

TI-Nspire™ CX CAS

Guida di riferimento

Per sapere di più sulle tecnologie TI, consultare la guida online all'indirizzo
education.ti.com/eguide.

Informazioni importanti

A meno che non sia indicato diversamente nella licenza fornita con un programma, Texas Instruments, relativamente ai programmi o ai materiali di riferimento, non rilascia alcuna garanzia, né esplicita né implicita, ivi comprese, a mero titolo esemplificativo, garanzie implicite di commercialibilità e idoneità per uno scopo particolare. Tali materiali, quindi, sono disponibili solo "così come forniti". In nessun caso Texas Instruments potrà essere ritenuta responsabile di danni speciali, collaterali, accidentali o consequenti, collegati o riconducibili all'acquisto o all'utilizzo di tali materiali. A prescindere da qualunque tipo di azione legale eventualmente intrapresa, la responsabilità di Texas Instruments è limitata all'importo indicato nella licenza del programma. Texas Instruments, inoltre, non potrà essere ritenuta responsabile per eventuali reclami, di qualunque tipo, riguardanti l'utilizzo di tali materiali da parte di terzi.

© 2024 Texas Instruments Incorporated

I prodotti reali possono differire leggermente dalle immagini pubblicate.

Sommario

Modelli di espressione	1
Elenco alfabetico	7
A	7
B	16
C	20
D	36
E	45
F	53
G	61
I	71
L	80
M	95
N	104
O	112
P	115
Q	122
R	125
S	140
T	160
U	173
V	173
W	175
X	177
Z	178
Simboli	185
TI-Nspire™ CX II - Comandi di Disegna	205
Programmazione grafica	205
Schermata grafica	205
Impostazioni e vista predefinite	206
Messaggi di errore della schermata grafica	207
Comandi non validi in modalità grafico	207
C	209
D	210
F	214
G	216
P	217
T	219
U	221

Elementi vuoti (nulli)	222
Scelte rapide per l'inserimento di espressioni matematiche	224
EOS™ (Equation Operating System) gerarchia	226
TI-Nspire CX II - Funzioni di programmazione di TI-Basic	228
Rientro automatico in Program Editor	228
Miglioramento dei messaggi di errore per TI-Basic	228
Costanti e valori	231
Codici di errore e messaggi	232
Codici di avvertenza e messaggi	241
Informazioni Generali	243
Indice	244

Modelli di espressione

I modelli di espressione rappresentano un metodo veloce per introdurre espressioni matematiche in notazione matematica standard. Un modello, quando inserito, viene visualizzato nella riga di introduzione come tanti quadratini al posto degli elementi che si possono inserire. Un cursore indica quale elemento si può inserire.

Utilizzare i tasti freccia o premere **tab** per spostare il cursore su ciascun elemento e digitare un valore o un'espressione per esso. Premere **enter** o **ctrl enter** per calcolare l'espressione.

Modello di frazione

Tasti **ctrl** **÷**



Nota: vedere anche / (divisione), pagina 186.

Esempio:

$$\frac{12}{8 \cdot 2} \quad \frac{3}{4}$$

Modello di esponente

Tasto **^**



Nota: digitare il primo valore, premere **^**, quindi digitare l'esponente. Per riportare il cursore sulla linea di base, premere la freccia a destra (**▶**).

Nota: vedere anche **^ (potenza)**, pagina 186.

Esempio:

$$2^3 \quad 8$$

Modello di radice quadrata

Tasti **ctrl** **x²**



Nota: vedere anche **√()** (radice quadrata), pagina 194.

Esempio:

Modello di radice ennesima

Tasti **ctrl** **^**



Nota: vedere anche **root()**, pagina 137.

Esempio:

Modello di funzione esponenziale e

Tasti 

e

Esempio:

Esponenziale naturale e elevato a potenza

Nota: vedere anche $e^()$, pagina 45.

Modello di log

Tasto 

log

Esempio:

$\log_{\frac{1}{4}}(2.)$

0.5

Calcola il logaritmo nella base specificata. Per la base 10 predefinita, omettere la base.

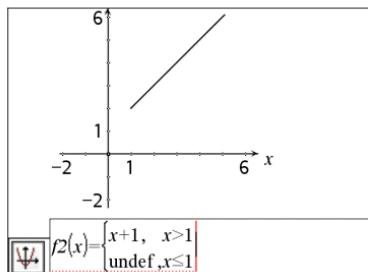
Nota: vedere anche $\log()$, pagina 91.

Modello di funzione piecewise a 2 tratti

Catalogo > 

Esempio:



Consente di creare espressioni e condizioni per funzioni definite a due-tratti. Per aggiungere un tratto, fare clic sul modello e ripeterlo.

Nota: vedere anche $\text{piecewise}()$, pagina 116.

Modello di funzione piecewise a N tratti

Catalogo > 

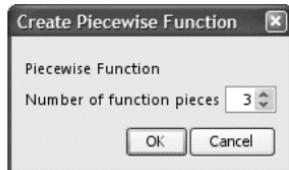
Consente di creare espressioni e condizioni per una funzione definita a N-tratti. Richiede l'introduzione di N .

Esempio:

vedere l'esempio del Modello di funzione piecewise a 2 tratti.

Modello di funzione piecewise a N tratti

Catalogo > 



Nota: vedere anche **piecewise()**, pagina 116.

Modello di sistema di 2 equazioni

Catalogo > 



Crea un sistema di due equazioni. Per aggiungere una riga a un sistema esistente, fare clic sul modello e ripeterlo.

Nota: vedere anche **system()**, pagina 160.

Esempio:

$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y\right) \quad x=\frac{5}{2} \text{ and } y=-\frac{5}{2}$$

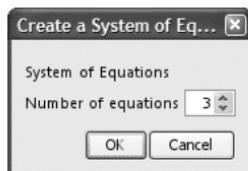
$$\text{solve}\left(\begin{cases} y=x^2-2 \\ x+2 \cdot y=-1 \end{cases}, x, y\right)$$

$$x=\frac{-3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1$$

Modello di sistema di N equazioni

Catalogo > 

Consente di creare un sistema di N . Richiede l'introduzione di N .



Nota: vedere anche **system()**, pagina 160.

Esempio:

vedere l'esempio del Modello di sistema di equazioni (2 equazioni).

Modello di valore assoluto

Catalogo > 

Nota: vedere anche **abs()**, pagina 7.

Esempio:

$$\left\{ 2, -3, 4, -4^3 \right\} \quad \{ 2, 3, 4, 64 \}$$

Modello di gg°pp'ss.ss"

Catalogo > 

 0°00"

Esempio:

Consente di inserire angoli nel formato gg°pp'ss.ss", dove gg è il numero di gradi decimali, pp è il numero di primi e ss.ss è il numero di secondi.

Modello di matrice (2 x 2)

Catalogo > 

 $\begin{bmatrix} \square & \square \\ \square & \square \end{bmatrix}$

Esempio:

Crea una matrice 2 x 2.

Modello di matrice (1 x 2)

Catalogo > 

 $\begin{bmatrix} \square & \square \end{bmatrix}$

Esempio:

$\text{crossP}([1 \ 2], [3 \ 4])$ $\begin{bmatrix} 0 & 0 & -2 \end{bmatrix}$

Modello di matrice (2 x 1)

Catalogo > 

 $\begin{bmatrix} \square \\ \square \end{bmatrix}$

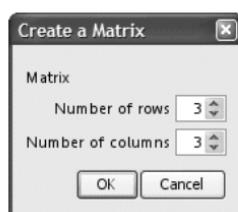
Esempio:

$\begin{bmatrix} 5 \\ 8 \end{bmatrix} \cdot 0.01$ $\begin{bmatrix} 0.05 \\ 0.08 \end{bmatrix}$

Modello di matrice (m x n)

Catalogo > 

Il modello appare dopo la richiesta di specificare il numero di righe e colonne.



Esempio:

$\text{diag} \begin{bmatrix} 4 & 2 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$ $\begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$

Modello di matrice ($m \times n$)

Catalogo > 

Nota: se si crea una matrice con un numero elevato di righe e colonne, è possibile che la visualizzazione richieda un po' di tempo.

Modello di sommatoria (Σ)

Catalogo >

$$\sum_{i=1}^n (a_i)$$

Esempio:

$$\sum_{n=3}^7 (n)$$

Nota: vedere anche $\Sigma()$ (`sumSeq`), pagina 195.

Modello di prodotto (Π)

Catalogo >

$$\begin{array}{|c|c|}\hline \square & \square \\ \hline \square & \square \\ \hline \end{array} (\square)$$

Esempio:

$$\frac{5}{\prod_{n=1}^{\infty} \left(\frac{1}{n}\right)} = \frac{1}{120}$$

Nota: vedere anche $\Pi()$ (**prodSeq**),
pagina 194.

Modello di derivata prima

Catalogo >

$$\frac{d}{d\tau}(\square)$$

Esempio:

Nota: vedere anche **d()** (derivata), pagina 194.

Modello di derivata seconda

Catalogo >

$$\frac{d^2}{d[\cdot]^2}(\cdot)$$

Esempio:

Nota: vedere anche **d()** (derivata), pagina 194.

$$\int_{\underline{a}}^{\underline{b}} f(x) dx$$

Esempio:

Elenco alfabetico

Gli elementi i cui nomi sono composti da caratteri non alfabetici (come ad esempio +, !, >) sono elencati alla fine della presente sezione, pagina 185. Se non diversamente specificato, tutti gli esempi della presente sezione sono stati eseguiti in modalità reset predefinita e tutte le variabili sono intese come non definite.

A

abs() (Valore assoluto)

Catalogo > 

$\text{abs}(\text{Lista}1) \Rightarrow \text{lista}$

$\text{abs}(\text{Matrice}1) \Rightarrow \text{matrice}$

Restituisce il valore assoluto dell'argomento.

Nota: vedere anche **Modello di valore assoluto**, pagina 3.

Se l'argomento è un numero complesso, restituisce il modulo del numero.

Nota: tutte le variabili non definite vengono considerate come variabili reali.

amortTbl()

Catalogo > 

$\text{amortTbl}(NPmt, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [\text{valoreArrotondato}]) \Rightarrow \text{matrice}$

Funzione di ammortamento che restituisce una matrice come una tabella di ammortamento per un set di argomenti TVM.

$NPmt$ è il numero di rate da includere nella tabella. La tabella inizia con la prima rata.

$N, I, PV, Pmt, FV, PpY, CpY$ e $PmtAt$ sono descritti nella tabella degli argomenti TVM, pagina 170.

$\text{amortTbl}(12, 60, 10, 5000, , , 12, 12)$

0	0.	0.	5000.
1	-41.67	-64.57	4935.43
2	-41.13	-65.11	4870.32
3	-40.59	-65.65	4804.67
4	-40.04	-66.2	4738.47
5	-39.49	-66.75	4671.72
6	-38.93	-67.31	4604.41
7	-38.37	-67.87	4536.54
8	-37.8	-68.44	4468.1
9	-37.23	-69.01	4399.09
10	-36.66	-69.58	4329.51
11	-36.08	-70.16	4259.35
12	-35.49	-70.75	4188.6

- Se si omette Pmt , viene utilizzata l'impostazione predefinita $Pmt=tvmPmt$ ($N, I, PV, FV, PpY, CpY, PmtAt$).
- Se si omette FV , viene utilizzata l'impostazione predefinita $FV=0$.

