

## 6. STATO CALCOLI IMMEDIATI

Il P6066 può essere impiegato per eseguire istantaneamente calcoli complessi senza dover utilizzare un programma BASIC. In questo caso il sistema deve essere nello stato calcoli immediati. Per commutare il sistema dallo stato comandi allo stato calcoli immediati si deve premere il tasto di console **CALC MODE**. Il sistema rimane nello stato suddetto finchè non si preme di nuovo il tasto **CALC MODE** che ricommuta il sistema nello stato comandi.

Lo stato calcoli immediati è naturalmente utile per la sua ovvia funzione: il sistema può essere usato come macchina da calcolo. Tuttavia, esso ha una seconda e forse più importante funzione: poichè le operazioni eseguite nello stato calcoli immediati non cancellano il contenuto della memoria principale, questo stato può essere un valido strumento per verificare i risultati ottenuti da un programma.

Molti programmi che sono eseguiti senza produrre alcuna segnalazione di errore possono, tuttavia, fornire dei risultati errati a causa della loro struttura logica. Fornendo la possibilità di verificare immediatamente i risultati di un programma -- mentre il programma è ancora nella memoria principale -- lo stato calcoli immediati offre un mezzo estremamente rapido e conveniente per scoprire degli errori di programmazione che altrimenti sarebbe difficile rilevare.

Quando il sistema P6066 è nello stato calcoli immediati si può introdurre:

- un comando di sistema
- un comando di richiamo di un programma di utilità
- una espressione da eseguire immediatamente

I comandi di sistema sono introdotti ed eseguiti come spiegato nel capitolo 3. Dopo che è eseguito un comando, il sistema ricommuta automaticamente nello stato comandi.

I comandi di richiamo dei programmi di utilità sono introdotti ed eseguiti come spiegato nella appendice A. Dopo che il programma è eseguito, il sistema ricomuta automaticamente nello stato comandi.

Le espressioni tipiche dello stato calcoli immediati sono descritte nei paragrafi che seguono.

Introduzione ed esecuzione di espressioni nello stato calcoli immediati

Quando il P6066 è nello stato calcoli immediati, si può richiedere al sistema di:

- eseguire una espressione numerica
- assegnare agli argomenti delle funzioni trigonometriche una particolare unità di misura (radianti, gradi sessagesimali o centesimali)
- assegnare un contenuto predefinito ai tasti funzione

Espressioni numeriche nello stato calcoli immediati

Una espressione numerica è composta da un insieme di operandi uniti da uno o più operatori. La valutazione di una espressione numerica fornisce come risultato un numero. Un operando numerico può essere una:

- costante numerica
- variabile numerica
- funzione numerica di sistema
- funzione numerica definita dall'utente
- espressione numerica racchiusa tra parentesi

Costanti numeriche

Una costante numerica, nel sistema P6066, è un numero espresso nel sistema decimale. Come si può vedere negli esempi seguenti, una costante numerica può essere introdotta, visualizzata o stampata con tre formati diversi: intero, virgola fissa o virgola mobile.

Formato intero      In virgola fissa      In virgola mobile

5	5.0	5.0 E +0
-5	-5.0	-5.0 E +0
+5	+5.0	+5.0 E +0

Formato intero

Una costante numerica espressa nel formato intero è composta da una o più cifre precedute, opzionalmente, dal segno. Da tastiera si possono introdurre costanti numeriche intere con al massimo 13 cifre. Il massimo numero che si può introdurre è : 9999999999999.

Formato in virgola  
fissa

Una costante numerica espressa nel formato in virgola fissa è composta da una o più cifre separate dal punto decimale e precedute, opzionalmente, dal segno. Una costante numerica nel formato in virgola fissa può essere introdotta da tastiera con non più di 13 cifre.

Nota: La costante  $\pi$  è una costante interna del sistema. Al contrario delle altre costanti numeriche essa è richiamata da tastiera mediante il nome PI. Il suo valore è 3.141592653590.

Formato in virgola  
mobile

Una costante numerica nel formato in virgola mobile (noto anche come "notazione scientifica") consiste in un numero intero od in virgola fissa seguito da E e da una o due cifre, opzionalmente precedute dal segno. Il massimo numero che si può introdurre da tastiera è: 9.999999999999E+99.

Rappresentazione interna

La rappresentazione interna di un numero è la forma con cui è rappresentato in memoria principale. Tutti i numeri sono rappresentati in memoria principale con due parti: la mantissa e l'esponente. Per esempio 5665 è rappresentato in memoria principale come: 5.665000000000E+03 di cui 5.665000000000 è la mantissa e 03 è l'esponente. La mantissa viene normalizzata in modo che essa sia sempre composta da una cifra compresa tra 1 e 9, seguita da 12 cifre decimali. Il valore rappresentato in memoria principale è dato dal prodotto della mantissa per 10 elevato all'esponente indicato.

Campo della rappresentazione  
interna

I numeri che possono essere elaborati devono rientrare nel campo di definizione da  $-9.999999999999E+99$  a  $-1E-99$ , lo zero, da  $+1E-99$  a  $+9.999999999999E+99$ , come indicato nella figura 6-1. La parte tratteggiata indica i numeri non definiti nella rappresentazione interna.

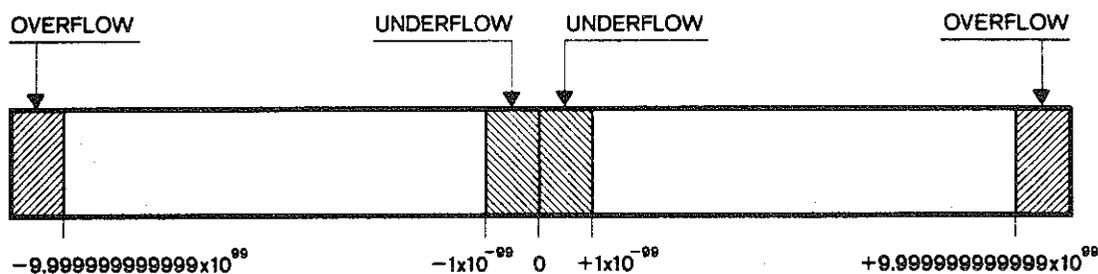


Figura 6-1 Campo della rappresentazione interna dei numeri

I numeri minori di  $9.999999999999 \times 10^{99}$  e maggiori di  $-9.999999999999 \times 10^{99}$  appartengono alle zone di OVERFLOW, ossia esprimono grandezze in valore assoluto più grandi di quelle rappresentabili in memoria principale.

I numeri maggiori di  $-1 \times 10^{-99}$  ed inferiori a  $+1 \times 10^{-99}$ , ma diversi da zero, appartengono alle zone di UNDERFLOW, ossia esprimono grandezze in valore assoluto più piccole di quelle rappresentabili in memoria principale.

Se durante l'elaborazione di una espressione si ottiene un risultato parziale con valore compreso in una zona di OVERFLOW, l'esecuzione della espressione è portata a termine assumendo come valore intermedio  $-9.999999999999 \times 10^{99}$  oppure  $9.999999999999 \times 10^{99}$ . Il risultato finale è stampato, se il tasto di console **PRINT ALL** è illuminato, ed il sistema visualizza il messaggio di errore ERROR 3.

Se durante l'elaborazione di una espressione si ottiene un risultato parziale con valore compreso in una zona di UNDERFLOW, l'esecuzione della espressione è portata a termine assumendo come valore intermedio zero. Il risultato finale è stampato, se il tasto di console **PRINT ALL** è illuminato, ed il sistema visualizza il messaggio di errore ERROR 4.

## Variabili numeriche

Una variabile numerica è un dato numerico, identificato con un nome, il cui valore è soggetto a cambiare durante l'esecuzione di una espressione. Nello stato calcoli immediati si possono utilizzare quattro variabili numeriche il cui nome è:  $\Phi$ ,  $\Phi\emptyset$ ,  $\Phi 1$ ,  $\Phi 2$ .

Per assegnare, da tastiera, un valore numerico ad una variabile, si deve premere il tasto **RESULT** ed, eventualmente, una delle tre cifre:  $\emptyset$ , 1 o 2; quindi il tasto **=** seguito dal numero da assegnare alla variabile.

Vediamo, ad esempio, come si assegna il valore 10 ad ognuna delle 4 variabili e cosa appare sul display come risultato di ogni assegnazione:

Quando si preme	Sul display appare
<b>RESULT</b> <b>=</b> 1 0 <b>END OF LINE</b>	$\Phi = 1\emptyset$
<b>RESULT</b> 0 <b>=</b> 1 0 <b>END OF LINE</b>	$\Phi \emptyset = 1\emptyset$
<b>RESULT</b> 1 <b>=</b> 1 0 <b>END OF LINE</b>	$\Phi 1 = 1\emptyset$
<b>RESULT</b> 2 <b>=</b> 1 0 <b>END OF LINE</b>	$\Phi 2 = 1\emptyset$

Per assegnare un valore ad una variabile come risultato dell'esecuzione di una espressione, si deve introdurre il nome della variabile seguita dal segno uguale e dall'espressione. Per esempio, premendo:

**RESULT** 2 **=** **RESULT** + **RESULT** 0 **+** **RESULT** 1 **-** **RESULT** 2 **END OF LINE**

il risultato della esecuzione dell'espressione  $\Phi + \Phi\emptyset + \Phi 1 - \Phi 2$  è assegnato alla variabile  $\Phi 2$ .

Vi sono due modi per visualizzare il valore corrente di una variabile. Con il primo si visualizza il valore della variabile e contemporaneamente si assegna tale valore alla variabile  $\Phi$ ; con il secondo si visualizza soltanto il valore della variabile.

