

Capitolo 20

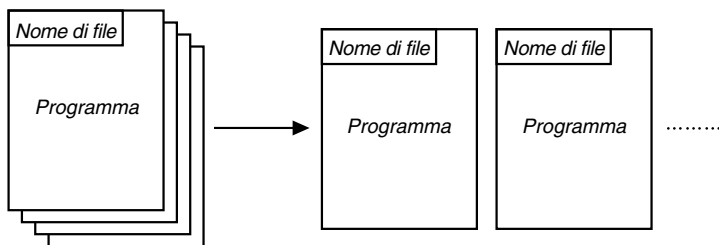


Programmazione

- 20-1 Prima della programmazione
- 20-2 Esempi di programmazione
- 20-3 Messa a punto di un programma
- 20-4 Calcolo del numero di byte utilizzati da un programma
- 20-5 Funzione di segreto
- 20-6 Ricerca di un file
- 20-7 Ricerca di dati all'interno di un programma
- 20-8 Modifica dei nomi di file e del contenuto di un programma
- 20-9 Cancellazione di un programma
- 20-10 Utili comandi per i programmi
- 20-11 Riferimento per i comandi
- 20-12 Visualizzazione del testo
- 20-13 Uso delle funzioni della calcolatrice nei programmi

20-1 Prima della programmazione

La funzione di programmazione serve per rendere facili e rapidi calcoli complessi e spesso ripetuti. I comandi e i calcoli vengono eseguiti sequenzialmente, proprio come le multiistruzioni per i calcoli manuali. È possibile memorizzare più programmi sotto dei nomi di file per poterli richiamare e modificare facilmente in seguito.



Scegliere l'icona **PRGM** nel menu principale ed entrare nel modo PRGM. Quando si esegue questa operazione, una lista di programmi appare sul display.

Area di memoria scelta —
(usare ▲ e ▼ per spostarsi)

Program List	
UC1A	: 37
TRIANGLE	: 17
AREA *	: 33
GRAPHICS	: 17
MEASURE	: 17
OCTONARY	: 17
[EXE][EDIT][NEW][DEL][DEL] ▸	

- **{EXE}/{EDIT}** ... {esecuzione}/{modifica} di un programma
- **{NEW}** ... {nuovo programma}
- **{DEL}/{DEL-A}** ... Cancellazione di {un programma specifico}/{tutti i programmi}
- **{SRC}/{REN}** ... {ricerca}/{cambiamento} del nome di un file
- **{LOAD}** ... {Carica un programma della biblioteca di programmi incorporata.}
* Per i dettagli, fare riferimento al manuale della biblioteca di programmi **a parte**.

- Se nella memoria non è memorizzato alcun programma quando si entra nel modo PRGM, il messaggio **"No Programs"** appare sul display e solo la voce NEW (**F3**) viene visualizzata nel menu delle funzioni.

I valori alla destra della lista dei programmi indicano il numero di byte utilizzati da ciascun programma.



Pag. 368

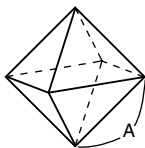
Pag. 362



20-2 Esempi di programmazione

Esempio 1 Per calcolare l'area e il volume di tre ottaedri regolari delle dimensioni indicate nella tabella sottostante.

Memorizzare la formula di calcolo sotto il nome di file OCTA.



Lunghezza di un lato (A)	Area (S)	Volume (V)
7 cm	cm ²	cm ³
10 cm	cm ²	cm ³
15 cm	cm ²	cm ³

Le seguenti sono le formule usate per il calcolo dell'area (S) e del volume (V) di un ottaedro regolare di cui è nota la lunghezza di un lato.

$$S = 2\sqrt{3}A^2, \quad V = \frac{\sqrt{2}}{3}A^3$$

Quando si introduce una nuova formula, innanzitutto registrare il nome di file e quindi introdurre il programma vero e proprio.

●Per registrare un nome di file

Esempio Per registrare il nome di file OCTA

- Notare che un nome di file può essere di una lunghezza massima di otto caratteri.

1. Visualizzare il menu della lista dei programmi e premere **F3** (NEW) per visualizzare un menu, che contiene le seguenti voci.

- **{RUN}/{BASE}** ... Introduzione di programmi per {calcoli generici}/ {specificazioni di base numerica}
- **{n0}** ... {registrazione della parola d'ordine}
- **{SYBL}** ... {menu dei simboli}

2. Introdurre il nome del file.

[O] [C] [T] [A]

Program Name
[OCTA0]

- Il cursore cambia forma a indicare l'introduzione di caratteri alfabetici.
- I seguenti sono i caratteri che è possibile usare in un nome di file:
Da A a Z, r, θ, spazi, [,], {, }, ', ", ~, da 0 a 9, ., +, -, ×, ÷
- Notare, tuttavia, che **[X.θ]** e **[]** non possono essere introdotti per il nome di un programma che contiene calcoli binari, ottali, decimali o esadecimali.



Pag. 360

- Usare **[F1]** (RUN) per introdurre un programma per calcoli generici (un programma da eseguire nel modo COMP). Per programmi che implicano specificazioni di sistema di numerazione, usare **[F2]** (BASE). Notare che i programmi introdotti dopo la pressione di **[F2]** (BASE) sono indicati da **[B]** alla destra del nome di file.
 - La pressione di **[F6]** (SYBL) visualizza un menu di simboli (' , " , ~) che possono essere introdotti.
 - È possibile cancellare un carattere durante l'introduzione di un nome di file spostando il cursore sul carattere che si desidera cancellare e premendo **[DEL]**.
3. Premere **[EXE]** per registrare il nome di file e passare allo schermo per l'introduzione del programma.



- La registrazione di un nome di file utilizza 17 byte di memoria.
- Lo schermo per l'introduzione del nome di file rimane visualizzato sul display se si preme **[EXE]** senza introdurre un nome di file.
- Per abbandonare lo schermo per l'introduzione del nome di file e ritornare alla lista dei programmi senza registrare un nome di file, premere **[EXIT]**.
- Quando si registra il nome di un programma che contiene calcoli binari, ottali, decimali o esadecimali, l'indicatore **[B]** viene aggiunto alla destra del nome di file.

●Per introdurre un programma

Le seguenti voci sono incluse nel menu delle funzioni dello schermo per l'introduzione del programma, che serve per l'introduzione dei programmi.

- **{TOP}/{BTM}** ... {inizio}/{fine} del programma
- **{SRC}** ... {ricerca}
- **{MENU}** ... {menu dei modi}
- **{SYBL}** ... {menu dei simboli}

●Per cambiare i modi in un programma

- La pressione di **[F4]** (MENU) mentre lo schermo per l'introduzione del programma è visualizzato sul display fa apparire un menu di cambiamento modo. È possibile usare questo menu per introdurre i cambiamenti di modo nei programmi creati.

- **{STAT}/{MAT}/{LIST}/{GRPH}/{DYNA}/{TABL}/{RECR}**

Per i dettagli su ciascuno di questi modi, fare riferimento a "Per scegliere un'icona" e alle sezioni di questo manuale che descrivono le operazioni possibili in ciascun modo.

- Il seguente menu appare ogni volta che si preme **[F4]** (MENU) durante l'introduzione di un programma che implica specificazioni di base numerica.
- **{d ~ o}/{LOG}**



Pag. 365



Pag. 364



Pag. 3



Pag. 5

- La pressione di **[F6]** (SYBL) visualizza un menu di simboli (' , " , ~ , * , / , #) che possono essere introdotti in un programma.
- La pressione di **[SHIFT] [SETUP]** visualizza un menu di comandi che possono essere usati per cambiare le impostazioni dello schermo di impostazione all'interno di un programma.
- **{ANGL}/{COOR}/{GRID}/{AXES}/{LABL}/{DISP}/{P/L} {DRAW}/{DERV}/{BACK}/{FUNC}/{SIML}/{S-WIN}/{LIST}/{LOCS} {T-VAR}/{ΣDSP}/{RESID}**

Per i dettagli su ciascuno di questi comandi, fare riferimento a "Menu dei tasti di funzione sullo schermo di impostazione".

Il seguente menu dei tasti di funzione appare se si preme **[SHIFT] [SETUP]** durante l'introduzione di un programma che contiene calcoli binari, ottali, decimali o esadecimali.

- **{Dec}/{Hex}/{Bin}/{Oct}**

Il contenuto effettivo di un programma è identico ai calcoli manuali. Quanto segue mostra in che modo viene eseguito il calcolo dell'area e del volume di un ottaedro regolare usando un calcolo manuale.

Area S **[2] [X] [SHIFT] [✓] [3] [X] <valore di A> [X²] [EXE]**
 Volume V **[SHIFT] [✓] [2] [÷] [3] [X] <valore di A> [∧] [3] [EXE]**

È possibile eseguire questo calcolo anche assegnando il valore della lunghezza di un lato alla variabile A.

Lunghezza di un lato A
 <valore di A> **[→] [ALPHA] [A] [EXE]**
 Area S **[2] [X] [SHIFT] [✓] [3] [X] [ALPHA] [A] [X²] [EXE]**
 Volume V **[SHIFT] [✓] [2] [÷] [3] [X] [ALPHA] [A] [∧] [3] [EXE]**

Tuttavia, se si introducono semplicemente i calcoli manuali come mostrato qui sopra, la calcolatrice li esegue dall'inizio alla fine, senza interruzioni. I seguenti comandi consentono di interrompere un calcolo per l'introduzione di valori e la visualizzazione di risultati intermedi.

- ?: Questo comando sospende temporaneamente l'esecuzione del programma e visualizza un punto interrogativo come segnale di pronto per l'introduzione di un valore da assegnare ad una variabile. La sintassi per questo comando è:
 ? → <nome della variabile>.
- ▲: Questo comando sospende temporaneamente l'esecuzione del programma e visualizza l'ultimo risultato di calcolo ottenuto o il testo. È simile alla pressione di **[EXE]** in un calcolo manuale.