- Le impostazioni predefinite di *PpY*, *CpY* e *PmtAt* sono le stesse delle funzioni TVM.

valoreArrotondato specifica il numero di cifre decimali di arrotondamento. Impostazione predefinita=2.

Le colonne nella matrice risultante appaiono nel seguente ordine: numero di rate, interesse pagato, capitale versato e saldo.

Il saldo visualizzato nella riga *n* è il saldo dopo la rata *n*.

È possibile utilizzare la matrice di output come input per le altre funzioni di ammortamento *ΣInt()* e *ΣPrn()*, pagina 196, e *bal()*, pagina 16.

and

Espressione booleana1 and

Espressione booleana2 \Rightarrow *Espressione booleana*

$$\begin{array}{c} x \geq 3 \text{ and } x \geq 4 \\ \{x \geq 3, x \leq 0\} \text{ and } \{x \geq 4, x \leq -2\} \end{array} \quad \begin{array}{c} x \geq 4 \\ \{x \geq 4, x \leq -2\} \end{array}$$

Lista booleana1 and Lista booleana2 \Rightarrow *Lista booleana*

Matrice booleana1 and Matrice booleana2 \Rightarrow *Matrice booleana*

Restituisce vero o falso o una forma semplificata dell'espressione immessa originariamente.

Intero1 and Intero2 \Rightarrow *intero*

Confronta due interi reali bit per bit utilizzando un'operazione **and**. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati bit corrispondenti, il risultato sarà 1 se entrambi sono uguali a 1; in caso contrario il risultato sarà 0. Il valore restituito rappresenta il risultato dei bit e viene visualizzato nella modalità base che è stata impostata.

In modalità base Esadecimale:

$$0h7AC36 \text{ and } 0h3D5F \quad 0h2C16$$

Importante: è zero, non la lettera O.

In modalità base Bin:

$$0b100101 \text{ and } 0b100 \quad 0b100$$

In modalità base Dec:

È possibile inserire gli interi in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli interi vengono considerati decimali (base 10).

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato.

37 and 0b100

4

Nota: un numero binario può contenere fino a 64 cifre (oltre al prefisso 0b). Un numero esadecimale può contenere fino ad 16 cifre.

angle() (Angolo)

Catalogo > 

Restituisce l'angolo dell'argomento, interpretando l'argomento come numero complesso.

In modalità angolo in gradi:

angle(0+2·i)

90

In modalità angolo in gradi centesimali:

angle(0+3·i)

100

In modalità angolo in radianti:

angle(Lista1)⇒lista

angle(Matrice1)⇒matrice

Restituisce una lista o una matrice di angoli degli elementi contenuti in *Lista1* o *Matrice1*, interpretando ciascun elemento come un numero complesso che rappresenta un punto di coordinate rettangolari bidimensionali.

ANOVA (Analisi della varianza)

Catalogo > 

ANOVA Lista1,Lista2[,Lista3,...,Lista20][,Flag]

Esegue l'analisi della varianza a una dimensione per confrontare le medie di un numero di popolazioni compreso tra due e venti. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Flag=0 per Dati, *Flag*=1 per Statistiche

Variabile di output	Descrizione
stat.F	Valore della statistica F
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Grado di libertà dei gruppi
stat.SS	Somma dei quadrati dei gruppi
stat.MS	Quadrati medi dei gruppi
stat.dfError	Gradi di libertà degli errori
stat.SSError	Somma dei quadrati degli errori
stat.MSError	Quadrato medio degli errori
stat.sp	Deviazione standard aggregata
stat.xbarlist	Media dell'input delle liste
stat.CLowerList	Intervalli di confidenza al 95% per la media di ogni lista di input
stat.CUpperList	Intervalli di confidenza al 95% per la media di ogni lista di input

ANOVA2way *Lista1*,*Lista2*
[,*Lista3*,...*Lista10*][,RigaLiv]

Esegue l'analisi a due dimensioni della varianza per confrontare le medie di un numero di popolazioni compreso tra due e dieci. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

LevRow=0 per Blocco

LevRow=2,3,...,Len-1, per Due Fattori, dove Len=lunghezza(Lista1)=lunghezza(Lista2) = ... = lunghezza(Lista10) e Len / LevRow ∈ {2,3,...}

Output: design blocco

Variabile di output	Descrizione
stat.F	Statistica F del fattore colonna
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Gradi di libertà del fattore colonna
stat.SS	Somma dei quadrati del fattore colonna
stat.MS	Quadrati medi del fattore colonna
stat.FBlock	Statistica F per fattore
stat.PValBlock	Probabilità minima in cui l'ipotesi nulla può essere rifiutata
stat.dfBlock	Gradi di libertà per fattore
stat.SSBlock	Somma dei quadrati per fattore
stat.MSBlock	Quadrati medi per fattore
stat.dfError	Gradi di libertà degli errori
stat.SSError	Somma dei quadrati degli errori
stat.MSError	Quadrati medi degli errori
stat.s	Deviazione standard dell'errore

Output FATTORE COLONNA

Variabile di output	Descrizione
stat.Fcol	Statistica F del fattore colonna
stat.PValCol	Valore di probabilità del fattore colonna
stat.dfCol	Gradi di libertà del fattore colonna
stat.SSCol	Somma dei quadrati del fattore colonna
stat.MSCol	Quadrati medi del fattore colonna

Output FATTORE RIGA

Variabile di output	Descrizione
stat.FRow	Statistica F del fattore riga
stat.PValRow	Valore di probabilità del fattore riga
stat.dfRow	Gradi di libertà del fattore riga
stat.SSRow	Somma dei quadrati del fattore riga
stat.MSRow	Quadrati medi del fattore riga

Output di INTERAZIONE

Variabile di output	Descrizione
stat.FInteract	F dell'interazione
stat.PValInteract	Valore di probabilità dell'interazione
stat.dfInteract	Gradi di libertà dell'interazione
stat.SSInteract	Somma dei quadrati dell'interazione
stat.MSInteract	Quadrati medi dell'interazione

Output di ERRORE

Variabile di output	Descrizione
stat.dfError	Gradi di libertà degli errori
stat.SSError	Somma dei quadrati degli errori
stat.MSError	Quadrati medi degli errori
s	Deviazione standard dell'errore

Ans (Ultimo risultato)

Tasti ctrl (→)

Ans⇒*valore*

Restituisce il risultato dell'ultima espressione calcolata.

56	56
56+4	60
60+4	64

approx() Approssima

Restituisce il calcolo dell'argomento come espressione contenente valori decimali, ove possibile, indipendentemente dalla modalità corrente **Auto** o **Approssimato**.

Equivale a inserire l'argomento e a premere **ctrl** **enter**.

approx $\left(\frac{1}{3}\right)$	0.333333
approx $\left(\left\{\frac{1}{3}, \frac{1}{9}\right\}\right)$	{0.333333, 0.111111}
approx $\{\sin(\pi), \cos(\pi)\}$	{0., -1.}
approx $\left[\left[\sqrt{2} \quad \sqrt{3}\right]\right]$	[1.41421 1.73205]
approx $\left[\left[\frac{1}{3} \quad \frac{1}{9}\right]\right]$	[0.333333 0.111111]

approx(Lista1)⇒lista

approx(Matrice1)⇒matrice

Restituisce una lista o una *matrice* nella quale ciascun elemento è stato calcolato con valori decimali, ove possibile.

approx $\{\sin(\pi), \cos(\pi)\}$	{0., -1.}
approx $\left[\left[\sqrt{2} \quad \sqrt{3}\right]\right]$	[1.41421 1.73205]

►approxFraction()

Lista ► **approxFraction([Tol])⇒lista**

Matrice ► **approxFraction**
([Tol])⇒matrice

Restituisce l'argomento come frazione utilizzando una tolleranza *Tol*. Se *tol* è omesso, viene utilizzata una tolleranza di 5.E-14.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando @>**approxFraction** (...).

$\frac{1}{2} + \frac{1}{3} + \tan(\pi)$	0.833333
0.8333333333333333 ► approxFraction(5.E-14)	$\frac{5}{6}$
{π, 1.5} ► approxFraction(5.E-14)	$\left[\frac{5419351}{1725033}, \frac{3}{2}\right]$

approxRational()

approxRational(Lista[, tol])⇒lista

approxRational(*Matrice*[, tol])⇒matrice

Restituisce l'argomento come frazione utilizzando una tolleranza *tol*. Se *tol* è omesso, viene utilizzata una tolleranza di 5.E-14.

approxRational(0.333, 5·10 ⁻⁵)	$\frac{333}{1000}$
approxRational({0.2, 0.33, 4.125}, 5.E-14)	$\left\{\frac{1}{5}, \frac{33}{100}, \frac{33}{8}\right\}$

arccos()

Vedere $\cos^{-1}()$, pagina 28.

arccosh()

Vedere $\cosh^{-1}()$, pagina 29.

arccot()

Vedere $\cot^{-1}()$, pagina 30.

arccoth()

Vedere $\coth^{-1}()$, pagina 31.

arccsc()

Vedere $\csc^{-1}()$, pagina 33.

arccsch()

Vedere $\csch^{-1}()$, pagina 33.

arcsec()

Vedere $\sec^{-1}()$, pagina 141.

arcsech()

Vedere $\sech^{-1}()$, pagina 141.

arcsin()

Vedere $\sin^{-1}()$, pagina 149.

arcsinh()

Vedere $\sinh^{-1}()$, pagina 150.

augment() (Affianca(concatena))**augment**(*Lista1*, *Lista2*) \Rightarrow *lista*

Restituisce una nuova lista in cui *Lista2* viene aggiunta (accostata) alla fine di *Lista1*.

augment({1,-3,2},{5,4}) {1,-3,2,5,4}

augment(*Matrice1*, *Matrice2*) \Rightarrow *matrice*

Restituisce una nuova matrice in cui *Matrice2* viene aggiunta alla fine di *Matrice1*. Se si usa il carattere "," le matrici devono avere uguale numero di righe; *Matrice2* viene aggiunta a *Matrice1* come nuove colonne. Non modifica *Matrice1* o *Matrice2*.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
$\begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 \\ 6 \end{bmatrix}$
augment(<i>m1,m2</i>)	$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$

avgRC() (Tasso di variazione media)**avgRC**(*Espr1*, *Var* [=Valore] [, *Incr*]) \Rightarrow *espressione***avgRC**(*Espr1*, *Var* [=Valore] [, *Lista1*]) \Rightarrow *lista***avgRC**(*Lista1*, *Var* [=Valore] [, *Incr*]) \Rightarrow *lista***avgRC**(*Matrice1*, *Var* [=Valore] [, *Incr*]) \Rightarrow *matrice*

Restituisce il rapporto incrementale (tasso di variazione media).

Espr1 può essere un nome di funzione definito dall'utente (vedere **Func**).

Se il *Valore* è specificato, lo stesso sostituisce qualsiasi assegnazione di variabile precedente o qualsiasi sovrapposizione corrente "|" della variabile.