1. Per visualizzare il valore di una variabile ed assegnare il suo valore alla variabile  $\Phi$ , si introduce il nome della variabile e si preme **END OF LINE**. Per esempio, per visualizzare il valore della variabile  $\Phi\emptyset$  ed assegnarlo alla variabile  $\Phi$  si preme:

**RESULT** 0 **END OF LINE**

2. Per visualizzare il valore di una variabile senza variare il valore della variabile  $\Phi$ , si utilizzi il seguente formato: variabile=variabile e quindi si preme **END OF LINE**. Per esempio, per visualizzare il valore della variabile  $\Phi$  senza modificare il valore di  $\Phi$  si preme: **RESULT** **0** **=** **RESULT** **0** **END OF LINE**

Nota: Ogni volta che si commuta il sistema nello stato calcoli immediati (premendo **CALC MODE**), le quattro variabili suddette sono eguagliate a zero. Se il sistema è nello stato calcoli immediati e si vogliono eguagliare a zero le quattro variabili contemporaneamente, si preme due volte **CALC MODE** —la prima volta si esce dallo stato calcoli immediati, la seconda volta vi si rientra.

La variabile  $\Phi$  come totalizzatore

Se si introduce un valore numerico od una espressione numerica da tastiera premendo il tasto **SUM**, invece del tasto **END OF LINE**, la variabile  $\Phi$  si comporta come un totalizzatore (accumulatore). Il valore introdotto od il risultato della espressione è aggiunto al valore precedente della variabile  $\Phi$ .

L'impiego della variabile  $\Phi$  come totalizzatore è mostrato negli esempi che seguono (si ricordi che quando si preme il tasto **CALC MODE**, per passare nello stato calcoli immediati, tutte le variabili sono automaticamente eguagliate a zero).

Quando si preme	Si pone
<b>CALC MODE</b>	$\Phi = 0, \Phi 0 = 0, \Phi 1 = 0, \Phi 2 = 0$
<b>RESULT</b> <b>=</b> <b>3</b> <b>END OF LINE</b>	$\Phi = 3$
<b>RESULT</b> <b>=</b> <b>9</b> <b>SUM</b>	$\Phi = 12$

Come si vede, premendo **SUM** si aggiunge 9 al precedente valore di  $\Phi$ .

Quando si preme	Si pone
<b>CALC MODE</b>	$\Phi = 0, \Phi 0 = 0, \Phi 1 = 0, \Phi 2 = 0$
<b>RESULT</b> <b>=</b> <b>1</b> <b>0</b> <b>END OF LINE</b>	$\Phi = 10 \quad \Phi 0 = 0$
<b>RESULT</b> <b>0</b> <b>=</b> <b>1</b> <b>0</b> <b>SUM</b>	$\Phi = 20 \quad \Phi 0 = 10$

Come si vede, premendo **SUM** per assegnare il valore 10 a  $\Phi$ , si aggiunge anche 10 al precedente valore di  $\Phi$ .

Quando si preme

Si pone

**CALC  
MODE**

$\Phi=0$ ,  $\Phi0=0$ ,  $\Phi1=0$ ,  $\Phi2=0$

RESULT 1 = 1 0 END OF LINE

$\Phi=0$   $\Phi1=10$

RESULT 2 = 1 0 0 END OF LINE

$\Phi=0$   $\Phi1=10$   $\Phi2=100$

RESULT = 2 0 END OF LINE

$\Phi=20$   $\Phi1=10$   $\Phi2=100$

RESULT = RESULT 1 + RESULT 2 SUM

$\Phi=130$   $\Phi1=10$   $\Phi2=100$

Come si vede, premendo **SUM** si aggiunge il valore di  $\Phi1 + \Phi2$  (ossia 110) al precedente valore di  $\Phi$  (ossia 20).

Quando si preme

Si pone

**CALC  
MODE**

$\Phi=0$ ,  $\Phi0=0$ ,  $\Phi1=0$ ,  $\Phi2=0$

RESULT 1 = 1 0 END OF LINE

$\Phi=0$   $\Phi1=10$

RESULT 2 = 1 0 0 END OF LINE

$\Phi=0$   $\Phi1=10$   $\Phi2=100$

RESULT = 2 0 END OF LINE

$\Phi=20$   $\Phi1=10$   $\Phi2=100$

RESULT 2 = RESULT 1 + RESULT 2 SUM

$\Phi=130$   $\Phi1=10$   $\Phi2=110$

Come si vede, premendo **SUM** non solo si assegna il valore di  $\Phi1 + \Phi2$  a  $\Phi2$ , ma si aggiunge anche tale valore al precedente valore di  $\Phi$ .

Per riassumere brevemente, il tasto **END OF LINE** serve per assegnare un valore ad una variabile; il tasto **SUM** serve sia per assegnare un valore ad una variabile che, contemporaneamente, per aggiungere tale valore al precedente valore della variabile  $\Phi$ .

## Operatori numerici

Gli operatori numerici definiscono quale operazione deve essere eseguita sui valori numerici degli operandi specificati. Essi producono come risultato un numero. Gli operatori numerici che si possono utilizzare sono:

## Operatore numerico

## Funzione

↑	Elevamento a potenza
/	Divisione
*	Moltiplicazione
+	Addizione e segno più
-	Sottrazione e segno meno

Le espressioni numeriche sono eseguite secondo il livello di priorità degli operatori che le costituiscono. Le operazioni con il più alto livello di priorità sono eseguite per prime; quelle con lo stesso livello di priorità sono eseguite nell'ordine da sinistra a destra.

Il P6066 osserva le regole dell'algebra per la definizione del livello di priorità di esecuzione di una operazione nell'ambito di una espressione numerica, per cui i livelli di priorità sono:

## Operatore numerico

## Livello di priorità

↑	Il più alto
* e /	↓
+ e -	Il più basso

Le parentesi ( ) e ( ) possono essere usate per cambiare l'ordine di esecuzione delle operazioni nell'ambito di una espressione numerica. Una espressione racchiusa tra parentesi è trattata come un singolo elemento numerico: viene valutata per ottenere il suo valore numerico, quindi tale valore è utilizzato per la valutazione della parte restante di una espressione più complessa di cui essa fa parte. Se più di una espressione è compresa tra parentesi, il calcolo inizia con la valutazione delle parentesi più interne. Nel seguito diamo ulteriori informazioni relative agli operandi numerici.

Elevamento a potenza

RESULT (1) ↑ RESULT (2)

Indica che il valore di  $\Phi 1$  deve essere elevato alla potenza di esponente dato da  $\Phi 2$

Se  $\Phi 1 = \Phi 2 = \emptyset$

$\Phi 1 \uparrow \Phi 2$  è uguale ad uno

Se  $\Phi_1 = 0$  e  $\Phi_2 < 0$

Si ha una segnalazione di OVERFLOW

Se  $\Phi_1 < 0$  e  $\Phi_2$  non è intero

Si ha una segnalazione di errore

Se  $\Phi_1 = 0$  e  $\Phi_2 > 0$

$\Phi_1 \uparrow \Phi_2$  è uguale a zero

### Moltiplicazione ed addizione

RESULT 1 \* RESULT 2

equivale a RESULT 2 \* RESULT 1

RESULT 1 + RESULT 2

equivale a RESULT 2 + RESULT 1

Come si vede, per la moltiplicazione e l'addizione, vale la proprietà commutativa.

RESULT 1 \* ( RESULT 2 \* RESULT )

non equivale sempre a

( RESULT 1 \* RESULT 2 ) \* RESULT

RESULT 1 + ( RESULT 2 + RESULT )

non equivale sempre a

( RESULT 1 + RESULT 2 ) + RESULT

perchè l'operazione tra parentesi in alcuni casi può dare un risultato troncato od arrotondato.

### Divisione e sottrazione

RESULT 1 / RESULT 2

Indica  $\Phi_1$  diviso per  $\Phi_2$

Se  $\Phi_2 = 0$

Si ha segnalazione di OVERFLOW

RESULT 1 - RESULT 2

Indica  $\Phi_1$  meno  $\Phi_2$

### Segno

- RESULT + ( - RESULT 1 ) + RESULT 2 - ( - RESULT 0 )

è ammesso

RESULT 1 + - RESULT 2

non è ammesso

Il segno + ed il segno - possono essere usati dopo una parentesi aperta o prima di una espressione numerica.

Nota: Due operatori numerici non possono essere usati uno di seguito all'altro ma devono essere separati da parentesi.

## Funzioni

Nello stato calcoli immediati si possono utilizzare funzioni numeriche di sistema o definite dall'utente.

### Funzioni numeriche di sistema

Le funzioni numeriche di sistema possono essere utilizzate introducendo da tastiera il loro nome e (tra parentesi) l'argomento su cui si applicano. L'argomento può essere una costante numerica, una variabile numerica o, in generale, una espressione numerica.

Nella tabella seguente sono riassunte, in ordine alfabetico, le funzioni numeriche di sistema disponibili; la lettera X indica l'argomento della funzione.

Funzione numerica	Descrizione
ABS(X)	Valore assoluto di X
ACS(X)	Arcocoseno (in radianti) di X
ASN(X)	Arcoseno (in radianti) di X
ATN(X)	Arcotangente (in radianti) di X
COS(X)	Coseno di X radianti
COT(X)	Cotangente di X radianti
EXP(X)	Esponenziale in base e di X
HCS(X)	Coseno iperbolico di X radianti
HSN(X)	Seno iperbolico di X radianti
HTN(X)	Tangente iperbolica di X radianti
INT(X)	Il più grande numero intero minore od uguale ad X
LGT(X)	Logaritmo in base 10 di X
LOG(X)	Logaritmo naturale di X
RND	Numero casuale compreso tra <u>zero</u> ed <u>uno</u>
SGN(X)	Segno di X (+1 per X positivo, 0 per 0, -1 per X negativo)

SIN(X)	Seno di X radianti
SQR(X)	Radice quadrata di X
TAN(X)	Tangente di X radianti

Tabella 6-1 Funzioni numeriche di sistema

Si possono inoltre utilizzare le seguenti funzioni:

LEN (string-exp) che ritorna il numero di caratteri di string-exp

SCN (string-exp, substring, n-occurrence, start-position) che ritorna la posizione della ennesima occorrenza in string-exp della sottostringa substring a partire dalla posizione specificata con start-position.