Pag. 369

- Per i dettagli sull'uso di questi e altri comandi, fare riferimento a "Utili comandi per i programmi".

I seguenti sono degli esempi sull'uso effettivo dei comandi ? e ▲.

SHIFT PRGM F4(?) → ALPHA A F6(>) F5(:)

2 X SHIFT ✓ 3 X ALPHA A X²

F6(>) F5(▲)

SHIFT ✓ 2 ÷ 3 X ALPHA A ▲ 3

```
=====OCTA =====
?→A:2×√3×A².
√2+3×A³_
```

SHIFT QUIT o EXIT EXIT

```
Program List
001H : ST
```

●Per eseguire un programma

1. Mentre la lista dei programmi è visualizzata sul display, usare ▲ e ▼ per evidenziare il nome del programma che si desidera eseguire.

2. Premere F1 (EXE) o EXE per eseguire il programma.

Proviamo ad eseguire il programma introdotto sopra.

Lunghezza di un lato (A)	Area (S)	Volume (V)
7 cm	169,7409791 cm ²	161,6917506 cm ³
10 cm	346,4101615 cm ²	471,4045208 cm ³
15 cm	779,4228634 cm ²	1590,990258 cm ³

```
Program List
001H : ST
```

F1 (EXE) o EXE

```
?
?
```

7 EXE
(Valore di A)

```
?
7
169.7409791
- DISP -
```

Risultato intermedio prodotto da ▲

EXE EXE

```
?
7
169.7409791
161.6917506
?
```

1 0 EXE

```
?
7
169.7409791
161.6917506
?
10
346.4101615
- DISP -
```

[EXE]

7	169.7409791
	161.6917506
?	
i0	346.4101615
	471.4045208

⋮

⋮



Pag. 378

- La pressione di [EXE] mentre il risultato finale del programma è visualizzato sul display riesegue il programma.
- È possibile eseguire un programma anche mentre si è nel **modo RUN** introducendo: Prog "<nome di file>" [EXE].
- Si verificherà un errore se il programma specificato da Prog "<nome di file>" non può essere trovato.

20-3 Messa a punto di un programma

Un problema in un programma che impedisce al programma di essere eseguito correttamente viene detto “baco” (errore), e il processo per l’eliminazione di tali problemi viene detto “messa a punto”. I seguenti sintomi indicano che il programma contiene degli errori e che è necessaria la messa a punto.

- Messaggi di errore che appaiono durante l’esecuzione di un programma
- Risultati che non sono quelli previsti

●Per eliminare gli errori che causano la comparsa dei messaggi di errore

Un messaggio di errore, come quello mostrato qui sotto, appare ogni volta che avviene qualcosa di illegale durante l’esecuzione di un programma.

```
Ma ERROR
```



Pag. 436

Quando appare tale messaggio, premere ◀ o ▶ per visualizzare il punto in cui è stato generato l’errore, insieme al cursore. Consultare la “Tabella dei messaggi di errore” per apprendere i rimedi da adottare per risolvere la situazione.

Pag. 360

- Notare che la pressione di ◀ o ▶ non visualizza la posizione dell’errore se il programma è protetto da una parola d’ordine.

●Per eliminare gli errori che causano risultati errati

Se il programma creato produce risultati che non sono quelli normalmente previsti, controllare il contenuto del programma ed effettuare i cambiamenti necessari.

Fare riferimento a “Modifica dei nomi di file e del contenuto di un programma” per i dettagli su come cambiare il contenuto di un programma.



Pag. 365

20-4 Calcolo del numero di byte utilizzati da un programma

Esistono due tipi di comandi: comandi da 1 byte* e comandi da 2 byte*.

* Un byte è un'unità di memoria che può essere utilizzata per la memorizzazione dei dati.

- Esempi di comandi da 1 byte: sin, cos, tan, log, (,), A, B, C, 1, 2, ecc.
- Esempi di comandi da 2 byte: Lbl 1, Goto 2, ecc.

Mentre il cursore si trova all'interno di un programma, ogni pressione di ◀ o ▶ fa spostare il cursore di un byte.

- È possibile controllare la quantità di memoria utilizzata e la quantità di memoria rimanente in qualsiasi momento scegliendo l'icona **MEM** nel menu principale ed entrando nel modo MEM. Per i dettagli, fare riferimento a "Stato della memoria (MEM)".



Pag. 24

20-5 Funzione di segreto

Quando si introduce un programma, è possibile proteggerlo con una parola d'ordine che limita l'accesso al contenuto del programma a coloro che conoscono la parola d'ordine. I programmi protetti dalla parola d'ordine possono essere eseguiti da chiunque senza che sia necessario introdurre la parola d'ordine.

●Per registrare una parola d'ordine

Esempio Per creare il file di un programma sotto il nome AREA e proteggerlo con la parola d'ordine CASIO

1. Mentre la lista dei programmi è visualizzata sul display, premere **F3** (NEW) e introdurre il nome di file del nuovo file di programma.

F3 (NEW)
A R E A

```
Program Name  
[AREA ]
```

2. Premere **F5** ($\pi 0$) e quindi introdurre la parola d'ordine.

F5 ($\pi 0$)
C A S I O

```
Program Name  
[AREA ]  
Password?  
[CASIO ]
```

- Il procedimento per l'introduzione della parola d'ordine è identico a quello per l'introduzione del nome di file.
3. Premere **EXE** per registrare il nome di file e la parola d'ordine. A questo punto è possibile introdurre il contenuto del file di programma.
 - La registrazione di una parola d'ordine utilizza 16 byte di memoria.
 - La pressione di **EXE** senza l'introduzione di una parola d'ordine registra soltanto il nome di file, senza una parola d'ordine.
 4. Dopo aver introdotto il programma, premere **SHIFT** **QUIT** per uscire dal file di programma e ritornare alla lista dei programmi. I file che sono protetti dalla parola d'ordine sono indicati da un asterisco alla destra del nome di file.

```
Program List  
OCIA : 37  
AREA * : 33
```

●Per richiamare un programma

Esempio Per richiamare il file dal nome AREA che è protetto dalla parola d'ordine CASIO

1. Nella lista dei programmi, usare **▲** e **▼** per spostare l'evidenziatura sul nome del programma che si desidera richiamare.

2. Premere **F2** (EDIT).

```
Program Name  
[AREA ]  
Password?  
[ ]
```

3. Introdurre la parola d'ordine e premere **EXE** per richiamare il programma.

- Il messaggio **"Mismatch"** appare se si introduce una parola d'ordine errata.

20-6 Ricerca di un file

Sono disponibili tre diversi metodi per cercare un nome di file specifico.

●Per trovare un file con la ricerca a scorrimento

Esempio Per usare la ricerca a scorrimento per richiamare il programma dal nome OCTA

1. Mentre la lista dei programmi è visualizzata sul display, usare \blacktriangle e \blacktriangledown per scorrere la lista dei nomi di programma finché non si trova quello desiderato.

```
Program List
OCTA      : 17
TRIANGLE  : 17
AREA      * : 33
GRAPHICS  : 17
MEASURE   : 17
OCTONARY  : 17
[EXE EDIT NEW DEL DELV ]
```

[F2]

2. Quando l'evidenziazione si trova sul nome del file desiderato, premere [F2] (EDIT) per richiamarlo.

```
====OCTA====
2*A:2*√3*A²
√2+3*A^3
```

●Per trovare un file con la ricerca per nome di file

Esempio Per usare la ricerca per nome di file per richiamare il programma dal nome OCTA

1. Mentre la lista dei programmi è visualizzata sul display, premere [F3] (NEW) e introdurre il nome del file che si desidera trovare.
 - Se il file che si sta cercando è protetto dalla parola d'ordine, è necessario introdurre anche la parola d'ordine.

[F3] (NEW)

[O] [C] [T] [A]

```
Program Name
[OCTA ]
```

2. Premere [EXE] per richiamare il programma.
 - Se nessun programma ha un nome che corrisponde a quello introdotto, viene creato un nuovo file usando il nome introdotto.

●Per trovare un file con la ricerca per iniziali

Esempio Per usare la ricerca per iniziali per richiamare il programma dal nome OCTA

1. Mentre la lista dei programmi è visualizzata sul display, premere [F6] (\triangleright) [F1] (SRC) e introdurre le iniziali del nome del file che si desidera trovare.

[F6] (\triangleright) [F1] (SRC)

[O] [C] [T]

```
Search For Program
[OCT ]
```



Pag. 360

2. Premere **EXE** per avviare la ricerca.

```
Program List
OCTA      : 37
OCTONARY  : 17
```

- Vengono richiamati tutti i file con nomi di file che iniziano con le iniziali introdotte.
 - Se nessun programma ha un nome di file che inizia con le iniziali introdotte, il messaggio **“Not Found”** appare sul display. In questo caso, premere **EXIT** per far scomparire il messaggio di errore.
3. Usare **▲** e **▼** per evidenziare il nome di file del programma che si desidera richiamare e quindi premere **F2** (EDIT) per richiamarlo.

20-7 Ricerca di dati all'interno di un programma

Esempio Per cercare la lettera "A" all'interno del programma dal nome OCTA

1. Richiamare il programma.
2. Premere **F3** (SRC) e introdurre il dato che si desidera cercare.

```
=====OCTA=====
2+A:2*√3*A²
√2+3*A³
```

F3 (SRC)

ALPHA **A**

```
Search For Text
-----
A_
-----
SVB
```

- Non è possibile specificare il simbolo di cambiamento di riga (\downarrow) o il comando di visualizzazione (\blacktriangle) come dato da cercare.
3. Premere **EXE** per avviare la ricerca. Il contenuto del programma appare sullo schermo con il cursore situato sul primo caso del dato specificato.

```
=====OCTA=====
?+A:2*√3*A²
√2+3*A³
```

```
<Search> SVB
```

Indica che l'operazione di ricerca è in corso.