Incr è il valore di incremento. Se *Incr* è omesso, viene impostato per default su 0.001.

Si noti che la funzione simile **centralDiff()** utilizza la formula del rapporto incrementale bilaterale.

B

bal()

bal(*NPmt, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], valoreArrotondato*) \Rightarrow valore

bal
(*NPmt, tabellaAmmortamento*) \Rightarrow valore

Funzione di ammortamento che calcola il saldo del piano di rientro dopo una rata specificata.

N, I, PV, Pmt, FV, PpY, CpY e PmtAt sono descritti nella tabella degli argomenti TVM, pagina 170.

NPmt specifica il numero della rata a partire dalla quale deve essere calcolato il saldo.

N, I, PV, Pmt, FV, PpY, CpY e PmtAt sono descritti nella tabella degli argomenti TVM, pagina 170.

- Se si omette *Pmt*, viene utilizzata l'impostazione predefinita *Pmt=tvmPmt* (*N,I,PV,FV,PpY,CpY,PmtAt*).
- Se si omette *FV*, viene utilizzata l'impostazione predefinita *FV=0*.
- Le impostazioni predefinite di *PpY*, *CpY* e *PmtAt* sono le stesse delle funzioni TVM.

valoreArrotondato specifica il numero di cifre decimali di arrotondamento. Impostazione predefinita=2.

bal(5,6,5.75,5000,,12,12)	833.11
tbl:=amortTbl(6,6,5.75,5000,,12,12)	
0 0. 0. 5000.	
1 -23.35 -825.63 4174.37	
2 -19.49 -829.49 3344.88	
3 -15.62 -833.36 2511.52	
4 -11.73 -837.25 1674.27	
5 -7.82 -841.16 833.11	
6 -3.89 -845.09 -11.98	
bal(4,tbl)	1674.27

bal(*NPmt,tabellaAmmortamento*)
 calcola il saldo dopo la rata numero
NPmt sulla base della tabella di
 ammortamento *tabellaAmmortamento*.
 L'argomento *tabellaAmmortamento*
 deve essere una matrice avente la forma
 descritta in **amortTbl()**, pagina 7.

Nota: vedere anche **ΣInt()** e **ΣPrn()**,
 pagina 196.

►Base2

Intero1 ►Base2⇒*intero*

Nota: è possibile inserire questo
 operatore dalla tastiera del computer
 digitando @>Base2.

Converte *Intero1* in un numero binario. I
 numeri binari o esadecimali hanno
 sempre, rispettivamente, il prefisso 0b o
 0h. Zero, non la lettera O, seguito da b o
 h.

0b *numeroBinario*

0h *numeroEsadecimale*

Un numero binario può contenere fino a
 64 cifre. Un numero esadecimale può
 contenere fino ad 16 cifre.

Senza prefisso, *Intero1* viene
 considerato decimale (base 10). Il
 risultato viene visualizzato in modalità
 binaria, indipendentemente dalla
 modalità Base impostata.

I numeri negativi sono visualizzati nella
 forma “a complemento di due”. Ad
 esempio,

–1 è visualizzato come

0xFFFFFFFFFFFFFF in modalità base
 Esadecimale 0b11...111 (la cifra 1
 ripetuta 64 volte) in modalità base
 Binaria

256►Base2	0b100000000
0h1F►Base2	0b11111

-2^{63} è visualizzato come 0h8000000000000000 in modalità base Esadecimale e 0b100...000 (63 zeri) in modalità base Binaria

Se viene indicato un intero decimale esterno alla gamma di una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Considerare i seguenti esempi di valori esterni alla gamma.

2^{63} diventa -2^{63} ed è visualizzato come 0h8000000000000000 in modalità base Esadecimale e 0b100...000 (63 zeri) in modalità base Binaria

2^{64} diventa 0 ed è visualizzato come 0h0 in modalità base Esadecimale e 0b0 in modalità base Binaria

$-2^{63} - 1$ diventa $2^{63} - 1$ ed è visualizzato come 0h7FFFFFFFFFFFFF in modalità base Esadecimale e 0b111...111 (64 1) in modalità base Binaria

Interol ►Base10→*intero*

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Base10.

Converte *Interol* in numero decimale (base 10). Le voci binarie o esadecimale devono sempre avere, rispettivamente, il prefisso 0b o 0h.

0b *numeroBinario*

0h *numeroEsadecimale*

Zero, non la lettera O, seguito da b o h.

0b10011►Base10	19
0h1F►Base10	31

Un numero binario può contenere fino a 64 cifre. Un numero esadecimale può contenere fino ad 8 cifre.

Senza prefisso, *Interol* viene considerato decimale (base10). Il risultato viene visualizzato in modalità decimale, indipendentemente dalla modalità Base impostata.

►Base16

Interol ►Base16⇒*intero*

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Base16.

Converte *Interol* in un numero esadecimale. I numeri binari o esadecimali hanno sempre, rispettivamente, il prefisso 0b o 0h.

0b *numeroBinario*

0h *numeroEsadecimale*

Zero, non la lettera O, seguito da b o h.

Un numero binario può contenere fino a 64 cifre. Un numero esadecimale può contenere fino ad 16 cifre.

Senza prefisso, *Interol* viene considerato decimale (base 10). Il risultato viene visualizzato in modalità esadecimale, indipendentemente dalla modalità Base impostata.

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Per ulteriori informazioni, vedere ►Base2, pagina 17.

256►Base16	0h100
0b111100001111►Base16	0hF0F

binomCdf() (Funzione della probabilità cumulativa per la distribuzione binomiale)

Catalogo > 

binomCdf(*n,p*)⇒*lista*

binomCdf(*n,p,valoreInferiore, valoreSuperiore*)⇒*numero* se *valoreInferiore* e *valoreSuperiore* sono numeri, *lista* se *valoreInferiore* e *valoreSuperiore* sono liste

binomCdf(*n,p,valoreSuperiore*) per $P(0 \leq X \leq \text{valoreSuperiore})$ ⇒*numero* se *valoreSuperiore* è un numero, *lista* se *valoreSuperiore* è una lista

Calcola la probabilità cumulativa per la distribuzione binomiale discreta con il numero di prove *n* e la probabilità di successo *p* per ciascuna prova.

Per $P(X \leq \text{valoreSuperiore})$, impostare *valoreInferiore*=0

binomPdf()

Catalogo > 

binomPdf(*n,p*)⇒*lista*

binomPdf(*n,p,ValX*)⇒*numero* se *ValX* è un numero, *lista* se *ValX* è una lista

Calcola una probabilità in corrispondenza di *ValX* per la distribuzione binomiale discreta con il numero *n* di prove e la probabilità *p* di successo per ogni prova.

C

ceiling() (Arrotondato per eccesso)

Catalogo > 

Restituisce il più vicino numero intero ≤ all'argomento.

`ceiling(.456)`

1.

L'argomento può essere un numero reale o complesso.

Nota: vedere anche **floor()**.

ceiling() (Arrotondato per eccesso)**Catalogo > ****ceiling(Lista1)⇒lista****ceiling(Matrice1)⇒matrice**

Restituisce una lista o matrice del valore arrotondato per eccesso di ciascun elemento.

$\text{ceiling}(\{-3.1, 1.2, 5\})$	$\{-3, 1, 3\}$
$\text{ceiling}\begin{bmatrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{bmatrix}$	$\begin{bmatrix} 0 & -3 \cdot i \\ 2. & 4 \end{bmatrix}$

centralDiff()**Catalogo > ****centralDiff(Espr1, Var [=Valore] [,Incr])⇒espressione****centralDiff(Espr1, Var [,Incr]) | Var=Valore⇒espressione****centralDiff(Espr1, Var [=Valore] [,Lista])⇒lista****centralDiff(Lista1, Var [=Valore] [,Incr])⇒lista****centralDiff(Matrice1, Var [=Valore] [,Incr])⇒matrice**

Restituisce la derivata numerica utilizzando la formula del rapporto incrementale bilaterale.

Se il *Valore* è specificato, lo stesso sostituisce qualsiasi assegnazione di variabile precedente o qualsiasi sovrapposizione corrente “|” della variabile.

Incr è il valore di incremento. Se *Incr* è omesso, viene impostato per default su 0.001.

Se si utilizza *Lista1* o *Matrice1*, l'operazione viene mappata sui valori della lista o sugli elementi della matrice.

Nota: vedere anche.

char() (Stringa di caratteri)**Catalogo > ****char(Intero)⇒carattere**

$\text{char}(38)$	$"\&"$
$\text{char}(65)$	$"A"$

Restituisce un carattere stringa corrispondente al numero *Intero* del set di caratteri del palmare. L'intervallo valido per intero *Intero* è compreso tra 0 e 65535.

 χ^2 way

χ^2 way *MatriceOss*

chi22way *MatriceOss*

Esegue una verifica χ^2 per l'associazione di numeri nella tabella a due variabili nella matrice osservata *MatriceOss*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Per informazioni sull'effetto di elementi vuoti in una matrice, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat. χ^2	Statistica Chi quadrato: somma (osservati - attesi) 2 /attesi
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Gradi di libertà per le statistiche chi quadrato
stat.ExpMat	Matrice della tabella di numeri elementari attesi, assumendo l'ipotesi nulla
stat.CompMat	Matrice di contributi statistici chi quadrato elementari

 χ^2 Cdf()

χ^2 Cdf

(

valoreInferiore

,*valoreSuperiore*,*gl*) \Rightarrow numero se
valoreInferiore e *valoreSuperiore* sono
 numeri, lista se *valoreInferiore* e
valoreSuperiore sono liste

chi2Cdf

(

valoreInferiore,
valoreSuperiore,*gl*) \Rightarrow numero se
valoreInferiore e *valoreSuperiore* sono
numeri, *list* se *valoreInferiore* e
valoreSuperiore sono liste

Calcola la probabilità della distribuzione χ^2
tra il *valoreInferiore* e il *valoreSuperiore*
per i gradi di libertà *gl* specificati.

Per $P(X \leq \text{valoreSuperiore})$, impostare
valoreInferiore= 0.

Per informazioni sull'effetto di elementi
vuoti in una lista, vedere "Elementi vuoti
(nulli)", pagina 222.

$\chi^2\text{GOF}$ *listaOss*,*listaAtt*,*gl*

chi2GOF *listaOss*,*listaAtt*,*gl*

Esegue una verifica per confermare che i dati
del campione appartengono a una
popolazione conforme a una data
distribuzione. *listaOss* è una lista di conteggi
e deve contenere numeri interi. Il riepilogo
dei risultati è memorizzato nella variabile
stat.results. (pagina 155).

Per informazioni sull'effetto di elementi
vuoti in una lista, vedere "Elementi vuoti
(nulli)", pagina 222.

Variabile di output	Descrizione
<i>stat.</i> χ^2	Statistica Chi quadrato: $\text{sum}((\text{osservati} - \text{attesi})^2 / \text{attesi})$
<i>stat.PVal</i>	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
<i>stat.df</i>	Gradi di libertà per le statistiche chi quadrato
<i>stat.CompList</i>	Contributi statistici chi quadrato elementari

χ²Pdf(ValX,gl)⇒numero se *ValX* è un numero, lista se *ValX* è una lista

chi2Pdf(ValX,gl)⇒numero se *ValX* è un numero, lista se *ValX* è una lista

Calcola la funzione della densità di probabilità (pdf) per la distribuzione χ^2 a un dato valore *ValX* per i gradi di libertà *gl* specificati.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

ClearAZ

Cancella tutte le variabili con il nome di un solo carattere nello spazio attività corrente.