Funzioni definite dall'utente

Il P6066 permette di definire le funzioni di frequente impiego e di assegnarle ad uno degli 8 tasti della sezione funzioni definibili.

Si possono assegnare due funzioni per ogni tasto suddetto (utilizzando il tasto SHIFT) quindi si ha la possibilità di avere contemporaneamente fino a 16 funzioni. Ogni funzione può essere definita con al massimo 73 caratteri, ma il numero totale dei caratteri che definiscono tutte le funzioni assegnate ai tasti funzione non può essere maggiore di 238.

Vediamo con un esempio come si definisce ed assegna una funzione ad un tasto funzione, mentre il sistema è nello stato calcoli immediati.

Supponiamo di voler calcolare le radici reali di una equazione di secondo grado del tipo  $ax^2+bx+c$  utilizzando i tasti funzione? Si deve procedere nel seguente modo. Alla variabile  $\Phi 2$  si assegneranno i coefficienti di tipo a. Alla variabile  $\Phi 1$  si assegneranno i coefficienti di tipo b. Alla variabile  $\Phi 0$  si assegneranno i coefficienti di tipo c. Quindi alla chiave F1 si assegni la stringa:

$$(-\Phi 1 + \text{SQR}(\Phi 1 * \Phi 1 - 4 * \Phi 2 * \Phi 0)) / (2 * \Phi 2)$$

premendo i tasti SHIFT FKEY# 1 e di seguito i tasti equivalenti alla stringa suddetta, seguiti da END OF LINE  
 Alla chiave F2 si assegni la stringa:

$$(-\Phi_1 - \text{SQR}(\Phi_1 * \Phi_1 - 4 * \Phi_2 * \Phi_0)) / (2 * \Phi_2)$$

premendo i tasti     e di seguito i tasti equivalenti alla stringa suddetta, seguiti da 

Per calcolare le radici reali di una equazione particolare, si introducano da tastiera i valori dei coefficienti assegnandoli nell'ordine alle variabili:

$\Phi_2 = a$   
 $\Phi_1 = b$   
 $\Phi_0 = c$

Dopo di che premendo:    

si ottiene, assegnandolo a  $\Phi$ , il valore di una radice.

Premendo:    

si ottiene il valore della seconda radice.

Con analoghe procedure si possono assegnare ai tasti funzione altre espressioni numeriche. Premendo un solo tasto tali espressioni sono richiamate nel buffer di tastiera ed eventualmente inserite in espressioni più complesse.

Nota: Nello stato calcoli immediati si possono utilizzare, premendo i tasti funzione, le stesse espressioni assegnate ad essi dal programma presente in memoria principale.

### Visualizzazione dei risultati

Il risultato della esecuzione di una espressione è visualizzato immediatamente sul display e, se il tasto di console  è acceso, stampato sul tabulato della stampante integrata. I risultati possono essere visualizzati (o stampati) in diversi formati che dipendono dalla posizione dell'indicatore dei decimali (vedi capitolo 1).

1. Quando l'indicatore dei decimali è nella posizione 0 (lo 0 è visibile nella relativa finestrella) non vengono visualizzate (stampate) le cifre decimali.
2. Quando l'indicatore dei decimali è in una posizione da 1 a 13 (la relativa cifra è visibile nella suddetta finestrella), vengono visualizzate (stampate)

da 1 a 13 cifre dopo il punto decimale se ciò è possibile.

3. Quando l'indicatore dei decimali è nella posizione Flt (Flt è visibile nella relativa finestrella), il numero è visualizzato nel formato interno ossia nella notazione scientifica (vedi paragrafo "Rappresentazione interna").
4. Quando l'indicatore dei decimali è nella posizione ST (ST è visibile nella finestrella):
  - i numeri interi sono visualizzati (stampati) con al massimo 8 cifre significative
  - i numeri con valore assoluto compreso tra 0.0999999995 e 99999999.4 sono visualizzati (stampati), con al massimo 8 cifre significative (se il numero è minore di 1 viene tralasciato lo zero che precede la parte decimale), nel formato in virgola fissa
  - i numeri con valore assoluto minore di 0.0999999995, che possono essere rappresentati con 8 cifre significative, sono visualizzati (stampati) nel formato in virgola fissa
  - tutti gli altri numeri sono rappresentati con al massimo 8 cifre significative nel formato in virgola mobile

Nota: L'indicatore dei decimali controlla anche il formato di visualizzazione e stampa del contenuto delle variabili numeriche  $\bar{\Phi}$ ,  $\bar{\Phi}\bar{\Phi}$ ,  $\bar{\Phi}1$ ,  $\bar{\Phi}2$ .

#### L'unità di misura degli angoli

Nel sistema P6066 gli angoli sono misurati normalmente in radianti; quindi, quando si forniscono gli argomenti alle funzioni trigonometriche, i valori introdotti sono assunti dal sistema come espressi in radianti. (Anche il valore ritornato dalla funzione ATN è espresso in radianti.)

Nello stato calcoli immediati è possibile scegliere l'unità di misura degli angoli, comunicando al sistema quella voluta, mediante la digitazione di uno dei seguenti comandi di predisposizione: SDEG, SGRAD, SRAD. (Quando si introduce SDEG e SGRAD, la luce di console

DEG GRAD si accende. Essa rimane accesa finchè si introduce SRAD, si esce dallo stato calcoli immediati, oppure il P6066 è spento.) Se si preme **S D E G**, i valori assegnati come argomenti alle funzioni trigonometriche sono misurati in gradi sessagesimali. (Anche il valore ritornato dalla funzione ATN è espresso in gradi sessagesimali.)

Si noti che i valori assegnati agli argomenti delle funzioni trigonometriche, dopo aver introdotto SDEG, non possono essere espressi in gradi, primi e secondi di grado, ma in termini di gradi, decimi, centesimi, millesimi etc. di grado.

Se si preme **S G R A D**, i valori assegnati come argomenti alle funzioni trigonometriche sono interpretati in gradi centesimali. (Anche il valore ritornato dalla funzione ATN è espresso in gradi centesimali.)

Se si preme **S R A D**, i valori assegnati agli argomenti delle funzioni trigonometriche sono interpretati in radianti. (Anche ATN ritorna un valore espresso in radianti.) Se si preme due volte **CALC MODE** si ottiene lo stesso risultato, ma le variabili  $\Phi$ ,  $\Phi\Phi$ ,  $\Phi1$  e  $\Phi2$  sono eguagliate a zero.

### Esempi di calcoli immediati

1. I seguenti esempi mostrano come l'impiego delle parentesi influisce sulla valutazione di una espressione:

Premendo	Sul display appare
$(4 + 3 \cdot 2 - 6) / 2$ <b>END OF LINE</b>	7
$(4 + 3) \cdot 2 - 6 / 2$ <b>END OF LINE</b>	11
$(4 + 3 \cdot 2 - 6) / 2$ <b>END OF LINE</b>	2
$((4 + 3) \cdot 2 - 6) / 2$ <b>END OF LINE</b>	4
$2 \uparrow 3 \uparrow 2$ <b>END OF LINE</b>	64

2 ↑ ( 3 ↑ 2 ) END OF LINE

2. L'esempio seguente mostra come si può calcolare la media di cinque numeri:

Premendo

RESULT = 0 END OF LINE

$\Phi$  è uguagliata a zero

5 SUM

$\Phi = 5$

6 SUM

$\Phi = 11$

1 4 SUM

$\Phi = 25$

3 SUM

$\Phi = 28$

1 2 SUM

$\Phi = 40$

RESULT / 5 END OF LINE

$\Phi = 8$

3. Negli esempi seguenti si mostra l'effetto prodotto posizionando l'indicatore dei decimali nelle posizioni indicate:

Premendo

Con l'indicatore dei decimali su

Sul display appare

P I END OF LINE

$\emptyset$

3

P I END OF LINE

1

3.1

P I END OF LINE

2

3.14

P I END OF LINE

13

3.1415926535900

P I END OF LINE

ST

3.1415927

P I END OF LINE

Flt

3.141592653590E+00



## 7. LO STATO DI DEBUGGING

### Premessa

Fornendo una verifica sintattica immediata di ogni linea introdotta da tastiera e fornendo una verifica della coerenza sintattica tra le istruzioni di un intero programma, durante la fase di pre-esecuzione, il P6066 riduce il tempo necessario per individuare, diagnosticare e correggere gli errori in un programma. Quando, tuttavia, un programma è eseguito, è necessario determinare se funziona correttamente. Vi possono essere, infatti, degli errori di tipo logico nella sua struttura.

Per verificare la validità di un programma, lo si deve eseguire con dei dati di prova che producano dei risultati predefiniti. Si paragonano, quindi, questi ultimi con i risultati prodotti dal P6066. Se i risultati ottenuti non coincidono con quelli previsti ed il programma è breve o abbastanza semplice, si può, di solito, trovare la causa degli errori analizzando il programma stesso: l'ordine delle istruzioni, la correttezza del suo algoritmo, etc.; ma, se un programma è lungo e complesso, il semplice metodo di paragonare i risultati ottenuti con quelli previsti non sempre permette di trovare la causa degli errori: sarà necessaria una ulteriore analisi.

Lo stato di debugging del P6066 offre un insieme completo di strumenti per l'analisi e la verifica dei programmi. Nello stato di debugging, l'operatore ha un completo controllo sulla esecuzione di un programma. Si può fermare e riprendere l'esecuzione del programma, si possono visualizzare e modificare i valori delle variabili, si può controllare il flusso logico del programma, si può, insomma, vedere e tracciare l'esecuzione del programma.

### Come accedere allo stato di debugging

Il sistema commuta nello stato di debugging quando: (1) si esegue il comando PREPARE, (2) viene eseguita una istruzione STOP, (3) si rileva un errore di tipo recuperabile (vedi appendice D) durante l'esecuzione di un programma,

oppure (4) si preme il tasto di console **STEP**. Quando il sistema è nello stato di debugging, il tasto di console **STEP** è illuminato.

