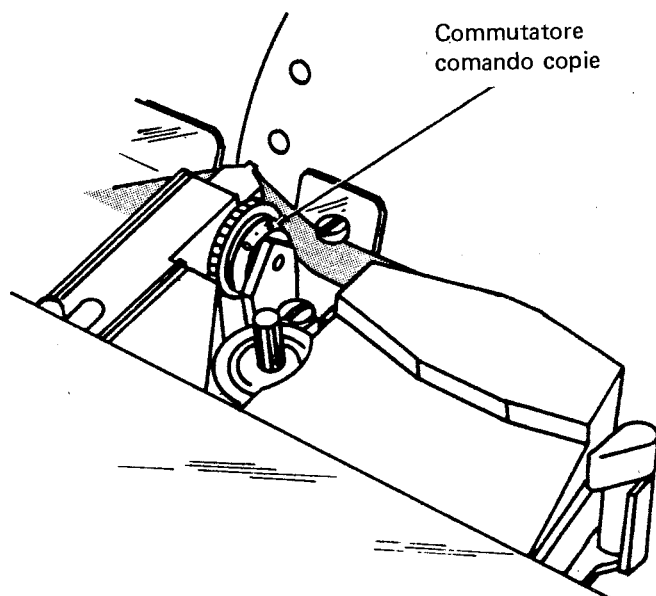
Nota: Per spostare i moduli indietro, ruotare all'indietro ambedue i pomelli di avanzamento della carta tirando contemporaneamente il modulo dietro la stampante cercando di evitare possibili accartocciamenti del modulo dietro la testina scrivente.

17. Chiudere lo scudo di plastica. Se si sta provvedendo all'installazione della stampatrice, ritornare al punto 7 della Procedura di Installazione della Stampatrice.

AVVERTIMENTO

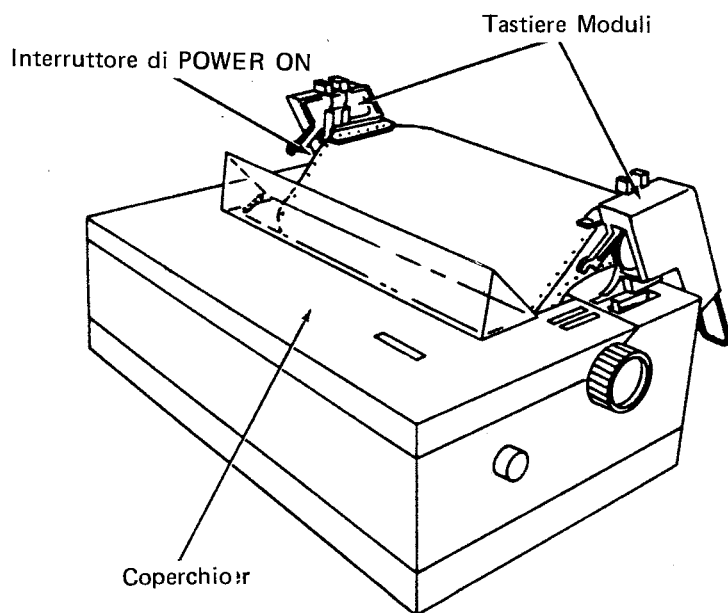
L'interruttore che individua la fine dei moduli va disattivato quando i rulli di alimentazione a frizione sono in funzione. Altrimenti gli aghi della testina scrivente potrebbe danneggiare la base del rullo di gomma se non vi fossero moduli inseriti nella stampatrice.

Come regolare il commutatore di comando delle copie in funzione dello spessore dei moduli