Se una o più variabili sono bloccate, questo comando visualizza un messaggio di errore ed elimina solo le variabili sbloccate. Vedere **unLock**, pagina 173.

ClrErr

Cancella lo stato di errore e imposta la variabile di sistema *errCode* su zero.

Per un esempio di **ClrErr**, vedere l'esempio 2 del comando **Try**, pagina 166.

L'istruzione **Else** del blocco **Try...Else...EndTry** dovrebbe utilizzare **ClrErr** o **PassErr**. Se l'errore deve essere elaborato o ignorato, utilizzare **ClrErr**. Se non si sa quale azione applicare all'errore, utilizzare **PassErr** per inviarlo al successivo blocco di gestione degli errori. Se non ci sono ulteriori blocchi di gestione degli errori **Try...Else...EndTry** in attesa di applicazione, la finestra di dialogo dell'errore viene visualizzata come normale.

Nota: vedere anche **PassErr**, pagina 116 e **Try**, pagina 166.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

colAugment() (Affianca colonna)

**colAugment(*Matrice1*,
Matrice2)**⇒*matrice*

Restituisce una nuova matrice in cui *Matrice2* viene aggiunta alla fine di *Matrice1*. Le matrici devono avere uguale numero di colonne; *Matrice2* viene aggiunta a *Matrice1* come nuove colonne. Non modifica *Matrice1* o *Matrice2*.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
$\begin{bmatrix} 5 & 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 & 6 \end{bmatrix}$
colAugment(<i>m1,m2</i>)	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$

colDim() (Dimensione colonna)

colDim(*Matrice*)⇒*espressione*

Restituisce il numero delle colonne contenute in *Matrice*.

colDim($\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}$)	3
--	---

Nota: vedere anche **rowDim()**.

colNorm() (Norma colonna)

colNorm(*Matrice*)⇒*espressione*

Restituisce il massimo delle somme dei valori assoluti degli elementi nelle colonne di *Matrice*.

$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$
colNorm(<i>mat</i>)	9

Nota: non sono ammessi elementi non definiti di una matrice. Vedere anche **rowNorm()**.

conj() (Coniugato)

conj(*Lista1*)⇒*lista*

conj(*Matrice1*)⇒*matrice*

Restituisce il complesso coniugato dell'argomento.

Nota: tutte le variabili non definite vengono considerate come variabili reali.

constructMat() (Costruisci matrice)

constructMat

```
(  
Espr  
,
```

 $Var1$
 $Var2, numRighe, numColonne) \Rightarrow matrice$

$$\text{constructMat}\left(\frac{1}{i+j}, i, j, 3, 4\right) = \begin{bmatrix} \frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$$

Restituisce una matrice sulla base degli argomenti.

Espr è un'espressione nelle variabili *Var1* e *Var2*. Gli elementi nella matrice risultante sono formati calcolando *Espr* per ciascun valore incrementato di *Var1* e *Var2*.

Var1 è incrementato automaticamente da 1 a *numRighe*. All'interno di ciascuna riga, *Var2* è incrementato da 1 a *numColonne*.

CopyVar (Copia variabile)

CopyVar *Var1*, *Var2*

CopyVar *Var1*,, *Var2*.

CopyVar *Var1*, *Var2* copia il valore della variabile *Var1* nella variabile *Var2*, creando *Var2* se necessario. La variabile *Var1* deve contenere un valore.

Define $a(x)=\frac{1}{x}$	Done
Define $b(x)=x^2$	Done
CopyVar <i>a,c</i> : <i>c</i> (4)	$\frac{1}{4}$
CopyVar <i>b,c</i> : <i>c</i> (4)	16

Se *Var1* è il nome di una funzione esistente definita dall'utente, copia la definizione di quella funzione nella funzione *Var2*. La funzione *Var1* deve essere definita.

CopyVar (Copia variabile)

Catalogo > 

Var1 deve soddisfare i requisiti validi per i nomi di variabile oppure deve essere un'espressione indiretta che viene semplificata in un nome di variabile che soddisfa i suddetti requisiti.

CopyVar *Var1.*, *Var2.* copia tutti i membri del gruppo di variabili *Var1.* nel gruppo *Var2.*, creando *Var2.* se necessario.

Var1. deve essere il nome di un gruppo di variabili esistente, come ad esempio i risultati statistici *stat.nn* o le variabili create utilizzando la funzione **LibShortcut()**. Se *Var2.* esiste già, questo comando sostituisce tutti i termini che sono comuni a entrambi i gruppi e aggiunge i termini che non esistono ancora. Se uno o più termini di *Var2.* sono bloccati, tutti i termini di *Var2.* vengono lasciati invariati

<i>aa.a:=45</i>	45				
<i>aa.b:=6.78</i>	6.78				
CopyVar <i>aa.</i> , <i>bb.</i>	<i>Done</i>				
getVarInfo()	<table border="1"><tr><td><i>aa.a</i> "NUM" "□" 0</td></tr><tr><td><i>aa.b</i> "NUM" "□" 0</td></tr><tr><td><i>bb.a</i> "NUM" "□" 0</td></tr><tr><td><i>bb.b</i> "NUM" "□" 0</td></tr></table>	<i>aa.a</i> "NUM" "□" 0	<i>aa.b</i> "NUM" "□" 0	<i>bb.a</i> "NUM" "□" 0	<i>bb.b</i> "NUM" "□" 0
<i>aa.a</i> "NUM" "□" 0					
<i>aa.b</i> "NUM" "□" 0					
<i>bb.a</i> "NUM" "□" 0					
<i>bb.b</i> "NUM" "□" 0					

corrMat() (Matrice di correlazione)

Catalogo > 

corrMat(*Lista1*,*Lista2*[...,[*Lista20*]])

Calcola la matrice di correlazione per la matrice affiancata [*Lista1* *Lista2* ... *Lista20*].

cos() (Coseno)

Tasto 

cos(*Lista1*) \Rightarrow *lista*

In modalità angolo in gradi:

cos(*Lista1*) restituisce una lista dei coseni di tutti gli elementi di *Lista1*.

In modalità angolo in gradi (gradi centesimali):

Nota: conformemente alla modalità di misurazione degli angoli impostata, l'argomento viene interpretato come angolo in gradi, gradianti o radienti. È possibile utilizzare $^\circ$, G o r per escludere provvisoriamente la modalità d'angolo selezionata.

In modalità angolo in radianti:

cos
(*matriceQuadrata1*) \Rightarrow *matriceQuadrata*

In modalità angolo in radianti:

cos() (Coseno)

Tasto 

Restituisce il coseno della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il coseno di ogni elemento.

Quando una funzione scalare $f(A)$ opera su *matriceQuadrata1* (A), il risultato viene calcolato dall'algoritmo:

$$\cos \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 0.212493 & 0.205064 & 0.121389 \\ 0.160871 & 0.259042 & 0.037126 \\ 0.248079 & -0.090153 & 0.218972 \end{bmatrix}$$

Calcola gli autovalori (λ_i) e gli autovettori (V_i) di A .

matriceQuadrata1 deve essere diagonalizzabile. Inoltre, non può avere variabili simboliche alle quali non sia stato assegnato un valore.

Forma le matrici:

$$B = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \text{ and } X = [V_1, V_2, \dots, V_n]$$

Quindi $A = X B X^{-1}$ e $f(A) = X f(B) X^{-1}$. Ad esempio, $\cos(A) = X \cos(B) X^{-1}$ dove:

$$\cos(B) =$$

$$\begin{bmatrix} \cos(\lambda_1) & 0 & \dots & 0 \\ 0 & \cos(\lambda_2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \cos(\lambda_n) \end{bmatrix}$$

Tutti i calcoli vengono eseguiti in virgola mobile.

cos⁻¹() (Arcocoseno)

Tasto 

$$\cos^{-1}(Lista1) \Rightarrow lista$$

In modalità angolo in gradi:

cos⁻¹(Lista1) restituisce la lista dell'inversa dei coseni di ciascun elemento di *Lista1*.

In modalità angolo in gradi (gradi centesimali):

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradi centesimali o gradi.

In modalità angolo in radianti:

cos⁻¹() (Arcocoseno)

Tasto 

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccos (...)**.

cos⁻¹

(matriceQuadrata1)⇒matriceQuadrata

Restituisce il coseno inverso della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il coseno inverso di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti e in modalità formato rettangolare complesso:

$$\cos^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$$

$$\begin{bmatrix} 1.73485+0.064606 \cdot i & -1.49086+2.10514 \\ -0.725533+1.51594 \cdot i & 0.623491+0.77836 \cdot i \\ -2.08316+2.63205 \cdot i & 1.79018-1.27182 \end{bmatrix}$$

Per vedere l'intero risultato, premere **▲**, quindi utilizzare **◀** e **▶** per spostare il cursore.

cosh() (Coseno iperbolico)

Catalogo > 

cosh(Lista1)⇒lista

In modalità angolo in gradi:

cosh(Lista1) restituisce una lista dei coseni iperbolici di ciascun elemento di *Lista1*.

cosh

(matriceQuadrata1)⇒matriceQuadrata

Restituisce il coseno iperbolico della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il coseno iperbolico di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti:

$$\cosh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$$

$$\begin{bmatrix} 421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628 \end{bmatrix}$$

cosh⁻¹() (Arcocoseno iperbolico)

Catalogo > 

cosh⁻¹(Lista1)⇒lista

cosh⁻¹(Lista1) restituisce una lista dell'inversa dei coseni iperbolici di ciascun elemento di *Lista1*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccosh (...)**.

cosh⁻¹() (Arcocoseno iperbolico)

Catalogo > 

cosh⁻¹

(matriceQuadrata1)⇒matriceQuadrata

Restituisce l'inversa del coseno iperbolico della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare l'inversa del coseno iperbolico di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti e in modalità formato rettangolare complesso:

$$\cosh^{-1} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2.52503+1.73485 \cdot i & -0.009241-1.4908i \\ 0.486969-0.725533 \cdot i & 1.66262+0.62349i \\ -0.322354-2.08316 \cdot i & 1.26707+1.79018i \end{bmatrix}$$

Per vedere l'intero risultato, premere **▲**, quindi utilizzare **◀** e **▶** per spostare il cursore.

cot() (Cotangente)

Tasto 

cot(Lista1)⇒lista

Nota: conformemente alla modalità di misurazione degli angoli impostata, l'argomento viene interpretato come angolo in gradi, gradianti o radianti. È possibile utilizzare $^{\circ}$, $^{\text{G}}$ o $^{\text{r}}$ per escludere provvisoriamente la modalità d'angolo selezionata.

In modalità angolo in gradi:

In modalità angolo in gradianti (gradi centesimali):

In modalità angolo in radianti:

cot⁻¹() (Arcocotangente)

Tasto 

cot⁻¹(Lista1)⇒lista

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradianti o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccot(...)**.