1. Il comando PREPARE (vedi capitolo 3) può essere introdotto durante lo stato comandi oppure durante lo stato calcoli immediati. Il programma presente in memoria principale è pre-eseguito ed il sistema commuta nello stato di debugging. Il tasto di console **STEP** si illumina.
2. Quando si esegue l'istruzione STOP, l'esecuzione del programma di cui essa fa parte è interrotta ed il sistema commuta nello stato di debugging. Sul display appare il messaggio STOP IN LINE line-num (dove line-num è il numero di linea dell'istruzione STOP).
3. Se, durante l'esecuzione di un programma, viene rilevato un errore di tipo recuperabile, l'esecuzione del programma è interrotta, il sistema commuta nello stato di debugging e viene visualizzato un appropriato messaggio di errore sul display.
4. Quando si preme il tasto di console **STEP**, l'esecuzione di un programma viene interrotta. Il sistema commuta nello stato di debugging e sul display appare il messaggio STEP IN LINE line-num (dove line-num è il numero di linea della successiva istruzione da eseguire). Se si preme di nuovo il tasto **STEP**, viene eseguita l'istruzione successiva, il sistema commuta di nuovo nello stato di debugging e un nuovo messaggio STEP IN LINE line-num appare sul display.

### Strumenti dello stato di debugging

Quando il sistema è nello stato di debugging, si possono utilizzare i seguenti strumenti per ricercare la causa degli errori di un programma presente in memoria principale:

1. Comandi dello stato di debugging

START  
STOP

## 2. Tasti di console

CONTINUE  
STEP  
TRACE

## 3. Prestazioni dello stato calcoli immediati.

## 4. Analisi ed impiego delle variabili globali di programma.

## 5. Impiego delle funzioni definite nel programma.

Comandi dello stato di debugging

I comandi START e STOP possono essere utilizzati durante lo stato di debugging per verificare parti di un programma senza doverlo eseguire completamente.

### 1. Comando START

Il comando START comunica al sistema di riprendere l'esecuzione di un programma, precedentemente interrotto, da un numero di linea specificato. La sintassi del comando è:

**STA [RT] line-num**

dove line-num è il numero di linea della istruzione da cui riprende l'esecuzione del programma.

### 2. Comando STOP

Il comando STOP comunica al sistema di interrompere l'esecuzione del programma presente in memoria principale dopo l'esecuzione di una istruzione predefinita e di commutare nello stato di debugging. La sintassi del comando è:

**STO [P] line-num**

dove line-num è l'istruzione in cui si interrompe l'esecuzione del programma. Si noti che se si introducono successivamente due comandi STOP line-num, l'ultima introduzione invalida la predisposizione alla sospensione della esecuzione comandata dalla precedente. Se nel programma non esiste una istruzione con il numero di linea line-num specificato nel comando STOP, oppure durante l'esecuzione non è mai incontrata una istruzione con tale numero di linea, l'esecuzione prosegue fino al termine

senza che sia prodotta alcuna segnalazione. Si noti infine che il comando STOP line-num è un comando di predisposizione per cui quando è introdotto da tastiera l'esecuzione del programma non riprende; per far ripartire l'esecuzione del programma si deve premere **CONTINUE** o **STEP** o introdurre il comando START.

## Tasti di console

Tre tasti di console sono particolarmente utili come strumenti di verifica dei programmi: **CONTINUE**, **STEP** e **TRACE**.

Nel seguito è spiegato il loro impiego e sono dati alcuni suggerimenti che permettono di utilizzare, nello stato di debugging, le prestazioni offerte dai tasti di console **NO PRINT** e **BREAK**.



Se mentre il sistema è nello stato di debugging, è premuto il tasto di console **CONTINUE**, riprende l'esecuzione del programma che è in memoria principale. Il tasto **CONTINUE** si illumina quanto è premuto mentre il sistema è nello stato di debugging. La funzione del tasto è attiva solamente quando il sistema si trova nello stato di debugging.



Se mentre il sistema è nello stato di debugging, è premuto il tasto di console **STEP**, si può eseguire passo a passo (ossia istruzione per istruzione) il programma presente in memoria principale. Ogni volta che una istruzione del programma è eseguita il sistema visualizza sul display il messaggio STEP IN LINE line-num, che specifica quale istruzione del programma sarà eseguita alla successiva pressione del tasto **STEP**. Dopo di che, ogni volta che si preme il tasto **STEP**, viene eseguita la successiva istruzione. (Si ricordi che il tasto di console **STEP** si illumina ogni volta che il sistema è nello stato di debugging.)



Il tasto di console **TRACE** permette di stampare il numero di linea di ogni istruzione eseguita, nell'ordine con cui essa è eseguita. Analizzando la stampa prodotta attivando la funzione TRACE si può seguire il flusso di esecuzione del programma. (Il tasto **TRACE** è acceso quando la funzione omonima è attiva.) Per utilizzare la funzione TRACE nello stato di debugging:

1. Premere il tasto **TRACE**.

2. Premere il tasto **CONTINUE** oppure introdurre il comando START.

Come conseguenza, si riprende l'esecuzione del programma e vengono stampati i numeri di linea delle istruzioni eseguite. Si noti che non vengono stampati i numeri di linea delle istruzioni non esegutive, come REM o DCL, e l'esecuzione del programma è realizzata come quando la funzione TRACE non è attiva -- per esempio, se una istruzione INPUT è eseguita, l'esecuzione del programma è interrotta e sul display appare un punto interrogativo che indica che il sistema è in attesa di dati da tastiera. (La funzione TRACE può essere attivata anche durante lo stato comandi, premendo **TRACE** e quindi introducendo il comando RUN.)



Quando la funzione del tasto **NO PRINT** è attiva, la stampante integrata è inibita; quindi vengono sopresse tutte le stampe richieste dal programma (mediante le istruzioni PRINT, PRINT USING, MAT PRINT e MAT PRINT USING) che è eseguito. Mentre si verifica un programma può essere utile sopprimere tali stampe -- così facendo si risparmia tempo di stampa e carta. Quando la funzione NO PRINT è attiva, il tasto **NO PRINT** è illuminato.



Il tasto **BREAK**, premuto nello stato di debugging, permette di terminare l'esecuzione di un programma e commuta il sistema nello stato comandi. Nello stato comandi, si può modificare il programma presente in memoria principale oppure introdurre un qualsiasi comando di sistema. Quando la funzione BREAK è attiva, il tasto **BREAK** è illuminato.

Prestazioni dello stato calcoli immediati

Quando il sistema è nello stato di debugging, si possono utilizzare tutte le prestazioni dello stato calcoli immediati -- si possono eseguire delle espressioni immediatamente, si possono usare le variabili  $\Phi$ ,  $\Phi \emptyset$ ,  $\Phi 1$  e  $\Phi 2$  etc. (Si veda il capitolo 6 per una descrizione completa delle prestazioni dello stato calcoli immediati.) Si noti che, tuttavia, ogni operazione tipica dello stato calcoli immediati è eseguita nello stato di debugging senza premere il tasto **CALC MODE** che non è operativo nello stato di debugging.

Analisi ed impiego delle variabili globali di un programma

Durante lo stato di debugging si può analizzare il contenuto di tutte le variabili globali del programma presente in memoria principale. Per determinare il contenuto di una variabile globale, si digiti il suo nome da tastiera (ed eventualmente gli indici, se si tratta di una variabile con indice), quindi si preme il tasto **END OF LINE**. Come risultato, il valore della variabile è visualizzato e stampato (se la funzione PRINT ALL è attiva.)

Per esempio, per visualizzare il valore della variabile A1 si preme: **A 1** **END OF LINE**

Per visualizzare il valore della variabile con indice B(2,5), si preme:

**B ( 2 , 5 )** **END OF LINE**

Si possono assegnare, da tastiera, valori alle variabili globali di un programma, modificando i loro valori correnti, sia direttamente che come risultato dell'esecuzione di una espressione.

Vediamo due esempi:

1. Per assegnare il valore 5 alla variabile globale A, si preme:

**A = 5** **END OF LINE**

2. Per assegnare il risultato dell'espressione

$B * \sqrt{1 + \text{SQR}(C)}$  alla variabile A, si preme:

**A = B \* RESULT 1 + SQR ( C )** **END OF LINE**

Nell'esempio A, B e C sono variabili globali del programma che è in memoria principale.

Nota: Nello stato di debugging non si può assegnare un valore ad una variabile locale. Per la definizione di variabile locale e di variabile globale si veda l'istruzione DEF/FNEND.

Impiego delle funzioni definite nel programma

Durante lo stato di debugging si possono utilizzare non solo le funzioni (numeriche o stringa) di sistema, ma anche le funzioni mono-linea o multi-linea definite nel programma presente in memoria principale.

Vediamo due esempi:

1. Per assegnare il valore della funzione mono-linea FNA, definita nel programma presente in memoria principale, alla variabile A, si preme:

A = FNA ( 8 , 9 ) END OF LINE

2. Per assegnare il prodotto del valore della funzione mono-linea FNA per la funzione multilinea FNC, alla variabile B, si preme:

B = FNA ( 3 , 5 ) \* FNC ( 8 ,  
7 ) END OF LINE

### Un esempio di debugging di un programma

In questo paragrafo diamo un esempio della tecnica impiegata per verificare, correggere ed ampliare un programma. Viene descritto l'impiego delle prestazioni del P6066 e l'interazione che si realizza tra l'utente, il sistema ed il linguaggio BASIC in una tipica sessione di debugging. I numeri sul margine sinistro della stampa ottenuta dal sistema corrispondono alle note che seguono nella descrizione. Le note spiegano cosa è stato fatto e perchè sono utilizzate le prestazioni dello stato di debugging descritte.