4. Premere **EXE** per trovare il secondo caso del dato.

```
=====OCTA=====
?+A:2*√3*A²
√2+3*A³
```

- Se nessun dato all'interno del programma corrisponde al dato specificato, il contenuto del programma appare con il cursore situato al punto da cui era stata iniziata la ricerca.
- Dopo che il contenuto del programma è stato visualizzato sullo schermo, è possibile usare i tasti del cursore per spostare il cursore in un'altra posizione prima di cercare il caso successivo del dato. Viene cercata soltanto la parte del programma che inizia dalla posizione attuale del cursore quando si preme **EXE**.
- Dopo che l'operazione di ricerca ha localizzato un caso del dato desiderato, l'introduzione di caratteri o lo spostamento del cursore causa la cancellazione dell'operazione di ricerca (scomparsa dell'indicatore di ricerca dal display).
- Se si commette un errore durante l'introduzione dei caratteri per la ricerca, premere **AC** per cancellare l'introduzione ed eseguire nuovamente l'introduzione dall'inizio.

20-8 Modifica dei nomi di file e del contenuto di un programma

●Per modificare un nome di file

Esempio Per cambiare il nome di un file da TRIANGLE in ANGLE

1. Mentre la lista dei programmi è visualizzata sul display, usare \blacktriangle e \blacktriangledown per spostare l'evidenziazione sul file di cui si desidera modificare il nome e quindi premere $\boxed{F6}$ (\triangleright) $\boxed{F2}$ (REN).

```
Rename
[ TRIANGLE ]
```

2. Effettuare i cambiamenti desiderati.

\boxed{DEL} \boxed{DEL} \boxed{DEL}

```
Rename
[ ANGLE ]
```

3. Premere \boxed{EXE} per registrare il nuovo nome e ritornare alla lista dei programmi.
 - Se le modifiche effettuate risultano in un nome di file che è identico al nome di un programma già memorizzato nella memoria, appare il messaggio "**Already Exists**". In questo caso, è possibile eseguire una delle seguenti due operazioni per correggere la situazione.
 - Premere \blacktriangleright o \blacktriangleleft per cancellare l'errore e ritornare allo schermo per l'introduzione del nome di file.
 - Premere \boxed{AC} per cancellare il nuovo nome di file e introdurne un altro.

●Per modificare il contenuto di un programma

1. Trovare il nome di file del programma desiderato nella lista dei programmi.
2. Richiamare il programma.

- I procedimenti usati per modificare il contenuto di un programma sono identici a quelli usati per modificare i calcoli manuali. Per i dettagli, fare riferimento a "Modifica di calcoli".
- I seguenti tasti di funzione sono utili anche quando si modifica il contenuto dei programmi.

$\boxed{F1}$ (TOP) Sposta il cursore all'inizio del programma.

```
=====OCTA =====
?→A:2×T3×A².
T2+3×A³
```

$\boxed{F2}$ (BTM) Sposta il cursore alla fine del programma.

```
=====OCTA =====
?→A:2×T3×A².
T2+3×A³_
```

Esempio 2 Per usare il programma OCTA per creare un programma che calcoli l'area e il volume di tetraedri regolari quando la lunghezza di un lato è nota.

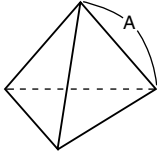


Pag. 20



Pag. 353

Usare TETRA come nome di file.



Lunghezza di un lato (A)	Area (S)	Volume (V)
7 cm	cm ²	cm ³
10 cm	cm ²	cm ³
15 cm	cm ²	cm ³

Le seguenti sono le formule per il calcolo dell'area (S) e del volume (V) di un tetraedro regolare di cui è nota la lunghezza di un lato.

$$S = \sqrt{3} A^2, \quad V = \frac{\sqrt{2}}{12} A^3$$

Usare le seguenti operazioni di tasto quando si introduce il programma.

Lunghezza di un lato (A) ... **SHIFT** **PRGM** **F4** (?) **→** **ALPHA** **A** **F6** (▷) **F5** (:)
 Area (S) **SHIFT** **✓** **3** **X** **ALPHA** **A** **x²** **F6** (▷) **F5** (▲)
 Volume (V) **SHIFT** **✓** **2** **÷** 1 2 **X** **ALPHA** **A** **△** **3**

Confrontare questo programma con il programma per il calcolo dell'area e del volume di un ottaedro regolare.

Lunghezza di un lato (A) ... **SHIFT** **PRGM** **F4** (?) **→** **ALPHA** **A** **F6** (▷) **F5** (:)
 Area (S) 2 **X** **SHIFT** **✓** **3** **X** **ALPHA** **A** **x²** **F6** (▷) **F5** (▲)
 Volume (V) **SHIFT** **✓** **2** **÷** 3 **X** **ALPHA** **A** **△** **3**

Come si può vedere, è possibile produrre il programma TETRA effettuando i seguenti cambiamenti nel programma OCTA.

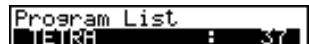
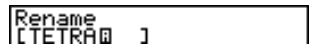
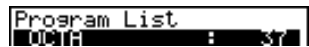
- Cancellazione di 2 **X** (elementi sottolineati con una linea ondulata sopra)
- Cambiamento di 3 in 1 2 (elementi sottolineati con una linea continua sopra)

Modifichiamo OCTA per produrre il programma TETRA.

1. Modificare il nome del programma.

F6 (▷) **F2** (REN) **T** **E** **T** **R** **A**

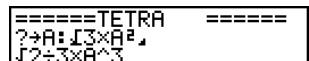
EXE



2. Modificare il contenuto del programma.

F2 (EDIT)

▶▶▶▶ DEL DEL



⏪ ⏩ SHIFT INS 1 2

```
====TETRA====
?-A:√3×A²,
√2+123×A³
```

DEL

```
====TETRA====
?-A:√3×A²,
√2+123×A³
```

SHIFT QUIT

Proviamo ad eseguire il programma.

Lunghezza di un lato (A)	Area (S)	Volume (V)
7 cm	84,87048957 cm ²	40,42293766 cm ³
10 cm	173,2050808 cm ²	117,8511302 cm ³
15 cm	389,7114317 cm ²	397,7475644 cm ³

F1 (EXE) o EXE

```
?

```

7 EXE

(Valore di A)

```
?
7
84.87048957
- DISP -
```

EXE EXE

```
?
7
84.87048957
40.42293766
?
```

1 0 EXE

```
?
7
84.87048957
40.42293766
?
10
173.2050808
- DISP -
```

EXE

```
?
7
84.87048957
40.42293766
?
10
173.2050808
117.8511302
```

⋮

⋮

20-9 Cancellazione di un programma

Sono disponibili due diversi metodi per cancellare un nome di file e il suo programma.

●Per cancellare un programma specifico

1. Mentre la lista dei programmi è visualizzata sul display, usare ▲ e ▼ per spostare l'evidenziatura sul nome del programma che si desidera cancellare.
2. Premere **F4** (DEL).
3. Premere **F1** (YES) per cancellare il programma scelto o **F6** (NO) per sospendere l'operazione senza cancellare nulla.

●Per cancellare tutti i programmi

1. Mentre la lista dei programmi è visualizzata sul display, premere **F5** (DEL-A).
2. Premere **F1** (YES) per cancellare tutti i programmi nella lista o **F6** (NO) per sospendere l'operazione senza cancellare nulla.
 - È possibile cancellare tutti i programmi anche usando il **modo MEM**. Per i dettagli, fare riferimento a "Cancellazione del contenuto della memoria".



Pag. 26

20-10 Utili comandi per i programmi

Oltre ai comandi per i calcoli, questa calcolatrice include anche una serie di comandi di relazione e di salto che possono essere usati per creare programmi che effettuino rapidamente e facilmente calcoli ripetuti.

Menu dei programmi

Premere **SHIFT** **PRGM** per visualizzare il menu dei programmi.

- **{COM}**/**{CTL}**/**{JUMP}**/**{CLR}**/**{DISP}**/**{REL}**/**{I/O}**
- **{?}** ... {comando di introduzione}
- **{▲}** ... {comando di emissione}
- **{:}** ... {comando di multiistruzione}

■ COM (menu dei comandi per i programmi)

La scelta di **{COM}** dal menu dei programmi visualizza le seguenti voci del menu delle funzioni.

- **{If}**/**{Then}**/**{Else}**/**{I-End}**/**{For}**/**{To}**/**{Step}**/**{Next}**/**{While}**/**{WEnd}**/**{Do}**/**{Lp-W}**
... Comandi **{If}**/**{Then}**/**{Else}**/**{IfEnd}**/**{For}**/**{To}**/**{Step}**/**{Next}**/**{While}**/**{WhileEnd}**/**{Do}**/**{LpWhile}**

■ CTL (menu dei comandi di controllo programmi)

La scelta di **{CTL}** dal menu dei programmi visualizza le seguenti voci del menu delle funzioni.

- **{Prog}**/**{Rtrn}**/**{Brk}**/**{Stop}** ... Comandi **{Prog}**/**{Return}**/**{Break}**/**{Stop}**

■ JUMP (menu dei comandi di salto)

La scelta di **{JUMP}** dal menu dei programmi visualizza le seguenti voci del menu delle funzioni.

- **{Lbl}**/**{Goto}** ... Comandi **{Lbl}**/**{Goto}**
- **{⇒}** ... {comando di salto}
- **{Isz}**/**{Dsz}** ... {salto e incremento}/**{salto e decremento}**

■ CLR (menu dei comandi di cancellazione)

La scelta di **{CLR}** dal menu dei programmi visualizza le seguenti voci del menu delle funzioni.