In modalità angolo in gradi:

$$\cot^{-1}(1) \quad 45.$$

In modalità angolo in gradianti (gradi centesimali):

$$\cot^{-1}(1) \quad 50.$$

In modalità angolo in radianti:

coth() (Cotangente iperbolica)

Catalogo > 

coth(Lista1)⇒lista

coth⁻¹(*Lista1*)⇒*lista*

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccoth**(...).

count()**count(*Valore1oLista1* [,*Valore2oLista2* [...]])**⇒*valore*

Restituisce il totale accumulato di tutti gli elementi negli argomenti che danno come risultato valori numerici.

Gli argomenti possono essere un'espressione, un valore, una lista o una matrice. È possibile mischiare tipi di dati e utilizzare argomenti di varie dimensioni.

Per una lista, una matrice o un intervallo di celle, viene calcolato ciascun elemento per determinare se dovrebbe essere incluso nel conteggio.

In Foglio elettronico, è possibile utilizzare un intervallo di celle al posto di qualsiasi argomento.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 222.

countif()**countif(*Lista,Criteri*)**⇒*valore*

Restituisce il totale accumulato di tutti gli elementi di *Lista* che soddisfano i *Criteri* specificati.

Criteri può essere:

- Un valore, un'espressione o una stringa. Ad esempio, **3** conta solo quegli elementi di *Lista* che sono semplificati nel numero 3.
- Un'espressione booleana contenente il simbolo **?** come segnaposto di ciascun

countIf({1,3,"abc",undef,3,1},3) 2

Conta il numero di elementi uguali a 3.

countIf({ "abc", "def", "abc",3 }, "def") 1

Conta il numero di elementi uguali a "def".

countIf({ 1,3,5,7,9 },?<5) 2

Conta 1 e 3.

elemento. Ad esempio, `?<5` conta solo quegli elementi di *Lista* che sono minori di 5.

In Foglio elettronico, è possibile utilizzare un intervallo di celle al posto di *Lista*.

Gli elementi vuoti (nulli) della lista vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 222.

Nota: vedere anche **sumIf()**, pagina 159 e **frequency()**, pagina 58.

`countIf({1,3,5,7,9},2<?<8)`

3

Conta 3, 5 e 7.

`countIf({1,3,5,7,9},?<4 or ?>6)`

4

Conta 1, 3, 7 e 9.

cPolyRoots()

cPolyRoots(*Poli,Var*)⇒*lista*

cPolyRoots(*ListaDiCoeff*)⇒*lista*

La prima sintassi, **cPolyRoots(*Poli,Var*)**, restituisce una lista di radici complesse del polinomio *Poli* in funzione della variabile specificata *Var*.

La seconda sintassi, **cPolyRoots(*ListaDiCoeff*)**, restituisce una lista di radici complesse per i coefficienti di *ListaDiCoeff*.

Nota: vedere anche **polyRoots()**, pagina 118.

crossP() (Prodotto vettoriale)

crossP(*Lista1, Lista2*)⇒*lista*

Restituisce sotto forma di lista il prodotto vettoriale di *Lista1* e *Lista2*.

Lista1 e *Lista2* devono essere uguali, 2 o 3.

crossP(*Vettore1, Vettore2*)⇒*vettore*

Restituisce un vettore riga o colonna (a seconda degli argomenti) corrispondente al prodotto vettoriale di *Vettore1* per *Vettore2*.

`crossP([1 2 3],[4 5 6])` `[-3 6 -3]`

`crossP([1 2],[3 4])` `[0 0 -2]`

crossP() (Prodotto vettoriale)**Catalogo > **

Vettore1 e *Vettore2* devono essere entrambi vettori riga o vettori colonna. Le dimensioni di entrambi devono essere uguali, 2 o 3.

csc() (Cosecante)**Tasto ****csc(Lista1) \Rightarrow lista**

In modalità angolo in gradi:

In modalità angolo in gradi (gradi centesimali):

In modalità angolo in radianti:

csc⁻¹() (Cosecante inversa)**Tasto ****csc⁻¹(Lista1) \Rightarrow lista**

In modalità angolo in gradi:

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradi centesimali o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccsc (...)**.

csc⁻¹(1)

90.

In modalità angolo in gradi (gradi centesimali):

csc⁻¹(1)

100.

In modalità angolo in radianti:

csech (Cosecante iperbolica)**Catalogo > ****csech(Lista1) \Rightarrow lista****csch⁻¹() (Cosecante iperbolica inversa)****Catalogo > ****csch⁻¹(Lista1) \Rightarrow lista**

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arccsch (...)**.

CubicReg *X, Y[, Freq] [, Categoria, Includi]*]

Calcola la regressione polinomiale cubicay = $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
stat.a, stat.b, stat.c, stat.d	Coefficienti di regressione
stat.R ²	Coefficiente di determinazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

cumulativeSum()

Catalogo > 

cumulativeSum(Lista1)⇒*lista*

Restituisce una lista delle somme cumulative degli elementi in *Lista1*, incominciando dall'elemento 1.

cumulativeSum(Matrice1)⇒*matrice*

Restituisce una matrice delle somme cumulative degli elementi di *Matrice1*. Ciascun elemento è la somma cumulativa della colonna, dall'alto al basso.

Un elemento vuoto (nullo) in *Lista1* o *Matrice1* produce un elemento vuoto nella lista o matrice risultante. Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 222.

cumulativeSum({1,2,3,4}) {1,3,6,10}

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
cumulativeSum(<i>m1</i>)	$\begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 12 \end{bmatrix}$

Cycle (Ripeti)

Catalogo > 

Ripeti

Trasferisce il controllo della funzione alla iterazione immediatamente successiva del ciclo corrente (**For**, **While** o **Loop**).

Cycle non è ammesso al di fuori delle tre strutture di ciclo (**For**, **While** o **Loop**).

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Funzione che elenca le somme degli interi da 1 a 100 saltando 50.

Define <i>g()</i> =Func	<i>Done</i>
Local <i>temp,i</i>	
0→ <i>temp</i>	
For <i>i</i> ,1,100,1	
If <i>i</i> =50	
Cycle	
<i>temp</i> + <i>i</i> → <i>temp</i>	
EndFor	
Return <i>temp</i>	
EndFunc	

g() 5000

►Cylind (Forma cilindrica)

Catalogo > 

Vettore ►Cylind

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Cylind.

Visualizza il vettore riga o colonna nella forma cilindrica [r,∠θ, z].

Vettore deve avere esattamente tre elementi. Può essere una riga o una colonna.

D**dbd()**Catalogo > 

dbd(*data1,data2*)⇒*valore*

Restituisce il numero di giorni tra la *data1* e la *data2* usando il metodo di conteggio dei giorni effettivi.

data1 e *data2* possono essere numeri o liste di numeri all'interno di un intervallo di date del calendario normale. Se sia *data1* che *data2* sono liste, esse devono contenere lo stesso numero di elementi.

data1 e *data2* devono essere comprese tra gli anni 1950 e 2049.

È possibile inserire le date in uno dei due formati, che differiscono esclusivamente per la posizione del punto decimale.

MM.GGAA (formata usato generalmente negli Stati Uniti)

GGMM.AA (formato usato generalmente in Europa)

dbd(12.3103,1.0104)	1
dbd(1.0107,6.0107)	151
dbd(3112.03,101.04)	1
dbd(101.07,106.07)	151

►DD (Visualizza angolo decimale)Catalogo > 

Valore1 ►DD⇒*valore*

Lista1 ►DD⇒*lista*

Matrice1 ►DD⇒*matrice*

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>DD.

Restituisce l'equivalente decimale dell'argomento espresso in gradi. L'argomento è un numero, una lista o una matrice interpretata in gradi, radianti o gradi dall'impostazione della modalità Angolo.

In modalità angolo in gradi:

(1.5°)►DD	1.5°
(45°22'14.3")►DD	45.3706°
{(45°22'14.3",60°0'0")}►DD	{45.3706°,60°}

In modalità angolo in gradi (gradi centesimali):

1►DD	90°
	10

In modalità angolo in radianti:

(1.5)►DD

85.9437°

►Decimal (Decimale)

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Decimal.

Visualizza l'argomento nella forma decimale. Questo operatore può essere utilizzato solo alla fine della riga di introduzione.

$\frac{1}{3}$ ►Decimal	0.333333
------------------------	----------

Define (Definisci)

Define *Var* = *Espressione*

Define *Funzione*(*Param1*, *Param2*, ...) = *Espressione*

Definisce la variabile *Var* o la funzione *Funzione* definita dall'utente.

Parametri, quali *Param1*, sono segnaposto per il passaggio di argomenti alla funzione. Quando si chiama una funzione definita dall'utente, occorre fornire argomenti (ad esempio, valori o variabili) corrispondenti ai parametri. Una volta chiamata, la funzione calcola *Espressione* utilizzando gli argomenti forniti.

Var e *Funzione* non possono essere il nome di una variabile di sistema né una funzione o un comando predefiniti.

Nota: questa forma di **Define** equivale all'esecuzione dell'espressione:
espressione → *Funzione*(*Param1*, *Param2*).

Define $g(x,y)=2 \cdot x - 3 \cdot y$	Done
$g(1,2)$	-4
$1 \rightarrow a: 2 \rightarrow b: g(a,b)$	-4
Define $h(x)=\text{when}(x < 2, 2 \cdot x - 3, -2 \cdot x + 3)$	Done
$h(-3)$	-9
$h(4)$	-5

Define (Definisci)

Catalogo > 

Define Funzione(Param1, Param2, ...) =
Func
Blocco
EndFunc

Define Programma(Param1, Param2, ...) = **Prgm**
Blocco
EndPrgm

In questa forma, il programma o la funzione definita dall'utente può eseguire un blocco di istruzioni multiple.

Blocco può essere una singola istruzione o una serie di istruzioni su righe separate. Inoltre **Blocco** può includere espressioni e istruzioni (quali **If**, **Then**, **Else** e **For**).

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Nota: vedere anche **Define LibPriv**, pagina 38 e **Define LibPub**, pagina 39.

Define $g(x,y)$ =Func
If $x > y$ Then
Return x
Else
Return y
EndIf
EndFunc

g(3,-7) 3

Define $g(x,y)$ =Prgm
If $x > y$ Then
Disp x , " greater than ", y
Else
Disp x , " not greater than ", y
EndIf
EndPrgm

Done

g(3,-7) 3 greater than -7

Done

Define LibPriv (Definisci libreria privata)

Catalogo > 

Define LibPriv Var = Espressione

Define LibPriv Funzione(Param1, Param2, ...) = **Espressione**

Define LibPriv Funzione(Param1, Param2, ...) = **Func**
Blocco
EndFunc

Define LibPriv Programma(Param1, Param2, ...) = **Prgm**
Blocco
EndPrgm

Funziona come **Define**, eccetto che definisce una variabile, una funzione o un programma libreria privata. Funzioni e programmi privati non sono elencati nel Catalogo.

Define LibPriv (Definisci libreria privata)

Catalogo > 

Nota: vedere anche **Define**, pagina 37 e
Define LibPub, pagina 39.