Supponiamo che un programmatore voglia realizzare un programma che genera tutti i numeri primi compresi in un certo campo predefinito. Come primo passo, per verificare l'algoritmo da utilizzare, introduce da tastiera un programma che calcola e stampa tutti i numeri primi esistenti tra 1 e 30. L'algoritmo utilizzato è il seguente: si assegnano ad una variabile I, in sequenza, i numeri interi compresi tra 1 e 30; ogni valore della variabile I è diviso, in sequenza, con i valori assunti da una variabile J (che variano da 2, alla radice quadrata di I); se il resto della divisione è zero, allora I non è numero primo, quindi viene assegnato ad I il numero I+1 e ripetuto il medesimo processo. Se il resto della divisione di I per i valori di J compresi tra 2 e la radice quadrata di I è diverso da zero, allora I è un numero primo e viene stampato; ad I è assegnato il numero I+1 e si riprende l'intero processo dall'inizio. Si impiegano i valori di J da 2 alla radice quadrata di I perchè se vi sono dei fattori primi di I questi saranno uno di tali valori.

```

NEW
10REM PROGRAMMA CHE CALCOLA I NUMERI PRIMI
20 PRINT " ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30 "
25PRINT
30 PRINT
40FOR I=2TO30
50K=SOR(I)
60FOR J=2TOK
70E=I/J-INT(I/J)
80IFE=0THEN100
90NEXT J
100PRINT I;
110NEXT I
120PRINT
125 PRINT
130PRINT " FINE DELL'ELENCO "
140END

```

① RUN  
 \*\*\*\* FORMALLY CORRECT PROGRAM \*\*\*\*  
 ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30

② 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23  
 24 25 26 27 28 29 30

FINE DELL'ELENCO  
 ③ LIST  
 FILE

```

0010 REM PROGRAMMA CHE CALCOLA I NUMERI PRIMI
0020 PRINT " ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30 "
0025 PRINT
0030 PRINT
0040 FOR I=2 TO 30 STEP 1
0050 LET K=SOR(I)
0060 FOR J=2 TO K STEP 1
0070 LET E=I/J-INT(I/J)
0080 IF E=0 THEN 100
0090 NEXT J
0100 PRINT I;
0110 NEXT I
0120 PRINT
0125 PRINT
0130 PRINT " FINE DELL'ELENCO "
0140 END

```

END OF LISTING

④ FETCH 40  
 0040 FOR I=2 TO 30 STEP 1  
 0040 FOR I=1 TO 30 STEP 1  
 ⑤ RUN  
 \*\*\*\* FORMALLY CORRECT PROGRAM \*\*\*\*  
 ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22  
 23 24 25 26 27 28 29 30

FINE DELL'ELENCO  
 ⑥ FETCH 70  
 0070 LET E=I/J-INT(I/J)  
 0070 LET E=I/J-INT(I/J)  
 RUN  
 \*\*\*\* FORMALLY CORRECT PROGRAM \*\*\*\*  
 ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30

⑦ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22  
 23 24 25 26 27 28 29 30

FINE DELL'ELENCO

1. Il programmatore ha introdotto da tastiera il suo programma di prova e lo ha fatto eseguire dal sistema introducendo il comando RUN. Come indicato dal messaggio: \*\*\*\* FORMALLY CORRECT PROGRAM \*\*\*\* (il tasto PRINT ALL è attivo), il sistema non ha rilevato alcun errore durante l'analisi di coerenza sintattica tra le istruzioni del programma.
2. Uno sguardo alla stampa prodotta dal programma è sufficiente per vedere che i numeri generati non sono i numeri primi da 1 a 30, ma l'elenco completo dei numeri interi da 2 a 30.
3. Il programmatore introduce il comando LIST per ottenere un listing del suo programma più facile da leggere. Nota un primo errore nella istruzione con numero di linea 40; la variabile I assume come valore iniziale 2 invece di 1.
4. Introduce il comando FETCH per trasferire nel buffer di tastiera l'istruzione suddetta che viene corretta.
5. Riesegue il programma. Questa volta il numero 1 è compreso nell'elenco dei numeri stampati, ma sono ancora presenti anche i numeri non primi. Una successiva analisi del listing del programma gli permette di notare un errore nella istruzione 70: ha digitato un 1 invece di I.
6. Introduce di nuovo un comando FETCH per correggere l'istruzione suddetta.
7. Una successiva esecuzione del programma produce gli stessi errori.
8. A questo punto il programmatore decide di utilizzare gli strumenti dello stato di debugging, per cui introduce il comando PREPARE ed il sistema passa nello stato di debugging.
9. Preme il tasto di console **STEP** (come indicato dal messaggio STEP IN LINE). Il programmatore continua a premere il tasto **STEP**, facendo eseguire ogni volta una istruzione. Ad un certo punto, egli osserva che l'istruzione 100 -- che dovrebbe stampare solamente i numeri primi -- viene eseguita ogni volta.

10. Per verificare il valore corrente della variabile I (il valore stampato dall'istruzione 100), digita I da tastiera e preme **END OF LINE**. Sul display appare il numero 4 -- che non è un numero primo. Ora il programmatore può restringere la causa dell'errore a tre possibilità:

- il calcolo nell'istruzione 70 è errato
- il test nell'istruzione 80 è errato
- il riferimento all'istruzione a cui saltare (contenuto nell'istruzione 80) è errato

Rianalizzando il programma da questi punti di vista si avvede che è il terzo punto la causa di errore: se  $E=0$  (cioè, se la divisione non fornisce alcun resto), l'esecuzione del programma dovrebbe proseguire dall'istruzione 110 per incrementare di 1 la variabile I. Il programmatore preme il tasto **BREAK** per commutare il sistema nello stato comandi e modificare il programma.

11. Introduce il comando **FETCH** per trasferire l'istruzione 80 nel buffer di tastiera. L'istruzione viene modificata.

12. Per assicurarsi che ora l'esecuzione del ciclo **FOR/NEXT** è corretta, il programmatore preme il tasto di console **TRACE** e quindi introduce il comando **RUN**.

13. Visto che la quarta volta in cui è eseguito il ciclo **FOR/NEXT**, l'istruzione 100 non è eseguita (è eseguita l'istruzione 110), preme di nuovo il tasto **TRACE** per inibirne la funzione. Il programma è così eseguito per intero, senza stampare i numeri di linea delle successive istruzioni eseguite.

14. Analizzando la stampa prodotta dal programma osserva che i numeri generati sono effettivamente tutti i numeri primi compresi tra 1 e 30.

15. Per avere una copia del programma corretto introduce di nuovo il comando **LIST**. Ora che i risultati prodotti dalla routine sono corretti, modifica il programma per ottenere una soluzione più generale del problema: genera un programma che produce tutti i numeri primi compresi tra due numeri interi N ed

M. Aggiunge le due istruzioni seguenti:

15 DISP " INTRODUCI I LIMITI N,M "  
16 INPUT N,M

e modifica le istruzioni 20 e 40.

```
⑧ PREPARE
  01 ROOM=28200 01
  STEP      IN LINE 20
  ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30
⑨ STEP      IN LINE 25

  STEP      IN LINE 30

  STEP      IN LINE 40
  STEP      IN LINE 50
  STEP      IN LINE 60
  STEP      IN LINE 100
  STEP      IN LINE 110
  STEP      IN LINE 50
  STEP      IN LINE 60
  STEP      IN LINE 100
  STEP      IN LINE 110
  STEP      IN LINE 50
  STEP      IN LINE 60
  STEP      IN LINE 100
  STEP      IN LINE 110
  STEP      IN LINE 50
  STEP      IN LINE 60
  STEP      IN LINE 70
  STEP      IN LINE 80
  STEP      IN LINE 100
  STEP      IN LINE 110
⑩ I
  4
  1 2 3 4
⑪ FETCH 80
  0080 IF E=0 THEN 100
  0080 IF E=0 THEN 110
⑫ RUN
  **** FORMALLY CORRECT PROGRAM ****
  ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30

  #30

  #40
  #50
  #60
  #100
  #110
  #50
  #60
  #100
  #110
```

```

#50
#60
#100
#110
#50
#60
#70
#80
#110
#50
#60
#70
#80
#90
#100
#110
13 #50
14 1 2 3 5 7 11 13 17 19 23 29
FINE DELL'ELENCO

```

16. Il programmatore introduce il comando RESEQUENCE prima di richiedere un listing finale del programma.

17. Stampa il nuovo listing. Infine registra nella libreria MAT il programma con il nome NUMPR, eseguendo il comando SAVE NUMPR,MAT.

15 LIST  
FILE

```

0010 REM PROGRAMMA CHE CALCOLA I NUMERI PRIMI
0020 PRINT " ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30 "
0025 PRINT
0030 PRINT
0040 FOR I=1 TO 30 STEP 1
0050 LET K=SQR(I)
0060 FOR J=2 TO K STEP 1
0070 LET E=I/J-INT(I/J)
0080 IF E=0 THEN 110
0090 NEXT J
0100 PRINT I;
0110 NEXT I
0120 PRINT
0125 PRINT
0130 PRINT " FINE DELL'ELENCO "
0140 END

```

END OF LISTING

```

15 DISP " INTRODUCI I LIMITI N,M "
16 INPUT N,M
FETCH 20
0020 PRINT " ECCO TUTTI I NUMERI PRIMI COMPRESI TRA 1 E 30 "
0020 PRINT " ECCO TUTTI I NUMERI PRIMI COMPRESI TRA N ED M "
FETCH 40
0040 FOR I=1 TO 30 STEP 1

```

```

0040 FOR I=N TO M STEP 1
(16) RESEQUENCE
(17) LIS
FILE
0010 REM PROGRAMMA CHE CALCOLA I NUMERI PRIMI
0020 DISP " INTRODUCI I LIMITI N,M "
0030 INPUT N,M
0040 PRINT " ECCO TUTTI I NUMERI PRIMI COMPRESI TRA N ED M "
0050 PRINT
0060 PRINT
0070 FOR I=N TO M STEP 1
0080 LET K=SQR(I)
0090 FOR J=2 TO K STEP 1
0100 LET E=I/J-INT(I/J)
0110 IF E=0 THEN 140
0120 NEXT J
0130 PRINT I;
0140 NEXT I
0150 PRINT
0160 PRINT
0170 PRINT " FINE DELL'ELENCO "
0180 END

```

END OF LISTING

SAVE NUMPR, MAT



## 8. IMPIEGO DEI TASTI FUNZIONE

Agli otto tasti funzione della sezione funzioni definibili, vedi figura 8-1, si possono assegnare, utilizzando anche il tasto **SHIFT**, fino a 16 funzioni definite dall'utente.