- **{Text}**/**{Grph}**/**{List}** ... Cancellazione di **{testo}**/**{grafico}**/**{lista}**

■ DISP (menu dei comandi di visualizzazione)

La scelta di {DISP} dal menu dei programmi visualizza le seguenti voci del menu delle funzioni.

- {Stat}/{Grph}/{Dyna} ... Tracciatura di {grafico statistico}/{grafico}/
{grafico dinamico}
- {F-Tbl} ... {menu dei comandi per tavola e grafico}
Le seguenti sono le voci che appaiono nel menu sopra menzionato.
 - {Tbl}/{G-Con}/{G-Plt} ... Comandi {DispF-Tbl}/{DrawFTG-Con}/
{DrawFTG-Plt}
- {R-Tbl} ... {calcolo di ricorsività e formula di ricorsività}
Le seguenti sono le voci che appaiono nel menu sopra menzionato.
 - {Tbl}/{Web}/{an-Cn}/{Σa-Cn}/{an-Pl}/{Σa-Pl} ... Comandi {DispR-Tbl}/
{DrawWeb}/{DrawR-Con}/{DrawRΣ-Con}/{DrawR-Plt}/{DrawRΣ-Plt}

■ REL (comandi degli operatori di relazione di salto condizionato)

La scelta di {REL} dal menu dei programmi visualizza le seguenti voci del menu delle funzioni.

- {=}/{≠}/{>}/{<}/{≥}/{≤} ... Operatori di relazione {=}/{≠}/{>}/{<}/{≥}/{≤}

■ I/O (comandi di introduzione/emissione)

La scelta di {I/O} dal menu dei programmi visualizza le seguenti voci del menu delle funzioni.

- {Lcte}/{Gtky}/{Send}/{Recv} ... Comandi {Locate}/{Getkey}/{Send}/{Receive}
- La comparsa del menu delle funzioni differisce leggermente per un programma che contiene calcoli binari, ottali, decimali o esadecimali, ma le funzioni del menu sono le stesse.

20-11 Riferimento per i comandi

■ Indice dei comandi

Break	378
ClrGraph	382
ClrList	382
ClrText	382
DispF-Tbl, DispR-Tbl	383
Do~LpWhile	377
DrawDyna	383
DrawFTG-Con, DrawFTG-Plt	383
DrawGraph	383
DrawR-Con, DrawR-Plt	384
DrawR Σ -Con, DrawR Σ -Plt	384
DrawStat	384
DrawWeb	384
Dsz	380
For~To~Next	375
For~To~Step~Next	376
Getkey	385
Goto~Lbl	380
If~Then	373
If~Then~Else	374
If~Then~Else~IfEnd	375
If~Then~IfEnd	374
Isz	381
Locate	385
Prog	378
Receive (.....	386
Return	379
Send (.....	387
Stop	379
While~WhileEnd	377
? (comando di introduzione)	372
▲ (comando di emissione)	372
: (comando di multiistruzione)	373
↶ (ritorno a capo)	373
⇒ (codice di salto)	381
=, ≠, >, <, ≥, ≤ (operatori di relazione)	387

Le seguenti sono le convenzioni utilizzate in questa sezione per la descrizione dei vari comandi.

- Testo in grassetto I comandi effettivi e gli altri elementi che devono sempre essere introdotti sono indicati in grassetto.
- {Parentesi graffe} Le parentesi graffe sono usate per racchiudere un numero di elementi, uno dei quali deve essere scelto quando si usa un comando. Non introdurre le parentesi graffe quando si introduce un comando.
- [Parentesi quadre] Le parentesi quadre sono usate per racchiudere gli elementi opzionali. Non introdurre le parentesi quadre quando si introduce un comando.
- Espressioni numeriche ... Espressioni numeriche (come 10, 10 + 20, A) indicano costanti, calcoli, costanti numeriche, ecc.
- Caratteri alfabetici I caratteri alfabetici indicano stringhe letterali (come AB).

■ Comandi operativi fondamentali

? (comando di introduzione)

Funzione: Richiede l'introduzione di valori per l'assegnazione alle variabili durante l'esecuzione del programma.

Sintassi: ? → <nome della variabile>

Esempio: ? → A ↵

Descrizione:

1. Questo comando interrompe momentaneamente l'esecuzione del programma e richiede l'introduzione di un valore o di un'espressione per l'assegnazione ad una variabile. Quando il comando di introduzione viene eseguito, "?" appare sul display e la calcolatrice rimane in attesa dell'introduzione.
2. L'introduzione in risposta al comando di introduzione deve essere un valore o un'espressione, e l'espressione non può essere una multiistruzione.

▲ (comando di emissione)

Funzione: Visualizza un risultato intermedio durante l'esecuzione del programma.

Descrizione:

1. Questo comando interrompe momentaneamente l'esecuzione del programma e visualizza il testo in caratteri alfabetici o il risultato del calcolo immediatamente precedente ad esso.
2. Il comando di emissione deve essere usato nei punti in cui normalmente si preme il tasto **EXE** durante un calcolo manuale.

: (comando di multiistruzione)

Funzione: Collega due istruzioni per l'esecuzione sequenziale senza interruzioni.

Descrizione:

1. A differenza del comando di emissione (\blacktriangleleft), le istruzioni collegate con il comando di multiistruzione vengono eseguite senza interruzioni.
2. Il comando di multiistruzione può essere usato per unire due espressioni di calcolo o due comandi.
3. È possibile usare anche un ritorno a capo indicato da \blacktriangleleft al posto del comando di multiistruzione.

 \blacktriangleleft (ritorno a capo)

Funzione: Collega due istruzioni per l'esecuzione sequenziale senza interruzioni.

Descrizione:

1. L'operazione del ritorno a capo è identica a quella del comando di multiistruzione.
2. L'uso di un ritorno a capo al posto del comando di multiistruzione rende il programma visualizzato più facile da leggere.

■ Comandi per i programmi (COM)**If~Then**

Funzione: L'istruzione Then viene eseguita solo quando la condizione If è vera (non zero).

Sintassi:

$$\text{If } \begin{array}{l} \text{<condizione>} \\ \text{espressione numerica} \end{array} \left\{ \begin{array}{c} \blacktriangleleft \\ : \\ \blacktriangleleft \end{array} \right\} \text{ Then } \text{<istruzione>} \left[\left\{ \begin{array}{c} \blacktriangleleft \\ : \\ \blacktriangleleft \end{array} \right\} \text{ <istruzione>} \right]$$

Parametri: Condizione, espressione numerica

Descrizione:

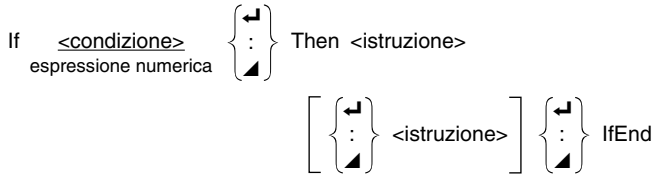
1. L'istruzione Then viene eseguita solo quando la condizione If è vera (non zero).
2. Se la condizione è falsa (0), l'istruzione Then non viene eseguita.
3. Una condizione If deve essere sempre accompagnata da un'istruzione Then. L'omissione dell'istruzione Then genera un errore.

Esempio: If A = 0 \blacktriangleleft
Then "A = 0"

If~Then~IfEnd

Funzione: L'istruzione Then viene eseguita solo quando la condizione If è vera (non zero). L'istruzione IfEnd viene sempre eseguita: dopo che è stata eseguita l'istruzione Then o direttamente dopo l'istruzione If quando la condizione If è falsa (0).

Sintassi:



Parametri: Condizione, espressione numerica

Descrizione:

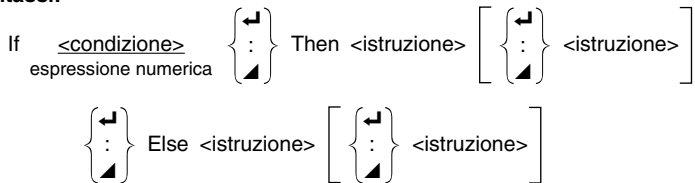
Questo comando è quasi identico a If~Then. L'unica differenza è che l'istruzione IfEnd viene sempre eseguita, indipendentemente da se la condizione If è vera (non zero) o falsa (0).

Esempio: If A = 0 ↵
 Then "A = 0" ↵
 IfEnd ↵
 "END"

If~Then~Else

Funzione: L'istruzione Then viene eseguita solo quando la condizione If è vera (non zero). L'istruzione Else viene eseguita quando la condizione If è falsa (0).

Sintassi:



Parametri: Condizione, espressione numerica

Descrizione:

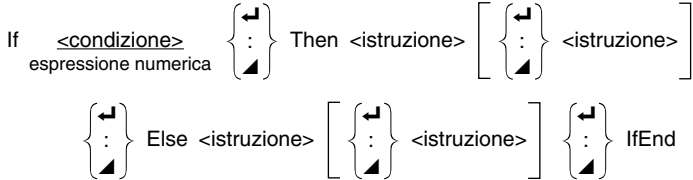
1. L'istruzione Then viene eseguita quando la condizione If è vera (non zero).
2. L'istruzione Else viene eseguita quando la condizione If è falsa (0).

Esempio: If A = 0 ↵
 Then "TRUE" ↵
 Else "FALSE"

If~Then~Else~IfEnd

Funzione: L'istruzione Then viene eseguita solo quando la condizione If è vera (non zero). L'istruzione Else viene eseguita quando la condizione If è falsa (0). L'istruzione IfEnd viene sempre eseguita dopo l'istruzione Then o l'istruzione Else.

Sintassi:



Parametri: Condizione, espressione numerica

Descrizione:

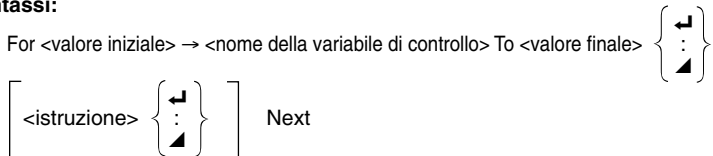
Questo comando è quasi identico a If~Then~Else. L'unica differenza è che l'istruzione IfEnd viene sempre eseguita, indipendentemente da se la condizione If è vera (non zero) o falsa (0).