Define LibPub (Definisci libreria pubblica)

Catalogo > 

Define LibPub *Var* = *Espressione*

Define LibPub *Funzione*(*Param1*, *Param2*,
...) = *Espressione*

Define LibPub *Funzione*(*Param1*, *Param2*,
...) = **Func**
Blocco
EndFunc

Define LibPub *Programma*(*Param1*,
Param2, ...) = **Prgm**
Blocco
EndPrgm

Funziona come **Define**, eccetto che definisce una variabile, una funzione o un programma libreria pubblica. Funzioni e programmi pubblici vengono elencati nel Catalogo dopo che la libreria è stata salvata e aggiornata.

Nota: vedere anche **Define**, pagina 37 e
Define LibPriv, pagina 38.

deltaList()

Vedere **ΔList()**, pagina 87.

DelVar

Catalog > 

DelVar *Var1*[, *Var2*] [, *Var3*] ...

DelVar *Var*.

Elimina dalla memoria la variabile o il gruppo di variabili specificato.

Se una o più variabili sono bloccate, questo comando visualizza un messaggio di errore ed elimina solo le variabili sbloccate. Vedere **unLock**, pagina 173.

DelVar *Var.* elimina tutti i membri del gruppo di variabili *Var.* (come ad esempio i risultati statistici *stat.nn* o le variabili create utilizzando la funzione **LibShortcut()**). Il punto *(.)* in questa forma del comando **DelVar** ne limita l'applicabilità all'eliminazione di un gruppo di variabili; non può essere applicato alla variabile semplice *Var.*

<i>aa.a</i> :=45	45
<i>aa.b</i> :=5.67	5.67
<i>aa.c</i> :=78.9	78.9
getVarInfo()	$\begin{bmatrix} aa.a & \text{"NUM"} & "45" \\ aa.b & \text{"NUM"} & "5.67" \\ aa.c & \text{"NUM"} & "78.9" \end{bmatrix}$
DelVar <i>aa.</i>	Done
getVarInfo()	"NONE"

delVoid()

delVoid(*Lista1*) \Rightarrow *lista*

Restituisce una lista che ha il contenuto di *Lista1* con tutti gli elementi vuoti (nulli) rimossi.

Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 222.

delVoid({1,void,3})	{1,3}
---------------------	-------

det() (Determinante)

det(*matriceQuadrata*[,
Tolleranza]) \Rightarrow *espressione*

Restituisce il determinante di *matriceQuadrata*.

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tolleranza*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tolleranza* viene ignorato.

- Se si usa **ctrl** **enter** oppure se si imposta la modalità **Auto** o **Approssimato** su Approssimato, i calcoli verranno eseguiti in virgola mobile.
- Se *Tolleranza* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:
5E-14 ·max(dim

det() (Determinante)**Catalogo > **

$$(matriceQuadrata) \cdot \text{rowNorm}$$

$$(matriceQuadrata)$$
diag() (Diagonale)**Catalogo > ****diag(Lista)⇒matrice****diag([2 4 6])**

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$
diag(matriceRiga)⇒matrice**diag(matriceColonna)⇒matrice**

Restituisce una matrice avente i valori dell'argomento lista o matrice nella diagonale principale.

diag(matriceQuadrata)⇒matriceRiga

Restituisce una matrice riga contenente gli elementi della diagonale principale di *matriceQuadrata*.

matriceQuadrata deve essere quadrata.

$$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$$
diag(Ans)

$$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$$
Catalogo > **dim() (Dimensione)****dim(Lista)⇒intero****dim({0,1,2})****3**

Restituisce le dimensioni di *Lista*.

dim(Matrice)⇒lista

$$\dim \begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{pmatrix}$$
{3,2}

Restituisce le dimensioni di Matrice nella forma di una lista a due elementi {righe, colonne}.

dim(Stringa)⇒intero**dim("Hello")****5**

Restituisce il numero di caratteri contenuti nella stringa *Stringa*.

dim("Hello "&"there")**11**

Disp *esprOrString1 [, esprOrString2] ...*

Visualizza gli argomenti nella cronologia di *Calculator*. Gli argomenti possono essere visualizzati in successione, separati da sottili spazi.

Questo comando è utile soprattutto in programmi e funzioni per assicurare la visualizzazione dei calcoli intermedi.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define *chars*(*start,end*)=Prgm

```
For i,start,end
Disp i, " ",char(i)
EndFor
EndPrgm
```

Done

chars(240,243)

240 ð

241 ñ

242 ö

243 ö

Done

DispAt**DispAt** *int,expr1 [,expr2 ...] ...*

DispAt consente di specificare la linea in cui l'espressione o la stringa specificata verrà visualizzata sullo schermo.

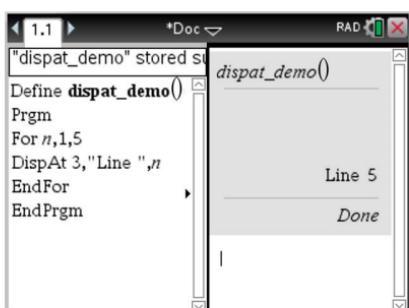
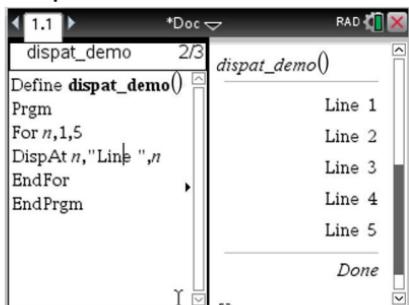
Il numero della linea può essere specificato sotto forma di espressione.

Si tenga presente che il numero della linea non è per lo schermo intero, bensì per l'area immediatamente dopo il comando/programma.

Questo comando consente un output di tipo dashboard da programmi in cui il valore di un'espressione o da una lettura del sensore viene aggiornato sulla stessa linea.

DispAt **Disp** possono essere utilizzati nello stesso programma.

DispAt

Esempio

Nota: il numero massimo è impostato su 8, perché corrisponde a uno schermo - pieno di linee sullo schermo del palmare - se le linee non hanno espressioni matematiche bidimensionali. Il numero esatto di linee dipende dal contenuto delle informazioni visualizzate.

Esempi indicativi:

Define z()=	Chiusura di z()
Prgm	Iterazione 1: Linea 1: N:1 Linea 2: Hello
For n,1,3	
DispAt 1,"N: ",n	
Disp "Hello"	
EndFor	Iterazione 2: Linea 1: N:2 Linea 2: Hello
EndPrgm	Linea 3: Hello
	Iterazione 3: Linea 1: N:3 Linea 2: Hello Linea 3: Hello Linea 4: Hello
Define z1()=	z1()
Prgm	Linea 1: N:3
For n,1,3	Linea 2: Hello
DispAt 1,"N: ",n	Linea 3: Hello
EndFor	Linea 4: Hello
For n,1,4	Linea 5: Hello
Disp "Hello"	
EndFor	
EndPrgm	

Condizioni di errore:

Messaggio di errore	Descrizione
Il numero della linea DispAt deve essere compreso tra 1 e 8	L'espressione valuta il numero della linea al di fuori dell'intervallo 1-8 (inclusi)
Argomenti mancanti	Nella funzione o nel comando mancano uno o più argomenti.
Nessun argomento	Uguale alla finestra di dialogo 'errore di sintassi' corrente
Troppi argomenti	Limitare il numero di argomenti. Stesso errore di Disp.

Messaggio di errore	Descrizione
Tipi di dati non valido	Il primo argomento deve essere un numero.
Nullo: DispAt nullo	L'errore nel tipo di dati "Hello World" viene generato a vuoto (se viene definita la richiamata)

►DMS (Gradi/primi/secondi)

Catalogo > 

Lista ►DMS

In modalità angolo in gradi:

Matrice ►DMS

$\begin{pmatrix} 45.371 \end{pmatrix} \blacktriangleright \text{DMS}$ $45^{\circ}22'15.6''$
 $\begin{pmatrix} \{45.371, 60\} \end{pmatrix} \blacktriangleright \text{DMS}$ $\{45^{\circ}22'15.6'', 60^{\circ}\}$

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @►DMS.

Interpreta l'argomento come un angolo e visualizza il numero DMS equivalente (GGGGGG°PP'SS.ss"). Per ulteriori informazioni sul formato DMS (gradi, primi, secondi) vedere °, ', " pagina 200.

Nota: ► quando DMS viene utilizzato in modalità angolo in radianti, converte i radianti in gradi. Se i dati inseriti sono seguiti dal simbolo dei gradi °, non verrà eseguita alcuna conversione. ►DMS può essere utilizzato solo alla fine di una riga di introduzione.

dotP() (Prodotto scalare)

Catalogo > 

dotP(Lista1, Lista2)⇒espressione

Restituisce il prodotto scalare di due liste.

dotP(Vettore1, Vettore2)⇒espressione

Restituisce il prodotto scalare di due vettori.

Entrambi devono essere vettori riga o colonna.

e^() (Funzione esponenziale)Tasto 

Nota: vedere anche **e** modello di funzione esponenziale, pagina 2.

Nota: premere  per visualizzare $e^()$ è diverso da accedere al carattere **E** dalla tastiera.

Un numero complesso può essere inserito nella forma polare $re^{i\theta}$. Usare questa forma solo nella modalità di misurazione degli angoli in radianti; nella modalità in gradi o gradi/centi causa un errore del dominio.

e^(*Lista1*) \Rightarrow *lista*

Restituisce **e** elevato alla potenza di ciascun elemento di *Lista1*.

e^
(*matriceQuadrata1*) \Rightarrow *matriceQuadrata*

Restituisce l'esponenziale della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare **e** elevato alla potenza di ciascun elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

$$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$$

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

eff()Catalogo > 

eff(*tassoNominale*, *CpY*) \Rightarrow *valore*

eff(5.75,12)

5.90398

Funzione finanziaria che converte il tasso di interesse nominale *tassoNominale* in un tasso effettivo annuo, essendo *CpY* il numero di periodi di capitalizzazione per anno.

tassoNominale deve essere un numero reale e *CpY* deve essere un numero reale > 0 .

Nota: vedere anche **nom()**, pagina 108.

eigVc() (Autowettore)

Catalogo > 

eigVc(*matriceQuadrata*)⇒*matrice*

Restituisce una matrice contenente gli autovettori per una *matriceQuadrata* reale o complessa, in cui ogni colonna del risultato corrisponde ad un autovalore. Tenere presente che un autovettore non è univoco; esso infatti può essere scalato per qualsiasi fattore costante. Gli autovettori vengono normalizzati, cioè se $V = [x_1, x_2, \dots, x_n]$, allora:

$$x_1^2 + x_2^2 + \dots + x_n^2 = 1$$

matriceQuadrata viene dapprima equilibrata con similitudini fino a quando le norme di riga e colonna sono il più vicino possibili allo stesso valore. *matriceQuadrata* viene quindi ridotta nella forma superiore di Hessenberg mentre gli autovettori vengono calcolati con una scomposizione in fattori di Schur.

eigVl() (Autovalore)

Catalogo > 

eigVl(*matriceQuadrata*)⇒*lista*

Restituisce la lista degli autovalori di una *matriceQuadrata* reale o complessa.

matriceQuadrata viene dapprima equilibrata con similitudini fino a quando le norme di riga e colonna sono il più vicino possibili allo stesso valore.

matriceQuadrata viene quindi ridotta nella forma superiore di Hessenberg mentre gli autovalori vengono calcolati dalla matrice superiore di Hessenberg.