Questo capitolo riprende in modo organico i diversi aspetti sull'impiego dei tasti funzione, già descritti nei capitoli precedenti (vedi capitolo 3 comando LDKEYS, capitolo 5 istruzione FKEY# e capitolo 6), per permettere al lettore di ritrovare facilmente le informazioni che gli permettono di sfruttare in modo completo questa prestazione importante del sistema P6066.

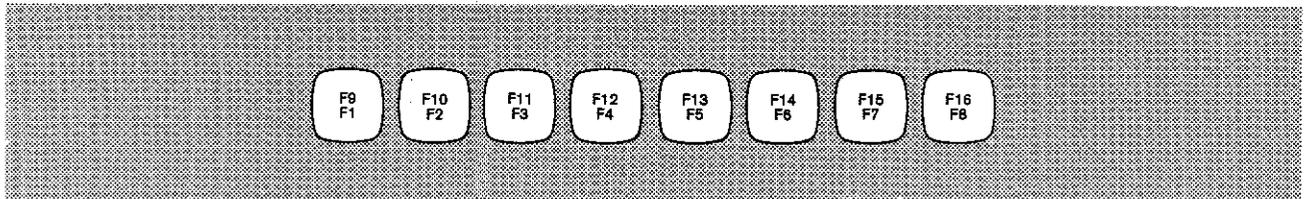


Figura 8-1 Tasti funzione

### Assegnazione di una funzione ad un tasto funzione

L'assegnazione di una funzione ad un tasto funzione può essere effettuata durante tre diversi stati del sistema, ma la struttura dell'assegnazione è sempre la seguente:

FKEY# n, stringa

dove

n

è un numero intero, da 1 a 16, che indica a quale tasto funzione (F1 + F16) deve essere assegnata la stringa specificata

stringa

indica quale espressione numerica o stringa di caratteri deve essere assegnata al tasto funzione specificato.

L'impiego della assegnazione di una funzione ad un tasto funzione varia in relazione allo stato del sistema. Gli stati del sistema in cui si può assegnare ai tasti funzione una funzione specifica sono:

- lo stato d'esecuzione programma
- lo stato calcoli immediati e lo stato di debugging
- lo stato comandi — mediante il comando LDKEYS

Ad ogni tasto funzione si può assegnare una stringa di caratteri, od una espressione numerica, con non più di 73 caratteri. Le stringhe di caratteri, o le espressioni numeriche, assegnate ai tasti funzione, sono registrati in un registro (buffer) di 238 caratteri (byte); quindi il numero totale di caratteri assegnati a tutti i tasti funzione non può essere maggiore di 238. Si noti che viene usato sempre lo stesso registro per registrare le stringhe di caratteri, o espressioni numeriche, assegnate ai tasti funzione, qualunque sia lo stato del sistema in cui tale assegnazione è effettuata. Per cui, le stringhe di caratteri, o espressioni numeriche, assegnate ai tasti funzione durante lo stato calcoli immediati, saranno cancellate quando verrà eseguito un programma in cui sono presenti delle istruzioni di assegnazione di funzione ai tasti funzione.

Nota: Per ricordare l'assegnazione fatta ai tasti funzione, si può utilizzare una striscia di carta con 16 caselle nelle quali si possono scrivere dei codici mnemonici relativi ai tasti funzione sottostanti (vedi figura 1-8).

Assegnazione di una  
funzione ai tasti fun-  
zione durante l'esecu-  
zione di un programma

I tasti funzione sono utilizzati, in genere, in programmi costituiti da diverse routine affinché l'operatore possa scegliere la routine da far eseguire. (Naturalmente si può utilizzare un programma per assegnare ai tasti funzione delle funzioni che saranno poi utilizzate nello stato calcoli immediati o nello stato di debugging.) Per assegnare da programma una stringa di caratteri o una espressione numerica ai tasti funzione si deve utilizzare l'istruzione FKEY#(vedi capitolo 5).

Quando l'istruzione FKEY# è eseguita, la stringa di caratteri a destra del segno di virgola è assegnata al tasto funzione specificato (quindi è memorizzata nel relativo registro). Dopo di che, quando il tasto funzione è premuto la stringa di caratteri ad esso assegnata è trasferita nel buffer di tastiera.

Diamo di seguito due esempi di impiego dei tasti funzione a cui è assegnata una funzione predefinita da programma.

Esempio 1: Vediamo come si crea un programma che permette di usare i tasti funzione in modo da poter scegliere quale routine nell'ambito del programma deve essere eseguita. Non diamo il programma completo ma solo la parte che evidenzia l'impiego delle istruzioni FKEY#. L'impiego dei tasti funzione permetterà di digitare valori di lunghezza, area e volume espressi con qualsiasi unità di misura, poichè opportune routine li convertiranno nelle unità di misura accettate dall'algoritmo "principale" del programma.

```

0010 FKEY #1, START 2000:
0020 FKEY #2, START 2110:
0030 FKEY #3, START 2210:
0040 FKEY #4, START 2310:
0050 FKEY #5, START 2410:
0060 FKEY #6, START 2510:
0070 FKEY #7, START 100:
0080 DISP "Scegli la routine ";
0090 STOP
0100 REM DATI IN METRI, METRI QUADRI, METRI CUBI
0110 INPUT L,A,U
0120
... .. }
... .. }          ALGORITMO PRINCIPALE
... .. }

2000 REM DATI IN CENTIMETRI, CENTIMETRI QUADRI, CENTIMETRI CUBI
2010 INPUT L,A,U

... .. }
... .. }          ROUTINE DI CONVERSIONE
... .. }

2100 GOTO 120
2110 REM DATI IN DECIMETRI, DECIMETRI QUADRI, DECIMETRI CUBI
2120 INPUT L,A,U

... .. }
... .. }          ROUTINE DI CONVERSIONE
... .. }

```

```

2200 GOTO 120
2210 REM DATI IN MILLIMETRI, MILLIMETRI QUADRI, MILLIMETRI CUBI
2220 INPUT L,A,V

... .. }
... .. }          ROUTINE DI CONVERSIONE
... .. }

2300 GOTO 120
2310 REM DATI IN MILLIMETRI, CENTIMETRI QUADRI, CENTIMETRI CUBI
2320 INPUT L,A,V

... .. }
... .. }          ROUTINE DI CONVERSIONE
... .. }

2400 GOTO 120
2410 REM DATI IN MILLIMETRI, CENTIMETRI QUADRI, MILLIMETRI CUBI
2420 INPUT L,A,V

... .. }
... .. }          ROUTINE DI CONVERSIONE
... .. }

2500 GOTO 120
2510 REM DATI IN CENTIMETRI, MILLIMETRI QUADRI, MILLIMETRI CUBI
2520 INPUT L,A,V

... .. }
... .. }          ROUTINE DI CONVERSIONE
... .. }

2600 GOTO 120
2610 END

```

Le prime istruzioni del programma sono delle assegnazioni ai tasti funzione del comando START che permettono di riprendere l'esecuzione del programma da istruzioni prefissate. Dopo di che segue una istruzione STOP che interrompe l'esecuzione del programma. L'utente potrà così, premendo il tasto funzione adeguato (da F1 a F7), lanciare l'esecuzione della routine di input che traduce i valori di lunghezza, area e volume (introdotti con una certa unità di misura) in metri, metri quadri e metri cubi. Dopo la routine di conversione è eseguito l'algoritmo principale del programma.

Nota: I due punti ":" alla fine di ogni assegnazione (istruzione 10 + 70) sono interpretati dal sistema come un END OF LINE per cui il comando START è introdotto nel sistema premendo solo il tasto funzione relativo senza dover premere successivamente END OF LINE .

Esempio 2: Se si vuole usare il sistema per convertire le misure inglesi nel sistema decimale e viceversa, si può scrivere un programma come il seguente:

```

0010 FKEY#1,*0.3048:
0020 FKEY#9,/0.3048:
0030 FKEY#2,*25.4:
0040 FKEY#10,/25.4:
0050 FKEY#3,*1.6099344:
0060 FKEY#11,/1.6099344:
0070 FKEY#4,*0.092903:
0080 FKEY#12,/0.092903:
0090 FKEY#5,*6.4516:
0100 FKEY#13,/6.4516:
0110 FKEY#6,*28.3168:
0120 FKEY#14,/28.3168:
0130 FKEY#7,*4.54609:
0140 FKEY#15,/4.54609:
0150 FKEY#8,*0.45360:
0160 FKEY#16,/0.45360:
0170 END

```

Quando il programma è eseguito le espressioni di conversione sono assegnate ai tasti funzione. Si può quindi utilizzare il sistema nello stato esecuzione di calcoli immediati per convertire il valore introdotto da tastiera premendo successivamente il rispettivo tasto funzione. Si avrà che:

Premendo	il tasto	si convertono	in
"	F 1	piedi	metri
"	F 2	pollici	mm
"	F 3	miglia	Kn
"	F 4	piedi quadri	m <sup>2</sup>
"	F 5	pollici quadri	cm <sup>2</sup>
"	F 6	piedi cubi	litri
"	F 7	galloni	litri
"	F 8	libbre	Kg
"	F 9	metri	piedi
"	F10	millimetri	pollici
"	F11	chilometri	miglia
"	F12	metri quadri	piedi quadri
"	F13	centimetri quadri	pollici quadri
"	F14	litri	piedi cubi
"	F15	litri	galloni
"	F16	chilogrammi	libbre

Assegnazione di una  
funzione ai tasti fun-  
zione durante lo stato  
calcoli immediati

Quando il sistema è nello stato calcoli immediati si possono assegnare ai tasti funzione delle espressioni o costanti utilizzate comunemente. L'assegnazione è effettuata come già descritto nel capitolo 6.