Esempio: ? → A ↙
 If A = 0 ↙
 Then "TRUE" ↙
 Else "FALSE" ↙
 IfEnd ↙
 "END"

For~To~Next

Funzione: Questo comando ripete qualsiasi cosa si trovi fra l'istruzione For e l'istruzione Next. Il valore iniziale viene assegnato alla variabile di controllo con la prima esecuzione, e il valore della variabile di controllo viene incrementato di uno con ogni esecuzione. L'esecuzione continua fino a quando il valore della variabile di controllo supera il valore finale.

Sintassi:



Parametri:

- Nome della variabile di controllo: Da A a Z
- Valore iniziale: Valore o espressione che produce un valore (ad es. $\sin x$, A, ecc.)
- Valore finale: Valore o espressione che produce un valore (ad es. $\sin x$, A, ecc.)

Descrizione:

1. Quando il valore iniziale della variabile di controllo è maggiore del valore finale, l'esecuzione continua dall'istruzione dopo Next, senza l'esecuzione delle istruzioni tra For e Next.
2. Un'istruzione For deve sempre avere un'istruzione Next corrispondente, e l'istruzione Next deve sempre venire dopo la sua istruzione For corrispondente.
3. L'istruzione Next definisce la fine del ciclo creato da For~Next, e pertanto deve essere sempre inclusa, altrimenti si verificherà un errore.

Esempio: For 1 → A To 10 ↵

A × 3 → B ↵

B ▲

Next

For~To~Step~Next

Funzione: Questo comando ripete qualsiasi cosa si trovi fra l'istruzione For e l'istruzione Next. Il valore iniziale viene assegnato alla variabile di controllo con la prima esecuzione, e il valore della variabile di controllo viene cambiato a seconda del valore di passo con ogni esecuzione. L'esecuzione continua fino a quando il valore della variabile di controllo supera il valore finale.

Sintassi:

For <valore iniziale> → <nome della variabile di controllo> To <valore finale> Step <valore di passo> { ↵ }
 :
 ▲

Next

Parametri:

- Nome della variabile di controllo: Da A a Z
- Valore iniziale: Valore o espressione che produce un valore (ad es. $\sin x$, A, ecc.)
- Valore finale: Valore o espressione che produce un valore (ad es. $\sin x$, A, ecc.)
- Valore di passo: Valore numerico (l'omissione di questo valore imposta il passo su 1)

Descrizione:

1. Questo comando è fondamentalmente identico a For~To~Next. L'unica differenza è che è possibile specificare il passo.
2. L'omissione del valore di passo imposta automaticamente il passo su 1.

- Se si rende il valore iniziale minore del valore finale e se si specifica un valore di passo positivo, la variabile di controllo sarà incrementata con ogni esecuzione. Se si rende il valore iniziale maggiore del valore finale e se si specifica un valore di passo negativo, la variabile di controllo sarà decrementata con ogni esecuzione.

Esempio: For 1 → A To 10 Step 0.1 ↵
 A × 3 → B ↵
 B ▲
 Next

Do~LpWhile

Funzione: Questo comando ripete comandi specifici fintantoché la sua condizione è vera (non zero).

Sintassi:

Do {
 ↵
 :
 ▲
 } ~ LpWhile <espressione>

Parametri: Espressione

Descrizione:

- Questo comando ripete i comandi contenuti nel ciclo fintantoché la sua condizione è vera (non zero). Quando la condizione diventa falsa (0), l'esecuzione procede dall'istruzione dopo l'istruzione LpWhile.
- Poiché la condizione viene dopo l'istruzione LpWhile, la condizione viene testata (controllata) dopo che tutti i comandi all'interno del ciclo sono stati eseguiti.

Esempio: Do ↵
 ? → A ↵
 A × 2 → B ↵
 B ▲
 LpWhile B >10

While~WhileEnd

Funzione: Questo comando ripete comandi specifici fintantoché la sua condizione è vera (non zero).

Sintassi:

While <espressione> {
 ↵
 :
 ▲
 } ~ WhileEnd

Parametri: Espressione

Descrizione:

- Questo comando ripete i comandi contenuti nel ciclo fintantoché la sua condizione è vera (non zero). Quando la condizione diventa falsa (0), l'esecuzione procede dall'istruzione dopo l'istruzione WhileEnd.

2. Poiché la condizione viene dopo l'istruzione While, la condizione viene testata (controllata) prima che i comandi all'interno del ciclo siano eseguiti.

Esempio: 10 → A ↵
 While A > 0 ↵
 A - 1 → A ↵
 "GOOD" ↵
 WhileEnd

■ Comandi di controllo programmi (CTL)

Break

Funzione: Questo comando interrompe l'esecuzione di un ciclo e continua dal comando successivo dopo il ciclo.

Sintassi: Break ↵

Descrizione:

1. Questo comando interrompe l'esecuzione di un ciclo e continua dal comando successivo dopo il ciclo.
2. Questo comando può essere usato per interrompere l'esecuzione di un'istruzione For, di un'istruzione Do e di un'istruzione While.

Esempio: While A>0 ↵
 If A > 2 ↵
 Then Break ↵
 IfEnd ↵
 WhileEnd ↵
 A ▲ ←————— Eseguito dopo Break

Prog

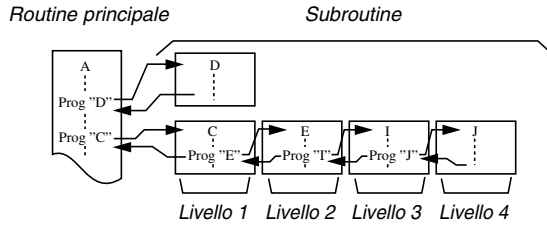
Funzione: Questo comando specifica l'esecuzione di un altro programma come subroutine. Nel modo RUN, questo comando esegue un nuovo programma.

Sintassi: Prog "nome di file" ↵

Esempio: Prog "ABC" ↵

Descrizione:

1. Anche quando questo comando si trova all'interno di un ciclo, la sua esecuzione interrompe immediatamente il ciclo e lancia la subroutine.
2. Questo comando può essere usato il numero di volte necessario all'interno di una routine principale per richiamare delle subroutine indipendenti per eseguire compiti specifici.
3. Una subroutine può essere utilizzata in più punti nella stessa routine principale, o può essere richiamata con qualsiasi numero di routine principali.



4. Il richiamo di una subroutine fa sì che essa venga eseguita dall'inizio. Al termine dell'esecuzione della subroutine, l'esecuzione ritorna alla routine principale, continuando dall'istruzione dopo il comando Prog.
5. Un comando Goto~Lbl all'interno di una subroutine è valido soltanto all'interno di quella subroutine. Esso non può essere usato per saltare ad un'etichetta al di fuori della subroutine.
6. Se una subroutine con il nome di file specificato dal comando Prog non esiste, si verifica un errore.
7. Nel **modo RUN**, l'introduzione del comando Prog e la pressione di **EXE** lancia il programma specificato dal comando.

Return

Funzione: Questo comando effettua il ritorno da una subroutine.

Sintassi: Return ↵

Descrizione:

L'esecuzione del comando Return all'interno di una routine principale fa interrompere l'esecuzione del programma.

Esempio: Prog "A" Prog "B"
 1 → A ↵ For A → B To 10 ↵
 Prog "B" ↵ B + 1 → C ↵
 C ▲ Next ↵
 Return

L'esecuzione del programma nel file A visualizza il risultato dell'operazione (11).

Stop

Funzione: Questo comando termina l'esecuzione di un programma.

Sintassi: Stop ↵

Descrizione:

1. Questo comando termina l'esecuzione di un programma.
2. L'esecuzione di questo comando all'interno di un ciclo termina l'esecuzione del programma senza che sia generato un errore.

Esempio: For 2 → I To 10 ↵
 If I = 5 ↵
 Then "STOP" : Stop ↵
 IfEnd ↵
 Next

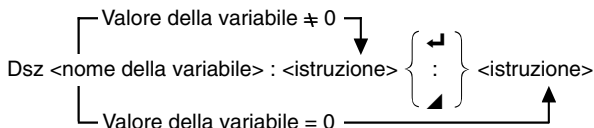
Questo programma conta da 2 a 10. Tuttavia, quando il conteggio raggiunge 5, esso termina l'esecuzione e visualizza il messaggio "STOP".

■ Comandi di salto (JUMP)

Dsz

Funzione: Questo comando è un salto di conteggio che decrementa di 1 il valore di una variabile di controllo, e quindi esegue il salto se il valore attuale della variabile è zero.

Sintassi:



Parametri:

Nome della variabile: Da A a Z, r, θ

[Esempio] Dsz B : Decrementa di 1 il valore assegnato alla variabile B.

Descrizione:

Questo comando decrementa di 1 il valore di una variabile di controllo, e quindi lo testa (controlla). Se il valore attuale è non zero, l'esecuzione continua con l'istruzione successiva. Se il valore attuale è zero, l'esecuzione salta all'istruzione dopo il comando di multiistruzione (:), il comando di visualizzazione (▲) o il ritorno a capo (↵).

Esempio: 10 → A : 0 → C :

Lbl 1 : ? → B : B+C → C :

Dsz A : Goto 1 : C ÷ 10

Questo programma richiede l'introduzione di 10 valori, e quindi calcola la media dei valori introdotti.

Goto~Lbl

Funzione: Questo comando esegue un salto incondizionato in un punto specifico.