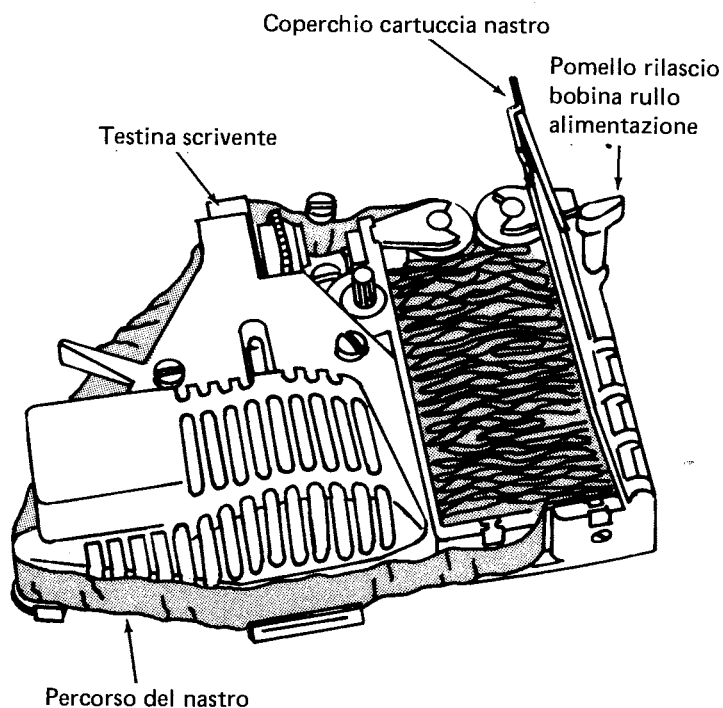


1. Allorchè si usano moduli singoli, impostare il commutatore comando copie su 0.
2. Allorchè si usano più moduli sovrapposti con carta carbone e l'ultimo foglio non risulta leggibile, si deve ruotare il commutatore di comando delle copie di 1 click per volta verso 0 per ottenere la leggibilità desiderata.
3. Se si usano più moduli sovrapposti ed il nastro sporca la prima copia, ruotare il commutatore di comando copie di un click per volta verso 8 fino a che l'imbrattamento finisce.

Come sostituire un nastro



1. Spegner la stampatrice.
2. Inclinare indietro il rullo di trascinamento dei moduli sollevando ambedue i lati anteriormente.
3. Sollevare il coperchio.



4. Esser sicuri che la testina scrivente sia spostata completamente a sinistra.

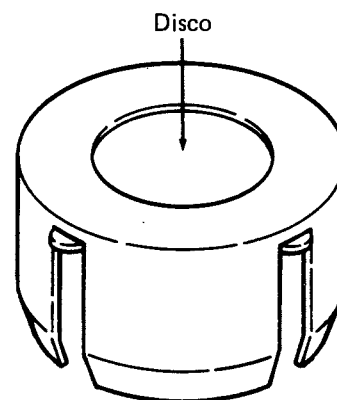
5. Ruotare il pomello per il rilascio della bobina del rullo di alimentazione in senso antiorario fino a che è diretto in avanti.

6. Aprire il coperchio della cartuccia del nastro.

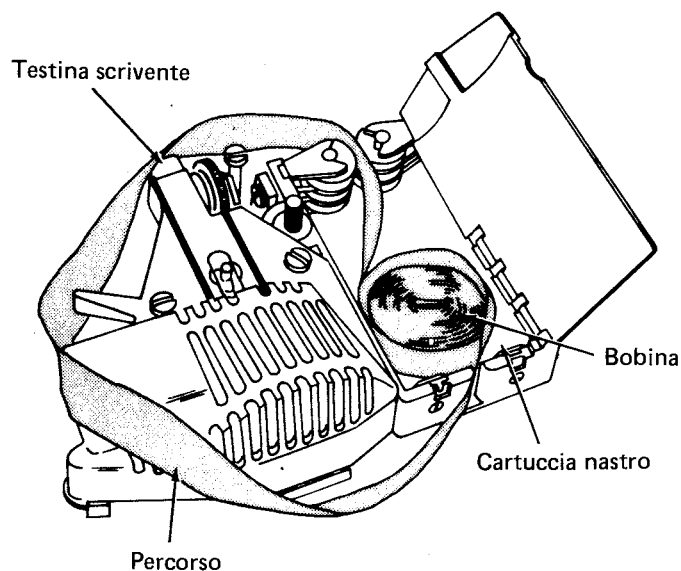
7. Indossare i guanti forniti insieme al nuovo nastro.