Esempio: Si usi il sistema P6066 per calcolare il valore medio e la deviazione standard di una serie di numeri introdotti da tastiera. Si fanno le seguenti assegnazioni ai tasti funzione:

```
FKEY#1.  $\Sigma$ =  
FKEY#2.  $\Sigma 1 = \Sigma 1 + \Sigma$ :  
FKEY#3.  $\Sigma 2 = \Sigma 2 + \Sigma 1^2$ :  
FKEY#4.  $\Sigma 0 = \Sigma 0 + 1$ :  
FKEY#5.  $\Sigma 1 / \Sigma 0$ :  
FKEY#6.  $\text{SQR}((\Sigma 0 * \Sigma 2 - \Sigma 1^2) / \Sigma 0 / (\Sigma 0 - 1))$ :  
FKEY#10.  $\Sigma 1 = 0$ :  
FKEY#11.  $\Sigma 2 = 0$ :  
FKEY#12.  $\Sigma 0 = 0$ :
```

dove F 2 calcola la sommatoria dei valori X introdotti da tastiera

" F 3 calcola la sommatoria di  $X^2$

" F 4 registra quanti numeri (Y) compongono la media

" F 5 calcola la media

" F 6 calcola la deviazione standard dalla media

" F10 azzera  $\Sigma X$

" F11 azzera  $\Sigma X^2$

" F12 azzera Y (il numero di dati che compongono la media)

La procedura è la seguente:

1 - Premere	F10	} si azzerano gli accumulatori
2 - Premere	F11	
3 - Premere	F12	
4 - Premere	F1	
5 - Introdurre	X	
6 - Premere	END OF LINE	si assegna X a $\Phi 0$

- 7 - Premere F2
- 8 - Premere F3
- 9 - Premere F4
- 10 - Si ripeta la procedura dal punto 4 per tutti i valori di X
- 11 - Per avere la media si preme F5
- 12 - Per avere la deviazione standard si preme F6
- 13 - Per ripetere i calcoli su un altro insieme di valori X si riparte dal numero 1.

Impiego dei tasti funzione nello stato comandi

Durante lo stato comandi si possono registrare sul disco sistema o sul floppy disk sistema le stringhe di caratteri od espressioni numeriche che sono state assegnate ai tasti funzione. In questo modo si possono ripristinare ai tasti funzione le funzioni desiderate. Se l'assegnazione di una funzione ad un tasto funzione è effettuata da programma, non è necessaria la suddetta operazione perchè ogni volta che il programma è eseguito ai tasti funzione vengono riassegnate le funzioni definite nel programma. Se però l'assegnazione di un contenuto ai tasti funzione è effettuata durante lo stato calcoli immediati, per poterla ripristinare, si ricorre al seguente comando di sistema:

STKEYS

che registra su disco sistema o sul floppy disk sistema il contenuto attuale dei tasti funzione. Quando il sistema è successivamente inizializzato, il buffer relativo ai tasti funzione viene caricato dal disco sistema o dal floppy disk sistema con il contenuto ivi registrato mediante il comando STKEYS. Se tale contenuto associato ai tasti funzione è modificato, sia da un programma che da nuove assegnazioni ai tasti funzione effettuate durante lo stato calcoli immediati, esso può essere ripristinato mediante il comando LDKEYS (vedi capitolo 3). I comandi suddetti sono utili quando si ha una notevole mole di gestione delle librerie. In questo caso si assegnano ai tasti funzione i comandi che richiamano in memoria principale ed eseguono i programmi di utilità che gestiscono le librerie. Quindi si registrano le assegnazioni suddette sul disco sistema o sul floppy disk sistema mediante il comando STKEYS. Quando le assegnazioni registrate sul disco sistema sono necessarie, si possono richiamare in memoria principale mediante il comando LDKEYS. Dopo di che non è necessario digitare ogni volta i comandi, richiesti, perchè basta premere l'appropriato tasto

funzione per richiamare in memoria principale il programma di utilità.

## 9. IMPIEGO DI UN VIDEO DISPLAY

Il P6066 può essere collegato ad un video display Olivetti DSM 6680 (vedi figura 9-1) utilizzando due diversi governi: DSC 6681 e DSC 6683. Il governo DSC 6681 permette solamente la visualizzazione di linee di caratteri alfanumerici, mentre il governo DSC 6683 permette inoltre di tracciare dei grafici e delle immagini comunque complesse.

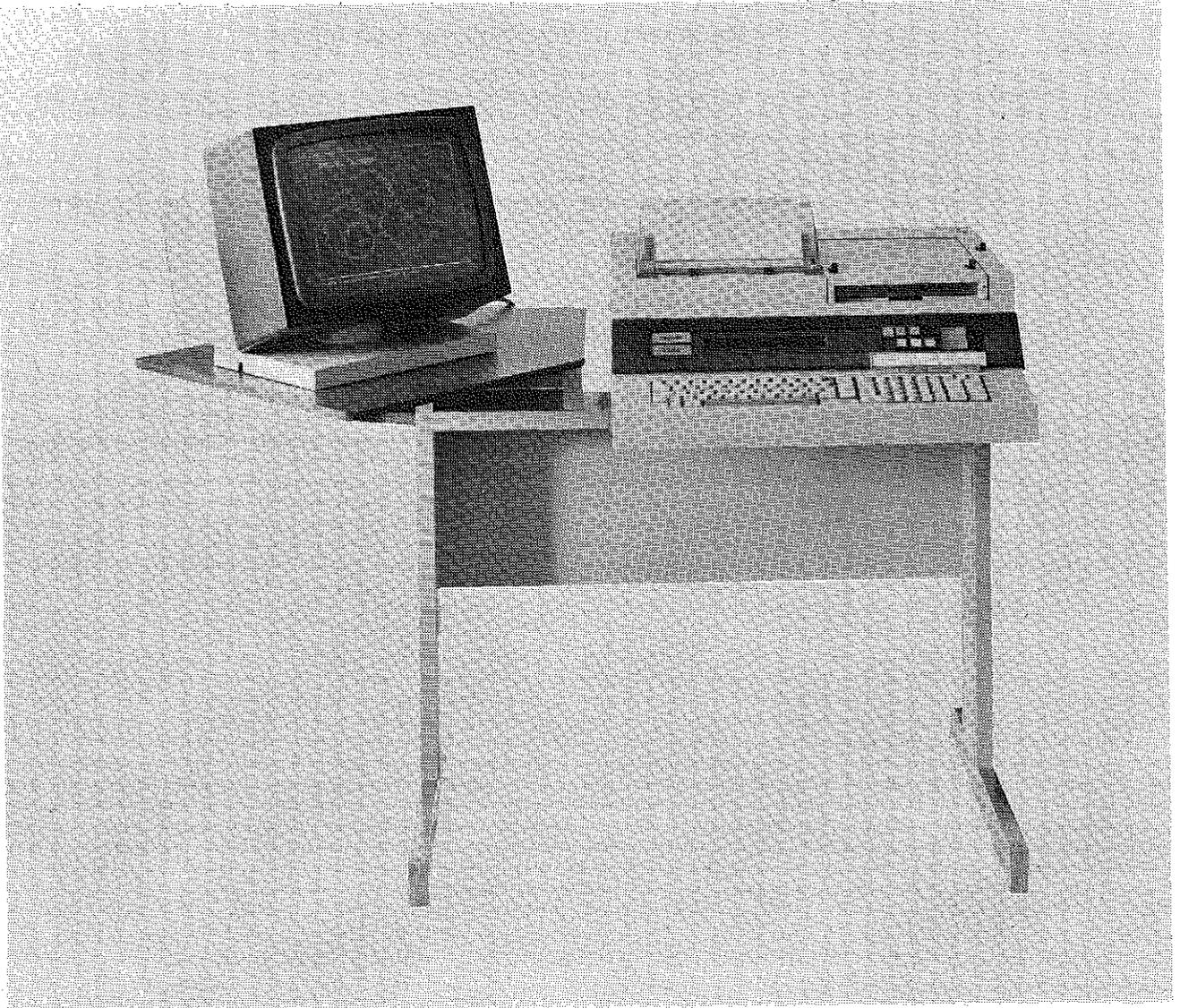


Figura 9-1 P6066 con video display Olivetti DSM 6680

In questo capitolo è descritto l'impiego del video display suddetto con il governo DSC 6681, per quanto riguarda il tracciamento di grafici si veda il manuale P6066 Prestazioni grafiche - Manuale del programmatore GP Code 3978350 Q (0).

Attivazione del video display

Per utilizzare il video display deve essere stato eseguito un comando CONFIGURE con l'operando EXD.

Informazioni visualizzate sullo schermo

Sullo schermo si possono visualizzare fino a 41 linee contemporaneamente, con un massimo di 80 caratteri per linea. Sullo schermo possono apparire:

- messaggi di sistema e messaggi di errore
- caratteri introdotti da tastiera
- testi e dati specificati in istruzioni DISP e DISP USING
- testi e dati specificati in istruzioni PRINT, PRINT USING, MAT PRINT e MAT PRINT USING
- informazioni fornite dai comandi CATALOG, ENVIRONMENT, LIST, LVTOC e dal programma di utilità FLPRINT

Per facilitare il riferimento al contenuto dello schermo supponiamo che le linee siano numerate da 0 a 40, come nella figura 9-2.