Sintassi: Goto <valore o variabile> ~ Lbl <valore o variabile>

Parametri: Valore (da 0 a 9), variabile (da A a Z, r, θ)

Descrizione:

1. Questo comando è costituito da due parti: Goto *n* (in cui *n* è un valore da 0 a 9) e Lbl *n* (in cui *n* è il valore specificato per Goto). Questo comando fa saltare l'esecuzione del programma all'istruzione Lbl il cui valore corrisponde a quello specificato dall'istruzione Goto.

2. Questo comando può essere usato per indietreggiare fino all'inizio di un programma o per saltare ad un punto qualsiasi all'interno di un programma.
3. Questo comando può essere usato in combinazione con salti condizionati e salti di conteggio.
4. Se non c'è alcuna istruzione Lbl con un valore corrispondente a quello specificato dall'istruzione Goto, si verifica un errore.

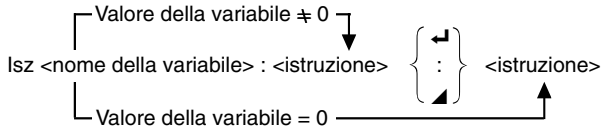
Esempio: ? → A : ? → B : Lbl 1 :
 ? → X : A × X + B ▲
 Goto 1

Questo programma calcola $y = AX + B$ per il numero di valori per ciascuna variabile che si desidera introdurre. Per abbandonare l'esecuzione di questo programma, premere **AC**.

Isz

Funzione: Questo comando è un salto di conteggio che incrementa di 1 il valore di una variabile di controllo, e quindi esegue il salto se il valore attuale della variabile è zero.

Sintassi:



Parametri:

Nome della variabile: Da A a Z, r, θ

[Esempio] Isz A : Incrementa il valore assegnato alla variabile A di 1.

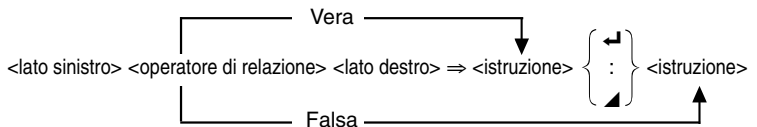
Descrizione:

Questo comando incrementa di 1 il valore di una variabile di controllo, e quindi lo testa (controlla). Se il valore attuale è non zero, l'esecuzione continua con l'istruzione successiva. Se il valore attuale è zero, l'esecuzione salta all'istruzione dopo il comando di multiistruzione (:), il comando di visualizzazione (▲) o il ritorno a capo (↵).

⇒ (codice di salto)

Funzione: Questo codice serve per impostare le condizioni per un salto condizionato. Il salto viene eseguito ogni volta che le condizioni sono false.

Sintassi:



**Parametri:**

Lato sinistro/lato destro: Variabile (da A a Z, r , θ), costante numerica, espressione per la variabile (come: $A \times 2$)

Operatore di relazione: =, \neq , >, <, \geq , \leq

Descrizione:

1. Il salto condizionato confronta il contenuto di due variabili o i risultati di due espressioni, e la decisione di eseguire o meno il salto viene presa in base al risultato del confronto.
2. Se il confronto restituisce un risultato vero, l'esecuzione continua con l'istruzione dopo il comando \Rightarrow . Se il confronto restituisce un risultato falso, l'esecuzione salta all'istruzione dopo il comando di multiistruzione (:), il comando di visualizzazione (\blacktriangleleft) o il ritorno a capo (\blacktriangledown).

Esempio: Lbl 1 : ? \rightarrow A :

$A \geq 0 \Rightarrow \sqrt{A} \blacktriangleleft$

Goto 1

Con questo programma, l'introduzione di un valore di zero o di un valore maggiore calcola e visualizza la radice quadrata del valore introdotto. L'introduzione di un valore minore di zero causa il ritorno al segnale di pronto per l'introduzione senza calcolare nulla.

■ Comandi di cancellazione (CLR)**ClrGraph**

Funzione: Questo comando cancella lo schermo del grafico.

Sintassi: ClrGraph \blacktriangledown

Descrizione: Questo comando cancella lo schermo del grafico durante l'esecuzione di un programma.

ClrList

Funzione: Questo comando cancella i dati della lista.

Sintassi: ClrList \blacktriangledown

Descrizione: Questo comando cancella il contenuto della lista attualmente scelta (da List 1 a List 6) durante l'esecuzione di un programma.

ClrText

Funzione: Questo comando cancella lo schermo del testo.

Sintassi: ClrText \blacktriangledown

Descrizione: Questo comando cancella il testo dallo schermo durante l'esecuzione di un programma.

Comandi di visualizzazione (DISP)**DispF-Tbl, DispR-Tbl**

Funzione: Questi comandi visualizzano tavole numeriche.

Sintassi:

DispF-Tbl ↵

DispR-Tbl ↵

Descrizione:

1. Questi comandi generano tavole numeriche durante l'esecuzione di un programma secondo le condizioni definite all'interno del programma.
2. DispF-Tbl genera una tavola di funzioni, mentre DispR-Tbl genera una tavola di ricorsività.

DrawDyna

Funzione: Questo comando esegue un'operazione di tracciatura di grafico dinamico.

Sintassi: DrawDyna ↵

Descrizione: Questo comando esegue un'operazione di tracciatura di grafico dinamico durante l'esecuzione di un programma secondo le condizioni di tracciatura definite all'interno del programma.

DrawFTG-Con, DrawFTG-Plt

Funzione: Questi comandi tracciano i grafici per le funzioni.

Sintassi:

DrawFTG-Con ↵

DrawFTG-Plt ↵

Descrizione:

1. Questi comandi tracciano i grafici per le funzioni secondo le condizioni definite all'interno del programma.
2. DrawFTG-Con produce un grafico del tipo a punti collegati, mentre DrawFTG-Plt produce un grafico del tipo a punti non collegati.

DrawGraph

Funzione: Questo comando traccia un grafico.

Sintassi: DrawGraph ↵

Descrizione: Questo comando traccia un grafico secondo le condizioni di tracciatura definite all'interno del programma.

DrawR-Con, DrawR-Plt

Funzione: Questi comandi tracciano i grafici per espressioni di ricorsività, con $a_n(b_n)$ come asse verticale e n come asse orizzontale.

Sintassi:

DrawR-Con ↵

DrawR-Plt ↵

Descrizione:

1. Questi comandi tracciano i grafici per espressioni di ricorsività, con $a_n(b_n)$ come asse verticale e n come asse orizzontale, secondo le condizioni definite all'interno del programma.
2. DrawR-Con produce un grafico del tipo a punti collegati, mentre DrawR-Plt produce un grafico del tipo a punti non collegati.

DrawR Σ -Con, DrawR Σ -Plt

Funzione: Questi comandi tracciano i grafici per espressioni di ricorsività, con $\Sigma a_n(\Sigma b_n)$ come asse verticale e n come asse orizzontale.

Sintassi:

DrawR Σ -Con ↵

DrawR Σ -Plt ↵

Descrizione:

1. Questi comandi tracciano i grafici per espressioni di ricorsività, con $\Sigma a_n(\Sigma b_n)$ come asse verticale e n come asse orizzontale, secondo le condizioni definite all'interno del programma.
2. DrawR Σ -Con produce un grafico del tipo a punti collegati, mentre DrawR Σ -Plt produce un grafico del tipo a punti non collegati.

DrawStat

Funzione: Questo comando traccia un grafico statistico.

Sintassi:

DrawStat ↵

Descrizione:

Questo comando traccia un grafico statistico secondo le condizioni definite all'interno del programma.

DrawWeb

Funzione: Questo comando traccia il grafico di convergenza/divergenza di un'espressione di ricorsività (grafico WEB).

Sintassi: DrawWeb [nome dell'espressione di ricorsività], [numero di righe] ↵

Esempio: DrawWeb $a_{n+1}(b_{n+1})$, 5 ↵

Descrizione:

1. Questo comando traccia il grafico di convergenza/divergenza di un'espressione di ricorsività (grafico WEB).
2. L'omissione della specificazione del numero di righe specifica automaticamente il valore default di 30.

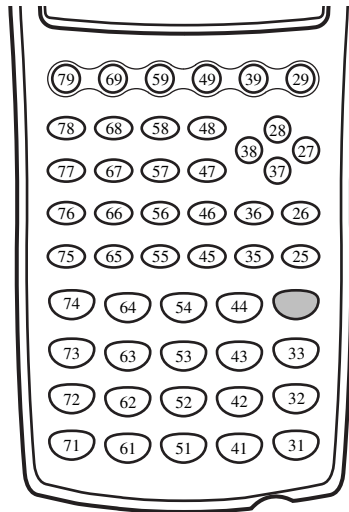
Comandi di introduzione/emissione (I/O)**Getkey**

Funzione: Questo comando restituisce il codice corrispondente al tasto premuto per ultimo.

Sintassi: Getkey ↵

Descrizione:

1. Questo comando restituisce il codice corrispondente al tasto premuto per ultimo.



2. Un valore di zero viene restituito se nessun tasto era stato premuto prima dell'esecuzione di questo comando.
3. Questo comando può essere usato all'interno di un ciclo.

Locate

Funzione: Questo comando visualizza caratteri alfanumerici in un punto specifico sullo schermo del testo.

Sintassi:

Locate <numero di colonna>, <numero di riga>, <valore>

Locate <numero di colonna>, <numero di riga>, <nome della variabile>

Locate <numero di colonna>, <numero di riga>, "<stringa>"

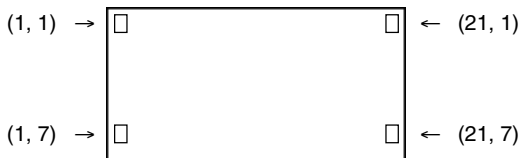
[Esempio] Locate 1, 1, "AB" ↵

Parametri:

- Numero di riga: Numero da 1 a 7
- Numero di colonna: Numero da 1 a 21
- Valore: Valore numerico
- Nome della variabile: Da A a Z
- Stringa: Stringa di caratteri

Descrizione:

1. Questo comando visualizza dei valori (incluso il contenuto della variabile) o un testo in un punto specifico sullo schermo del testo.
2. La riga viene specificata con un valore compreso fra 1 e 7, mentre la colonna viene indicata con un valore compreso fra 1 e 21.