In modalità formato rettangolare complesso:

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$$

eigVl(*m1*)

$$\begin{bmatrix} -0.800906 & 0.767947 \\ 0.484029 & 0.573804 + 0.052258 \cdot i \\ 0.352512 & 0.262687 + 0.096286 \cdot i \end{bmatrix} \quad 0.5738 \rightarrow$$

Per vedere l'intero risultato, premere ▲, quindi utilizzare ▲ e ▶ per spostare il cursore.

Else

Vedere If, pagina 72.

Elsef

```
If Espressione booleana1 Then
  Blocco1
ElseIf Espressione booleana2 Then
  Blocco2
:
ElseIf Espressione booleanaN Then
  BloccoN
EndIf
:
```

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

```
Define g(x)=Func
  If x≤-5 Then
    Return 5
  ElseIf x>-5 and x<0 Then
    Return -x
  ElseIf x≥0 and x≠10 Then
    Return x
  ElseIf x=10 Then
    Return 3
  EndIf
EndFunc
```

*Done***EndFor****Vedere For, pagina 56.****EndFunc****Vedere Func, pagina 60.****EndIf****Vedere If, pagina 72.****EndLoop****Vedere Loop, pagina 94.****EndPrgm****Vedere Prgm, pagina 120.****EndTry****Vedere Try, pagina 166.****EndWhile****Vedere While, pagina 177.**

euler(*Espr*, *Var*, *varDipendente*, {*Var0*, *VarMax*}, *varDipendente0*, *incrVar* [, *incrEulero*]) \Rightarrow matrice

euler(*SistemaDiEspr*, *Var*, *ListaDiVarDipendenti*, {*Var0*, *VarMax*}, *ListaDiVarDipendenti0*, *incrVar* [, *incrEulero*]) \Rightarrow matrice

euler(*ListaDiEspr*, *Var*, *ListaDiVarDipendenti*, {*Var0*, *VarMax*}, *ListaDiVarDipendenti0*, *incrVar* [, *incrEulero*]) \Rightarrow matrice

Utilizza il metodo di Eulero per risolvere il sistema

$$\frac{d \text{ depVar}}{d \text{ Var}} = \text{Expr}(\text{Var}, \text{depVar})$$

con *varDipendente* (*Var0*)=*varDipendente0* nell'intervallo [*Var0*,*VarMax*]. Restituisce una matrice la cui prima riga definisce i valori ottenuti di *Var* e la cui seconda riga definisce il valore del primo componente della soluzione per i valori di *Var* corrispondenti, e così via.

Espr è il secondo membro che definisce l'equazione differenziale ordinaria (ODE).

SistemaDiEspr è il sistema di secondi membri che definiscono il sistema di ODE (corrisponde all'ordine di variabili dipendenti in *ListaDiVarDipendenti*).

ListaDiEspr è una lista di secondi membri che definisce il sistema di ODE (corrisponde all'ordine di variabili dipendenti in *ListaDiVarDipendenti*).

Var è la variabile indipendente.

ListaDiVarDipendenti è una lista di variabili dipendenti.

{*Var0*, *VarMax*} è una lista a due elementi che indica la funzione di integrare da *Var0* a *VarMax*.

Equazione differenziale:

$$y' = 0.001 * y * (100 - y) \text{ e } y(0) = 10$$

$$\begin{aligned} \text{euler}\{0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\} \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9 & 11.8712 & 12.9174 & 14.042 \end{bmatrix} \end{aligned}$$

Per vedere l'intero risultato, premere \blacktriangle , quindi utilizzare \blacktriangleleft e \triangleright per spostare il cursore.

Sistema di equazioni

$$\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

$$\text{con } y1(0) = 2 \text{ e } y2(0) = 5$$

$$\begin{aligned} \text{euler}\left\{\begin{array}{l} -y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{array}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right\} \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. & 5. \\ 2. & 1. & 1. & 3. & 27. & 243. \\ 5. & 10. & 30. & 90. & 90. & -2070. \end{bmatrix} \end{aligned}$$

ListaDiVarDipendenti0 è una lista di valori iniziali di variabili dipendenti.

incrVar è un numero diverso da zero tale che **sign(incrVar) = sign (VarMax-Var0)** e sono restituite soluzioni a $Var0+i \cdot incrVar$ per tutti i valori di $i=0,1,2,\dots$ tali che $Var0+i \cdot incrVar$ sia in $[var0,VarMax]$ (potrebbe non esserci un valore di soluzione a *VarMax*).

incrEulero è un numero intero positivo (predefinito a 1) che definisce il numero di incrementi di Eulero tra i valori ottenuti. La dimensione effettiva dell'incremento utilizzato dal metodo di Eulero è *incrVar/incrEulero*.

eval ()

Menu Hub

eval(*Espr*) \Rightarrow *stringa*

eval() è valido solo in TI-Innovator™ Hub. Argomento del comando delle istruzioni di programmazione **Get**, **GetStr** e **Send**. Il software calcola l'espressione *Espr* e sostituisce l'istruzione **eval()** con il risultato come stringa di caratteri.

L'argomento *Espr* deve essere semplificato in un numero reale.

Impostare l'elemento blu del LED RGB su metà intensità.

<i>lum:=127</i>	127
Send "SET COLOR.BLUE eval(lum)"	
Done	

Reimpostare l'elemento blu su OFF.

Send "SET COLOR.BLUE OFF"	Done
---------------------------	------

L'argomento **eval()** deve essere semplificato in un numero reale.

Send "SET LED eval("4") TO ON"
"Error: Invalid data type"

Programmare per aumentare gradualmente l'elemento rosso

Define fadein() =
Prgm
For <i>i</i> ,0,255,10
Send "SET COLOR.RED eval(<i>i</i>)"
Wait 0.1
EndFor
Send "SET COLOR.RED OFF"
EndPrgm

Eseguire il programma.

fadein()	Done
<i>n:=0.25</i>	0.25
<i>m:=8</i>	8
<i>n·m</i>	2.
Send "SET COLOR.BLUE ON TIME eval(n·m)"	Done
<i>iostr.SendAns "SET COLOR.BLUE ON TIME 2"</i>	

Anche se **eval()** non mostra il suo risultato, è possibile visualizzare la stringa del comando Hub risultante dopo l'esecuzione del comando controllando ciascuna delle seguenti variabili speciali.

iostr.SendAns
iostr.GetAns
iostr.GetStrAns

Nota: vedere anche **Get** (pagina 62), **GetStr** (pagina 69) e **Send** (pagina 141).

Exit (Esci)

Catalogo > 

Exit

Permette di uscire dal blocco corrente **For**, **While** o **Loop**.

Exit non è ammesso al di fuori delle tre strutture iterative (**For**, **While** o **Loop**).

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Descrizione della funzione:

Define *g()*=Func Done
 Local *temp,i*
 $0 \rightarrow temp$
 For *i,1,100,1*
 $temp+i \rightarrow temp$
 If *temp>20* Then
 Exit
 EndIf
 EndFor
 EndFunc

g() 21

exp() (e alla potenza)

Tasto 

Restituisce *e* elevato alla potenza di *Espr1*.

Restituisce *e* elevato alla potenza di *Valore1*.

Nota: vedere anche *e* modello di funzione esponenziale, pagina 2.

exp() (e alla potenza)

Tasto 

Un numero complesso può essere inserito nella forma polare $re^{i\theta}$. Usare questa forma solo nella modalità di misurazione degli angoli in radianti; nella modalità in gradi o gradi/centri causa un errore del dominio.

exp(Lista1)⇒lista

Restituisce e elevato alla potenza di ciascun elemento di *Lista1*.

exp

(matriceQuadrata1)⇒matriceQuadrata

Restituisce l'esponenziale della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare e elevato alla potenza di ciascun elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$	$\begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$
--	---

expr() (Da stringa a espressione)

Catalogo > 

expr(Stringa)⇒espressione

Restituisce la stringa di caratteri contenuta in *Stringa* come espressione e la esegue subito.

ExpReg (Regressione esponenziale)

Catalogo > 

ExpReg X, Y [, [Freq][, Categoria, Includi]]

Calcola la regressione esponenziale $y = a \cdot (b)^x$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot (b)^x$
stat.a, stat.b	Coefficienti di regressione
stat.r ²	Coefficiente di determinazione lineare di dati trasformati
stat.r	Coefficiente di correlazione per dati trasformati (<i>x</i> , $\ln(y)$)
stat.Resid	Residui associati al modello esponenziale
stat.ResidTrans	Residui associati all'adattamento lineare dei dati trasformati
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

factor() (Fattorizza)**Catalogo > **

factor(*numeroRazionale*) restituisce il numero razionale scomposto in fattori primi. Per i numeri composti, il tempo di elaborazione cresce in modo esponenziale secondo il numero di cifre del secondo fattore più grande. Ad esempio, la scomposizione in fattori di un intero di 30 cifre può richiedere più di un giorno, mentre la scomposizione di un numero di 100 cifre può richiedere più di un secolo.

<code>factor(152417172689)</code>	123457·1234577
<code>isPrime(152417172689)</code>	false

Per arrestare manualmente un calcolo:

- **Palmare:** Tenere premuto il tasto  **on** e premere **enter** più volte.
- **Windows®:** Tenere premuto il tasto **F12** e premere **Invio** più volte.
- **Macintosh®:** Tenere premuto il tasto **F5** e premere **Invio** più volte.
- **iPad®:** L'app mostra un prompt. È possibile continuare ad attendere o annullare.

Se si desidera soltanto determinare se un numero è primo, utilizzare **isPrime()**. Ciò risulta molto più veloce, in particolare se *numeroRazionale* non è primo e se il secondo fattore più grande ha più di cinque cifre.

FCdf() (Probabilità di distribuzione F)**Catalogo > **

FCdf
(
estremoInf,
estremoSup,*glNumer*,*glDenom*) \Rightarrow *numero*
se *estremoInf* e *estremoSup* sono numeri,
lista se *estremoInf* e *estremoSup* sono liste

FCdf
(
estremoInf,
estremoSup,*glNumer*,*glDenom*) \Rightarrow *numero*

FCdf() (Probabilità di distribuzione F)

Catalogo > 

se *estremoInf* e *estremoSup* sono numeri,
lista se *estremoInf* e *estremoSup* sono liste

Calcola la probabilità di distribuzione F tra
lowBound e *upBound* per il *dfNumero*
specificato (gradi di libertà) e *dfDenom*.

Per $P(X \leq upBound)$, impostare *lowBound* = 0.

Fill (Riempì)

Catalogo > 

varMatrice deve esistere già.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow amatrix$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Fill 1.01, <i>amatrix</i>	Done

Fill 1.01, *amatrix*

amatrix $\begin{bmatrix} 1.01 & 1.01 \\ 1.01 & 1.01 \end{bmatrix}$

varLista deve esistere già.