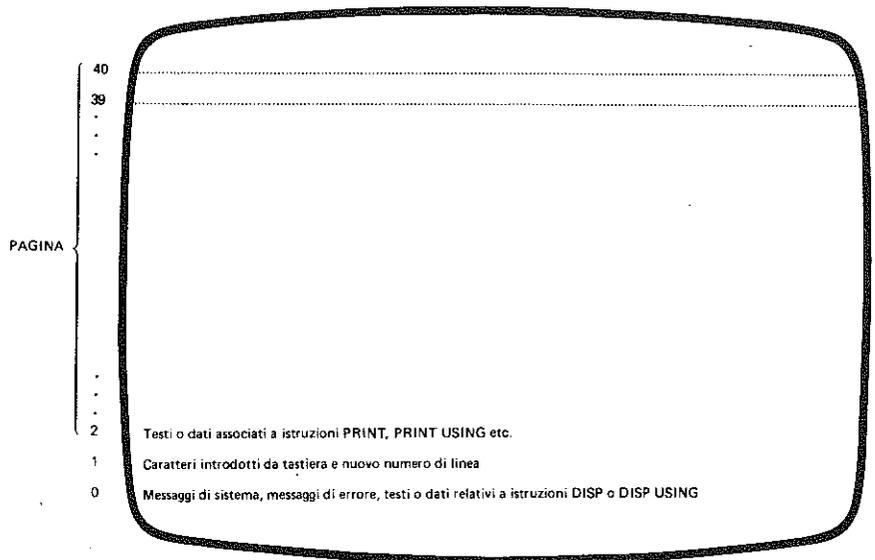


Figura 9-2 Informazioni visualizzate sul video display

Sulla linea 0 sono visualizzati tutti i messaggi di sistema, i messaggi di errore ed i testi o dati specificati nelle istruzioni DISP e DISP USING. Se **PRINT ALL** è attivo, il contenuto della linea 0 è contemporaneamente visualizzato sulla linea 2 ed i precedenti contenuti delle linee da 2 a 39 sono spostati di una linea verso l'alto.

Sulla linea 1 sono visualizzati i caratteri introdotti da tastiera durante la generazione o l'editing di un programma o file testo, oppure in risposta ad un'istruzione di INPUT. Nella linea è visualizzato anche un cursore che indica la posizione in cui sarà inserito il successivo carattere digitato. Se la linea digitata non è corretta, quando è premuto **END OF LINE** essa rimane nella posizione precedente per essere corretta. Nel caso in cui sia corretta, scompare dallo schermo dopo che si è premuto **END OF LINE**, od è trasferita sulla linea 2 se **PRINT ALL** è attivo ed i precedenti contenuti delle linee da 2 a 39 sono spostati di una linea verso l'alto.

Si noti infine che, se è stata attivata la numerazione automatica, ogni volta che una linea corretta è accettata dal sistema il numero di linea successivo è vi-

sualizzato all'inizio della linea 1.

Sulla linea 2 sono visualizzati i testi ed i dati specificati nelle istruzioni PRINT, PRINT USING, MAT PRINT e MAT PRINT USING. In questo caso i contenuti precedenti delle linee da 2 a 39 sono spostati verso l'alto di una linea.

Sulle linee da 2 a 40 sono visualizzate le informazioni fornite dai comandi CATALOG, ENVIRONMENT, LIST, LVOTC e dal programma di utilità FLPRINT. Inoltre possono essere visualizzate stringhe di caratteri specificate in un'istruzione DISP con un formato particolare che descriveremo più avanti.

#### Impaginazione

Le linee da 2 a 40 compongono una "pagina" di 39 linee che possono essere visualizzate contemporaneamente. Se le informazioni dirette allo schermo, da un programma od un comando, sono costituite da più di 39 linee, alcune linee saranno cancellate e non si potranno mai leggere. Per evitare questo inconveniente si deve utilizzare il tasto **NO PRINT**. Infatti, attivando la funzione NO PRINT, tasto **NO PRINT** illuminato, oltre ad ottenere un risparmio di carta sulla stampante integrata, in quanto se ne inibisce il funzionamento, s'interrompe l'esecuzione del programma o del comando di sistema dopo che sullo schermo sono apparse le 39 linee della prima pagina. Il tasto **CONTINUE** s'illumina per avvertire l'utente di questa situazione e permettergli di leggere le informazioni visualizzate. Premendo il tasto **CONTINUE** l'esecuzione del programma o del comando riprende dal punto interrotto e continua fino alla fine o finchè una nuova pagina di 39 linee non sia visualizzata sullo schermo. Quando questo avviene l'esecuzione s'interrompe nuovamente ed il tasto **CONTINUE** s'illumina di nuovo. L'utente può così leggere tutte le pagine che sono visualizzate fino alla fine, premendo successivamente il tasto **CONTINUE**.

#### Istruzioni BASIC e comandi per il video display

Per l'impiego del video display, oltre alle istruzioni BASIC ed ai comandi menzionati precedentemente, si possono utilizzare una nuova forma di istruzione DISP ed i comandi ERASE e REVERSE che sono descritti nelle pagine seguenti.



Istruzione DISP

Funzione

Permette la visualizzazione di una stringa di caratteri in una linea qualsiasi del video display.

Formato

**DISP "control-char op-code [line-position char-position] [char-string]"**

dove:

control-char

è il carattere  $\theta$

op-code

*W, Z, Y, X*

è uno dei seguenti caratteri: ~~W, Z, Y, X~~, B, A, S

line-position

è un carattere della tabella ISO usato per rappresentare la posizione di una linea sul video

char-position

è un carattere della tabella ISO usato per rappresentare la posizione di un carattere in una linea sul video

char-string

è una stringa di caratteri che deve essere visualizzata sul video

Azione

*attuale di char-string*

Quando l'istruzione è eseguita con ~~X~~<sup>W</sup> come op-code, <sup>i caratteri</sup> la <sup>carattere</sup> linea nella posizione line-position è completamente <sup>carattere</sup> sostituita con una linea contenente la stringa char-string nella posizione specificata con char-position. <sup>carattere</sup> La stringa è preceduta e seguita da eventuali spazi.

Quando l'istruzione è eseguita con ~~Z~~<sup>Z</sup> come op-code, l'istruzione funziona come nel caso precedente, ma la linea è visualizzata invertendo il suo colore con

*con X = W, ma il carb. precedente non viene cancellato*  
*con X = Z, idem.*

quello dello sfondo dello schermo e viceversa.

Quando l'istruzione è eseguita con B come op-code, sono soppresse le visualizzazioni del contenuto delle istruzioni PRINT che seguono nel programma. Se è specificato l'operando char-string la stringa corrispondente è visualizzata nella linea 0 (vedi la figura 9-2).

Quando l'istruzione è eseguita con A come op-code, sono riattivate le visualizzazioni del contenuto delle istruzioni PRINT che seguono nel programma. Se è specificato l'operando char-string la stringa corrispondente è visualizzata nella linea 0 (vedi figura 9-2).

Quando l'istruzione è eseguita con S come op-code, sposta di una linea verso l'alto le linee da 2 a 39 con perdita della linea 40 (vedi figura 9-2). Se è specificato l'operando char-string la stringa corrispondente è visualizzata nella linea 0.

#### Note

1. Per introdurre il carattere di controllo  $\theta$  si devono premere contemporaneamente i tasti **CONTROL** e **I**.
2. La posizione della linea è specificata mediante un carattere della tabella ISO il cui valore decimale corrispondente varia da 32 a 70 (vedi l'Appendice E). La linea dello schermo in cui sarà visualizzata la stringa di caratteri è quella il cui numero è uguale al valore decimale corrispondente al carattere specificato meno 30. Quindi le stringhe di caratteri possono essere visualizzate su una linea qualunque da 2 a 40 come mostrato in figura 9-2.
3. La posizione da cui è visualizzata una stringa di caratteri è specificata mediante un carattere della tabella ISO il cui valore decimale corrispondente varia da 32 a 111 (vedi l'Appendice E). La posizione nella linea da cui inizierà la stringa di caratteri è la posizione data dal valore decimale corrispondente al carattere specificato meno 31. Quindi in ogni linea si possono visualizzare fino a 80 caratteri.
4. La posizione di una linea e la posizione d'inizio di una stringa di caratteri nell'ambito di una linea possono essere specificate usando la funzione

di sistema CHR\$. Se si utilizza CHR\$ l'istruzione DISP avrà il seguente formato:

DISP "control-char op -code" =CHR\$(30+n) +CHR\$(31+m) + "char-string"

dove n è un intero da 2 a 40 ed m è un intero da 1 a 80. Il valore specificato per n corrisponde ad una linea del video (vedi figura 9-2); il valore specificato per m è la posizione nella linea in cui inizierà la stringa di caratteri. Per ulteriori informazioni sulla funzione CHR\$ si veda il capitolo 4.

5. Se è specificata una stringa di caratteri che supera la 80esima posizione di una linea, la stringa è troncata alla 80esima posizione.
6. La stringa di caratteri può essere specificata come variabile stringa (vedi l'esempio 3).

#### Esempi

1. La seguente istruzione DISP visualizza la stringa di caratteri:

Calcolo matriciale

nella linea 10 iniziando dalla posizione 9

```
100 DISP "0K(( Calcolo matriciale"
```

2. In questo caso viene usata la funzione di sistema CHR\$ per visualizzare la stringa di caratteri:

Calcolo matriciale

nella linea 10 iniziando dalla posizione 9

```
100 DISP "0K"+CHR$(30+10)+CHR$(31+9)+"Calcolo matriciale"
```

3. Nel seguente esempio il valore di A\$ (che è la stringa di caratteri Calcolo matriciale) è visualizzato nella linea 10 iniziando dalla posizione 9.

```
100 DISP "0K(" +A$
```







Comando REVERSE

Funzione

Converte il colore dei caratteri visualizzati nel colore dello sfondo dello schermo e viceversa.

Formato

**REV[ERSE]**

Azione

REVERSE converte il colore dei caratteri visualizzati nel colore dello sfondo dello schermo e viceversa.