**Esempio:** Cls ↵

Locate 7, 1, "CASIO CFX"

Questo programma visualizza il testo "CASIO CFX" al centro dello schermo.

- In alcuni casi, il comando ClrText deve essere eseguito prima dell'esecuzione del programma sopra descritto.

Receive (

Funzione: Questo comando riceve i dati da un dispositivo esterno.

Sintassi: Receive (<dati>)

Descrizione:

1. Questo comando riceve i dati da un dispositivo esterno.
2. I seguenti tipi di dati possono essere ricevuti con questo comando:
 - Singoli valori assegnati alle variabili
 - Dati di matrici (tutti i valori - singoli valori non possono essere specificati)
 - Dati di lista (tutti i valori - singoli valori non possono essere specificati)
 - Dati di immagine

Send (

Funzione: Questo comando invia i dati ad un dispositivo esterno.

Sintassi: Send (<dati>)

Descrizione:

1. Questo comando invia i dati ad un dispositivo esterno.
2. I seguenti tipi di dati possono essere inviati con questo comando:
 - Singoli valori assegnati alle variabili
 - Dati di matrici (tutti i valori - singoli valori non possono essere specificati)
 - Dati di lista (tutti i valori - singoli valori non possono essere specificati)

Operatori di relazione di salto condizionato (REL)

=, ≠, >, <, ≥, ≤

Funzione: Questi operatori di relazione sono usati in combinazione con il comando di salto condizionato.

Sintassi:

<lato sinistro> <operatore di relazione> <lato destro> ⇒ <istruzione> $\left\{ \begin{array}{c} \leftarrow \\ : \\ \blacktriangleleft \end{array} \right\}$ <istruzione>

Parametri:

Lato sinistro/lato destro: Variabile (da A a Z, r, θ), costante numerica, espressione per la variabile (come: A × 2)

Operatore di relazione: =, ≠, >, <, ≥, ≤

Descrizione:

1. I seguenti sei operatori di relazione possono essere usati nel comando di salto condizionato.
 - <lato sinistro> = <lato destro> : Vero se <lato sinistro> è uguale a <lato destro>
 - <lato sinistro> ≠ <lato destro> : Vero se <lato sinistro> non è uguale a <lato destro>
 - <lato sinistro> > <lato destro> : Vero se <lato sinistro> è maggiore di <lato destro>
 - <lato sinistro> < <lato destro> : Vero se <lato sinistro> è minore di <lato destro>
 - <lato sinistro> ≥ <lato destro> : Vero se <lato sinistro> è maggiore di o uguale a <lato destro>
 - <lato sinistro> ≤ <lato destro> : Vero se <lato sinistro> è minore di o uguale a <lato destro>
2. Fare riferimento a “⇒ (codice di salto)” per i dettagli sull’uso del salto condizionato.



20-12 Visualizzazione del testo

È possibile includere un testo in un programma semplicemente racchiudendolo fra virgolette doppie. Tale testo appare sul display durante l'esecuzione del programma, il che significa che è possibile aggiungere etichette a segnali di pronto per l'introduzione e a risultati.

Programma	Visualizzazione
? → X	?
"X =" ? → X	X = ?

- Se il testo è seguito da una formula di calcolo, accertarsi di inserire un comando di visualizzazione (▲), un ritorno a capo (↵) o un comando di multiistruzione (;) fra il testo e il calcolo.
- L'introduzione di più di 21 caratteri fa spostare il testo sulla riga successiva. La visualizzazione scorre automaticamente se il testo occupa completamente lo schermo.

20-13 Uso delle funzioni della calcolatrice nei programmi



Pag. 80

■ Uso delle operazioni sulle righe di una matrice in un programma

Questi comandi consentono di manipolare le righe di una matrice in un programma.

- Per questo tipo di programma, accertarsi di usare il **modo MAT** per introdurre la matrice, e quindi di passare al **modo PRGM** per introdurre il programma.

● Per scambiare il contenuto di due righe (Swap)

Esempio 1 Per scambiare i valori della riga 2 e della riga 3 nella seguente matrice:

$$\text{Matrice A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

La seguente è la sintassi per l'uso di questo programma.

Swap A, 2, 3

Nome della matrice

L'esecuzione di questo programma produce il seguente risultato.

(Modo MAT)

	1	2
1	3	2
2	5	6
3	1	4

● Per calcolare una moltiplicazione scalare (*Row)

Esempio 2 Per calcolare la moltiplicazione scalare della riga 2 della matrice nell'esempio 1, moltiplicando per 4

La seguente è la sintassi per l'uso di questo programma.

*Row 4, A, 2

Nome della matrice

Moltiplicatore

L'esecuzione di questo programma produce il seguente risultato.

(Modo MAT)

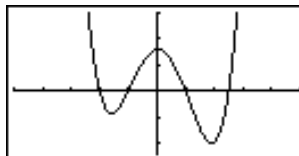
	1	2
1	3	2
2	12	16
3	1	4



- ③ Y = Type \leftarrow
- "X ^ 4 - X ^ 3 - 24X^2 + 4X + 80" \rightarrow Y1 \leftarrow
④
- ⑤ G SelOn 1 \leftarrow
- ⑥ Orange G1 \leftarrow
- ⑦ DrawGraph

- ③ F4 F4 F3 F1
- ④ VARS F4 F1 EXIT EXIT
- ⑤ F4 F4 F1 F1 EXIT
- ⑥ F4 F2
- ⑦ SHIFT PRGM F6 F2 F2

L'esecuzione di questo programma produce il risultato qui indicato.



■ Uso di funzioni per grafici dinamici in un programma

L'uso delle funzioni per i grafici dinamici in un programma rende possibile effettuare operazioni ripetute di grafico dinamico. Quanto segue indica come specificare la gamma per il grafico dinamico all'interno di un programma.

• Gamma per il grafico dinamico

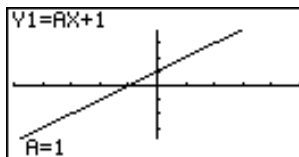
- 1 \rightarrow D Start \leftarrow
- 5 \rightarrow D End \leftarrow
- 1 \rightarrow D pitch \leftarrow

Esempio di programma

- ClrGraph \leftarrow
- View Window -5, 5, 1, -5, 5, 1 \leftarrow
- Y = Type \leftarrow

- "AX + 1" \rightarrow Y1 \leftarrow
①
 - ② D SelOn 1 \leftarrow
 - ③ D Var A \leftarrow
 - 1 \rightarrow ④ D Start \leftarrow
 - 5 \rightarrow ⑤ D End \leftarrow
 - 1 \rightarrow ⑥ D pitch \leftarrow
 - ⑦ DrawDyna
- ① VARS F4 F1 EXIT EXIT
 - ② F4 F5 F1
 - ③ F3
 - ④ VARS F5 F1
 - ⑤ F2
 - ⑥ F3
 - ⑦ SHIFT PRGM F6 F2 F3

L'esecuzione di questo programma produce il risultato qui indicato.



■ Uso delle funzioni di tavola e grafico in un programma

Le funzioni di tavola e grafico in un programma consentono di generare tavole numeriche e di eseguire operazioni di tracciatura di grafici. Quanto segue indica i vari tipi di sintassi che bisogna usare quando si esegue la programmazione con le funzioni di tavola e grafico.

- Impostazione della gamma per la tavola
 - 1 → F Start ↵
 - 5 → F End ↵
 - 1 → F pitch ↵
- Generazione della tavola numerica
 - DispF-Tbl ↵
- Operazione di tracciatura del grafico
 - Tipo a punti collegati: DrawFTG-Con ↵
 - Tipo a punti non collegati: DrawFTG-Pit ↵

Esempio di programma

ClrGraph ↵

ClrText ↵

View Window 0, 6, 1, -2, 106, 2 ↵

Y = Type ↵

"3X² - 2" → Y1 ↵

① T SelOn 1 ↵

0 → ② F Start ↵

6 → ③ F End ↵

1 → ④ F pitch ↵

⑤ DispF-Tbl ↵

⑥ DrawFTG-Con

① **F4** **F6** **F1** **F1**

② **VAR** **F6** **F1** **F1**

③ **F2**

④ **F3**

⑤ **SHIFT** **PRGM** **F6** **F2** **F4** **F1**

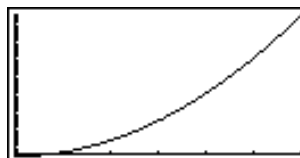
⑥ **SHIFT** **PRGM** **F6** **F2** **F4** **F2**

L'esecuzione di questo programma produce il risultato qui indicato.

Tavola numerica

X	Y1
0	-2
1	1
2	10
3	25

Grafico



■ Uso delle funzioni di tavola e grafico di ricorsività in un programma

L'incorporamento delle funzioni di tavola e grafico di ricorsività in un programma consente di generare tavole numeriche e di eseguire operazioni di tracciatura di grafici. Quanto segue indica i vari tipi di sintassi che bisogna usare quando si esegue la programmazione con le funzioni di tavola e grafico di ricorsività.