8. Estrarre il vecchio nastro dalle guide facendo attenzione a liberarlo dal morsetto sulla testina scrivente.

9. Richiamare tutto il nastro a partire dalla cima della cartuccia del nastro. Recuperare poi tutto il nastro e buttarlo.



Contenitore nastro

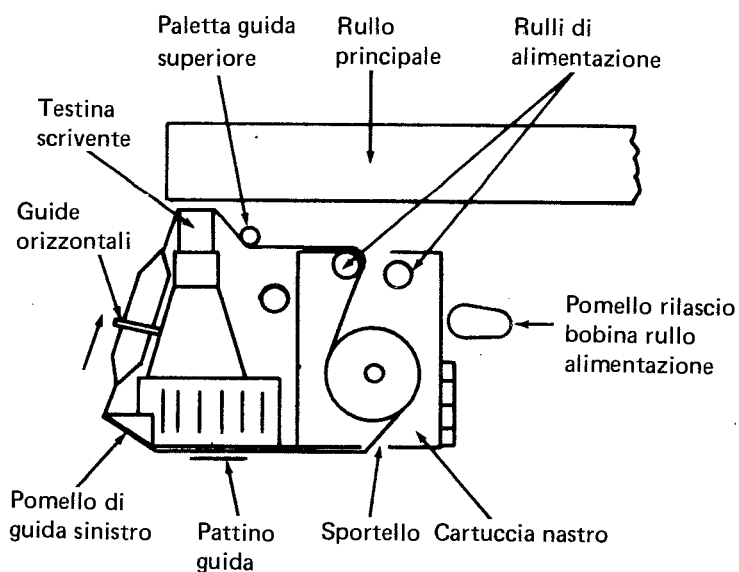


10. Espellere il nuovo nastro dal suo contenitore immettendolo nel vano della cartuccia premendo sul disco.

11. Togliere il disco dal nastro e buttare sia il disco che il contenitore.

12. Fermare leggermente con una mano la bobina di avvolgimento ed estrarre circa 25 cm di nastro dalla bobina.

13. Seguire il percorso prestabilito dalla bobina fino alla testina scrivente.



14. Infilare superiormente il nastro tra i rulli di alimentazione e la parte interna del paletto guida-superiore.
15. Ruotare il pomello di rilascio della bobina del rullo di alimentazione per serrare i rulli di alimentazione.
16. Infilare il nastro tra la testina scrivente ed il rullo. Esser sicuri che il nastro vada sotto il morsetto della testina scrivente.
17. Infilare inferiormente il nastro tra lo sportello, e la base della cartuccia nastro.
18. Infilare il nastro attraverso il pattino guida ed intorno al paletto sinistro di guida.
19. Inserire la parte orizzontale del nastro ruotato tra le due guide orizzontali.
20. Infine spostare indietro e avanti la testina scrivente verso il rullo per eliminare il gioco del nastro. Proseguire il movimento della testina stampante fino a che il nastro ruota esattamente. Lasciare poi la testina scrivente tutta a sinistra.
21. Chiudere il coperchio della cartuccia nastro.
22. Chiudere il coperchio della stampatrice ed accenderla.
23. Rimettere a posto il trascina moduli. di

IBM 5100

BASIC Manuale di riferimento

SA13-0086-00

I vostri commenti su questa pubblicazione potranno contribuire a migliorarla. Vi preghiamo di utilizzare lo spazio sottostante, citando ove possibile, i riferimenti alla pagina e al paragrafo; suggerimenti e commenti diventano proprietà della IBM.

Si prega di **non** utilizzare questo foglio per richiedere informazioni tecniche su sistemi e programmi o pubblicazioni; per tali esigenze si consiglia di rivolgersi al rappresentante IBM o alla Filiale IBM interessata. Se desiderate una risposta vogliate specificare il vostro nominativo, la mansione e l'indirizzo d'ufficio.

Nome:
Mansione/Titolo
Indirizzo

Piegare sulle linee tratteggiate, chiudere e spedire.
Grazie per la vostra collaborazione.

Gradiremmo i Vostri commenti

Questa pubblicazione fa parte di una biblioteca che costituisce la fonte di riferimento per analisi, 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