- Introduzione della formula di ricorsività
 a_{n+1} Type \leftarrow Specifica il tipo di grafico.
 "3 $a_n + 2$ " $\rightarrow a_{n+1}$ \leftarrow
 "4 $b_n + 6$ " $\rightarrow b_{n+1}$ \leftarrow
- Impostazione della gamma per la tavola
 1 \rightarrow R Start \leftarrow
 5 \rightarrow R End \leftarrow
 1 \rightarrow a_0 \leftarrow
 2 \rightarrow b_0 \leftarrow
 1 \rightarrow a_n Start \leftarrow
 3 \rightarrow b_n Start \leftarrow
- Generazione della tavola numerica
 DispR-Tbl \leftarrow
- Operazione di tracciatura del grafico
 Tipo a punti collegati: DrawR-Con \leftarrow , DrawR Σ -Con \leftarrow
 Tipo a punti non collegati: DrawR-Plt \leftarrow , DrawR Σ -Plt \leftarrow
- Grafico statistico di convergenza/divergenza (grafico WEB)
 DrawWeb a_{n+1} , 10 \leftarrow

Esempio di programma

```

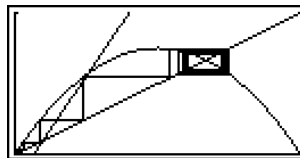
ClrGraph  $\leftarrow$ 
View Window 0, 1, 1, 0, 1, 1  $\leftarrow$ 
①  $a_{n+1}$  Type  $\leftarrow$  ① [F4] [F6] [F2] [F3] [F2] [EXIT]
" -3 $a_n^2 + 3a_n$ "  $\rightarrow a_{n+1}$   $\leftarrow$  ② [F4] [F2]
" 3 $b_n - 0.2$ "  $\rightarrow b_{n+1}$   $\leftarrow$ 
0  $\rightarrow$  ③ R Start  $\leftarrow$  ③ [VAR] [F6] [F2] [F2] [F1]
6  $\rightarrow$  R End  $\leftarrow$ 
0.01  $\rightarrow$   $a_0$   $\leftarrow$ 
0.11  $\rightarrow$   $b_0$   $\leftarrow$ 
0.01  $\rightarrow$   $a_n$  Start  $\leftarrow$ 
0.11  $\rightarrow$   $b_n$  Start  $\leftarrow$ 
④ DispR-Tbl  $\blacktriangleleft$  ④ [SHIFT] [PRGM] [F6] [F2] [F5] [F1]
⑤ DrawWeb ⑥  $a_{n+1}$ , 30 ⑤ [SHIFT] [PRGM] [F6] [F2] [F5] [F2] [EXIT] [EXIT] [EXIT]
⑥ [F4] [F6] [F2] [F4] [F3]
    
```

L'esecuzione di questo programma produce il risultato qui indicato.

Tavola numerica

$n+1$	$3n+1$	$bn+1$
0	0.01	0.11
1	0.0297	0.13
2	0.0864	0.19
3	0.2369	0.37

Grafico di ricorsività



Pag. 234

■ Uso delle funzioni di ordinamento di liste in un programma

Queste funzioni consentono di ordinare i dati nelle liste in ordine ascendente o in ordine discendente.

- Ordine ascendente

① SortA (List 1, List 2, List 3)

— Liste da ordinare (possono essere specificate fino a sei liste)

① **F4** **F3** **F1** **EXIT** ② **OPTN** **F1** **F1**

- Ordine discendente

SortD (List 1, List 2, List 3)

— Liste da ordinare (possono essere specificate fino a sei liste)

■ Uso della funzione di calcolo di risoluzione in un programma

È possibile incorporare una funzione di calcolo di risoluzione in un programma.

La seguente è la sintassi per l'uso della funzione di risoluzione in un programma.

Solve ($f(x)$, n , a , b)

— Limite superiore
— Limite inferiore
— Valore stimato iniziale

Esempio di programma

① Solve ($2X^2 + 7X - 9$, 1, 0, 1) ① **OPTN** **F4** **F1**

- Nella funzione $f(x)$, solo X può essere usata come una variabile nelle espressioni. Le altre variabili (da A a Z, r, θ) vengono considerate come costanti, e il valore attualmente assegnato a quella variabile viene applicato durante il calcolo.
- L'introduzione delle parentesi di chiusura, del limite inferiore a e del limite superiore b può essere omessa.
- Le soluzioni ottenute usando la funzione di risoluzione possono includere errori.
- Notare che non è possibile usare un'espressione di calcolo di risoluzione, differenziali, differenziali quadratici, integrazioni, valore massimo/minimo o calcoli Σ all'interno di un termine di un calcolo di risoluzione.



■ Uso dei calcoli statistici e dei grafici in un programma

L'inclusione di calcoli statistici e di operazioni di tracciatura di grafici in un programma consente di calcolare i dati statistici e di tracciarne il grafico.

● Per impostare le condizioni e tracciare un grafico statistico

Dopo "StatGraph" è necessario specificare le seguenti condizioni per il grafico:

- Stato di tracciatura/non tracciatura del grafico (DrawOn/DrawOff)
- Tipo di grafico
- Posizione del dato per l'asse delle x (nome della lista)
- Posizione del dato per l'asse delle y (nome della lista)
- Posizione del dato per la frequenza (nome della lista)
- Tipo di segno
- Colore per il grafico



Le condizioni per il grafico necessarie dipendono dal tipo di grafico. Fare riferimento a "Cambiamento dei parametri per il grafico".

- La seguente è una tipica specificazione delle condizioni per il grafico per un diagramma a nube di punti o per un grafico a spezzata xy .

S-Gph1 DrawOn, Scatter, List1, List2, 1, Square, Blue ↵

Nel caso di un grafico a spezzata xy , sostituire "Scatter" nella specificazione sopra indicata con " xy Line".

- La seguente è una tipica specificazione delle condizioni per il grafico per la tracciatura di punti di una probabilità normale.

S-Gph1 DrawOn, NPPlot, List1, Square, Blue ↵

- La seguente è una tipica specificazione delle condizioni per il grafico per un grafico a variabile singola.

S-Gph1 DrawOn, Hist, List1, List2, Blue ↵

Lo stesso formato può essere utilizzato per i seguenti tipi di grafici, sostituendo semplicemente "Hist" nella specificazione sopra indicata con il tipo di grafico applicabile.

Istogramma: Hist
 Riquadro per la mediana: MedBox
 Riquadro per la media: MeanBox
 Distribuzione normale: N-Dist
 Linea spezzata: Broken



- La seguente è una tipica specificazione delle condizioni per il grafico per un grafico di regressione.

S-Gph1 DrawOn, Linear, List1, List2, List3, Blue ↵

Lo stesso formato può essere utilizzato per i seguenti tipi di grafici, sostituendo semplicemente "Linear" nella specificazione sopra indicata con il tipo di grafico applicabile.

- Regressione lineare: Linear
- Mediana-Mediana: Med-Med
- Regressione quadratica: Quad
- Regressione cubica: Cubic
- Regressione quartica: Quart
- Regressione logaritmica: Log
- Regressione esponenziale: ... Exp
- Regressione per potenze: Power

- La seguente è una tipica specificazione delle condizioni per il grafico per un grafico di regressione sinusoidale.

S-Gph1 DrawOn, Sinusoidal, List1, List2, Blue ↵

- La seguente è una tipica specificazione delle condizioni per il grafico per un grafico di regressione logistica.

S-Gph1 DrawOn, Logistic, List1, List2, Blue ↵

Esempio di programma

ClrGraph ↵

① S-Wind Auto ↵

{1, 2, 3} → List 1 ↵

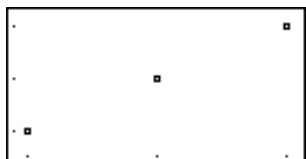
{1, 2, 3} → List 2 ↵

④ S-Gph1 DrawOn, Scatter, List1, List2, 1, Square, Blue ↵

⑨ DrawStat

①	SHIFT	SETUP	F6	F6	F3	F1
②	OPTN	F1	F1			
③	F1	EXIT	EXIT			
④	F4	F1	F2	F1	EXIT	
⑤	F1	F1	EXIT			
⑥	F2	F4	EXIT			
⑦	EXIT	F4	F1			
⑧	EXIT	F5	F1			
⑨	SHIFT	PRGM	F6	F2	F1	

L'esecuzione di questo programma produce il diagramma a nube di punti qui mostrato.



■ Esecuzione di calcoli statistici

- Calcolo statistico a variabile singola

① 1-Variable List 1, List 2

Dato per la frequenza (Frequency)

Dato per l'asse delle x (XList)

① **F4** **F1** **F6** **F1**

```
1-Variable
x̄ = 2.33333333
Σx = 14
Σx² = 36
x̄n = 0.74535599
x̄n-1 = 0.81649658
n = 6
```

- Calcolo statistico a doppia variabile

2-Variable List 1, List 2, List 3

Dato per la frequenza (Frequency)

Dato per l'asse delle y (YList)

Dato per l'asse delle x (XList)

```
2-Variable
x̄ = 2
Σx = 6
Σx² = 14
x̄n = 0.81649658
x̄n-1 = 1
n = 3
```

- Calcolo statistico di regressione

① LinearReg List 1, List 2, List 3

Tipo di
calcolo*

Dato per la frequenza (Frequency)

Dato per l'asse delle y (YList)

Dato per l'asse delle x (XList)

① **F4** **F1** **F6** **F6** **F1**

```
LinearRes
a = 1
b = 0
r = 1
r² = 1
y = ax + b
```

- * Uno qualsiasi dei seguenti calcoli può essere specificato come tipo di calcolo.

LinearReg Regressione lineare
 Med-MedLine . Calcolo di mediana-mediana
 QuadReg Regressione quadratica
 CubicReg Regressione cubica
 QuartReg Regressione quartica
 LogReg Regressione logaritmica
 ExpReg Regressione esponenziale
 PowerReg Regressione per potenze

- Calcolo statistico di regressione sinusoidale

SinReg List 1, List 2

_____ Dato per l'asse delle y (Y List)

_____ Dato per l'asse delle x (X List)

- Calcolo statistico di regressione logistica

LogisticReg List 1, List 2

_____ Dato per l'asse delle y (Y List)

_____ Dato per l'asse delle x (X List)