$\{1,2,3,4,5\} \rightarrow alist$	$\{1,2,3,4,5\}$
Fill 1.01, <i>alist</i>	Done

Fill 1.01, *alist*

alist $\{1.01,1.01,1.01,1.01,1.01\}$

FiveNumSummary

Catalog > 

FiveNumSummary *X*[, *Freq*]
[, *Categoria*, *Includi*]]

Fornisce una versione abbreviata delle statistiche a 1 variabile nella lista *X*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

X rappresenta una lista contenente i dati.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici di categoria dei dati corrispondenti di *X*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Un elemento vuoto (nullo) in qualsiasi lista *X*, *Freq* o *Categoria* produce un elemento vuoto corrispondente in tutte queste liste. Per ulteriori informazioni sugli elementi vuoti, vedere pagina 222

Variabile di output	Descrizione
stat.MinX	Minimo dei valori x
stat.Q ₁ X	1° quartile di x
stat.MedianX	Mediana di x
stat.Q ₃ X	3° quartile di x
stat.MaxX	Massimo dei valori x

floor() (Arrotondato per difetto)

Catalogo >

Restituisce il numero intero più grande che è \leq all'argomento. Questa funzione è identica a *int()*.

`floor(-2.14)`

-3.

L'argomento può essere un numero reale o complesso.

floor(ListaI)⇒lista

`floor({3/2, 0, -5.3})`

{1, 0, -6.}

floor(MatriceI)⇒matrice

`floor[[1.2 3.4], [2.5 4.8]]`

[1. 3.
2. 4.]

Restituisce una lista o matrice del valore arrotondato per difetto di ciascun elemento.

Nota: vedere anche *ceiling()* e *int()*.

For *Var, Basso, Alto [, Incr]*

Blocco

EndFor

Esegue iterativamente le istruzioni di *Blocco* per ciascun valore di *Var*, da *Basso* a *Alto*, secondo incrementi pari a *Incr*.

Var non deve essere una variabile di sistema.

Incr può essere un valore positivo o negativo. Il valore predefinito è 1.

Blocco può essere una singola istruzione o una serie di istruzioni separate dal carattere ":".

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define *g()*=Func

Done

Local *tempsum,step,i*

0→*tempsum*

1→*step*

For *i,1,100,step*

tempsum+i→*tempsum*

EndFor

EndFunc

g()

5050

format() (Formato)

Catalogo >

stringaFormato è una stringa e deve essere espressa nella forma: "F[n]", "S[n]", "E[n]", "G[n][c]", dove le porzioni racchiuse tra parentesi [] sono facoltative.

F[n]: formato fisso. n rappresenta il numero di cifre da visualizzare dopo la virgola decimale.

S[n]: formato scientifico. n rappresenta il numero di cifre da visualizzare dopo la virgola decimale.

E[n]: formato tecnico. n rappresenta il numero di cifre dopo la prima cifra significativa. L'esponente è modificato secondo multipli di tre e la virgola decimale viene spostata verso destra di zero, una o due cifre.

format(1.234567,"B")	"1.235"
format(1.234567,"s2")	"1.23e0"
format(1.234567,"e3")	"1.235e0"
format(1.234567,"g3")	"1.235"
format(1234.567,"g3")	"1,234.567"
format(1.234567,"g3,r:")	"1:235"

G[n][c]: analogo al formato fisso, separa inoltre le cifre a sinistra del separatore decimal in gruppi di tre. c specifica il carattere separatore dei gruppi; il valore predefinito è la virgola. Se c è un punto, il separatore decimale viene visualizzato come virgola.

[Rc]: tutti gli indicatori precedenti possono essere seguiti dal suffisso di radice Rc, dove c è un singolo carattere che specifica che cosa sostituire al punto della radice.

fPart() Funzione parte frazionaria

fPart(*Espr1*)⇒espressione

fPart(-1.234)	-0.234
---------------	--------

fPart(*Lista1*)⇒lista

fPart({1, -2.3, 7.003})	{0, -0.3, 0.003}
-------------------------	------------------

fPart(*Matrice1*)⇒matrice

Restituisce la parte frazionaria dell'argomento.

Per una lista o una matrice, restituisce le parti frazionarie degli elementi.

L'argomento può essere un numero reale o complesso.

FPdf() (Probabilità di distribuzione F)

FPdf(*valX,glNumer,glDenom*)⇒numero se *ValX* è un numero, lista se *ValX* è una lista

FPdf(*valX,glNumer,glDenom*)⇒numero se *ValX* è un numero, lista se *ValX* è una lista

Calcola la probabilità di distribuzione F in *valX* per il *glNumer* (gradi di libertà) e *glDenom* specificati.

freqTable►list()

Catalogo >

**freqTable►list(Lista1,Lista
ListaInteriFreq)⇒lista**

Restituisce una lista contenente gli elementi di *Lista1* espansi secondo le frequenze in *ListaInteriFreq*. Questa funzione può essere utilizzata per costruire una tabella di frequenze per l'applicazione Dati e statistiche.

Lista1 può essere qualsiasi lista valida.

ListaInteriFreq deve avere la stessa dimensione di *Lista1* e deve contenere solo elementi interi non negativi. Ciascun elemento specifica il numero di volte che l'elemento corrispondente di *Lista1* verrà ripetuto nella lista dei risultati. Un valore zero esclude l'elemento corrispondente di *Lista1*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **freqTable@>list(...)**.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

frequency()

Catalogo >

**frequency
(Lista1,listaContenitori)⇒lista**

Restituisce una lista contenente i conteggi degli elementi di *Lista1*. I conteggi sono basati su intervalli (contenitori) definiti nell'argomento *listaContenitori*.

Se *listaContenitori* è {b(1), b(2), ..., b(n)}, gli intervalli specificati sono {?≤b(1), b(1)<?≤b(2), ..., b(n-1)<?≤b(n), b(n)>?}. La lista risultante è un elemento più lungo di *listaContenitori*.

freqTable►list({1,2,3,4},{1,4,3,1})
{1,2,2,2,2,3,3,3,4}
freqTable►list({1,2,3,4},{1,4,0,1})
{1,2,2,2,2,4}

datalist:={1,2,e,3,π,4,5,6,"hello",7}
{1,2,2.71828,3,3.14159,4,5,6,"hello",7}
frequency{datalist,{2.5,4.5}} {2,4,3}

Spiegazione del risultato:

2 elementi di *Datalist* sono ≤2.5

4 elementi di *Datalist* sono >2.5 e ≤4.5

3 elementi di *Datalist* sono >4.5

L'elemento "hello" è una stringa e non può essere collocata in alcun contenitore definito.

Ciascun elemento del risultato corrisponde al numero di elementi di *Lista1* che rientrano nell'intervallo di quel contenitore. Espresso nei termini della funzione **countIf()**, il risultato è {
countIf(list, ?≤b(1)), countIf(list, b(1)<?≤b(2)), ..., countIf(list, b(n-1)<?≤b(n)), countIf(list, b(n)>?) }.

Gli elementi di *Lista1* che non possono essere “inseriti in un contenitore” vengono ignorati. Gli elementi vuoti (nulli) vengono ugualmente ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

In Foglio elettronico, è possibile utilizzare un intervallo di celle al posto di entrambi gli argomenti.

Nota: vedere anche **countIf()**, pagina 31.

FTest_2Samp (Verifica F su due campioni)

FTest_2Samp *Lista1,Lista2[,Freq1[,Freq2[,Ipotesi]]]*

FTest_2Samp *Lista1,Lista2[,Freq1[,Freq2[,Ipotesi]]]*

(Input lista dati)

FTest_2Samp *sx1,n1,sx2,n2[,Ipotesi]*

FTest_2Samp *sx1,n1,sx2,n2[,Ipotesi]*

(Input statistiche riepilogo)

Consente di eseguire una verifica F su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Per $H_a: \sigma_1 > \sigma_2$, impostare *Ipotesi*>0

Per $H_a: \sigma_1 \neq \sigma_2$ (default), impostare *Ipotesi* =0

Per $H_a: \sigma_1 < \sigma_2$, impostare *Ipotesi*<0

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.F	Statistica F calcolata per la sequenza di dati
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.dfNumer	Gradi di libertà del numeratore = $n_1 - 1$
stat.dfDenom	Gradi di libertà del denominatore = $n_2 - 1$
stat.sx1, stat.sx2	Deviazioni standard dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</i>
stat.x1_bar	Medie dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</i>
stat.x2_bar	
stat.n1, stat.n2	Dimensione dei campioni

Func*Blocco***EndFunc**

Modello per la creazione di una funzione definita dall'utente.

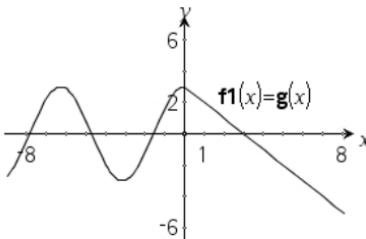
Blocco può essere una singola istruzione, una serie di istruzioni separate dal carattere ":" o una serie di istruzioni su righe separate. La funzione può utilizzare l'istruzione **Return** per restituire un dato risultato.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Definizione di una funzione Piecewise (definita a tratti):

```
Define g(x)=Func           Done
  If x<0 Then
    Return 3·cos(x)
  Else
    Return 3-x
  EndIf
EndFunc
```

Risultato della rappresentazione grafica $g(x)$



gcd() (Massimo comun divisore)[Catalogo > !\[\]\(c0f333d6e4938e0decc679cb8884fdc5_img.jpg\)](#)**gcd**(*Numero1*, *Numero2*) \Rightarrow espressione**gcd**[18,33]

3

Restituisce il massimo comune divisore (gcd) dei due argomenti. Il gcd di due frazioni è il **massimo comune divisore** dei rispettivi numeratori diviso per il **minimo comune multiplo (lcm)** dei loro denominatori.

In modalità Auto o Approssimato, il gcd di numeri decimali in virgola mobile è 1.0.

gcd(*Lista1*, *Lista2*) \Rightarrow lista**gcd**{12,14,16}, {9,7,5} {3,7,1}

Restituisce i massimi comuni divisori degli elementi corrispondenti in *Lista1* e *Lista2*.

gcd(*Matrice1*, *Matrice2*) \Rightarrow matrice**gcd** $\left[\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}, \begin{bmatrix} 4 & 8 \\ 12 & 16 \end{bmatrix}\right]$ $\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$

Restituisce i massimi comuni divisori degli elementi corrispondenti in *Matrice1* e *Matrice2*.

geomCdf() (Probabilità geometrica cumulata)[Catalogo > !\[\]\(96596a1f8856aad8594d0fe233391f2d_img.jpg\)](#)**geomCdf**

()

p

,*valoreInferiore*,*valoreSuperiore*) \Rightarrow numero se *valoreInferiore* e *valoreSuperiore* sono numeri, lista se *valoreInferiore* e *valoreSuperiore* sono liste

geomCdf(*p*,*valoreSuperiore*) per $P(1 \leq X \leq \text{valoreSuperiore}) \Rightarrow$ numero se *valoreSuperiore* è un numero, lista se *valoreSuperiore* è una lista

Calcola una probabilità geometrica cumulata da *valoreInferiore* a *valoreSuperiore* con la probabilità di esiti favorevoli *p* specificata.

Per $P(X \leq \text{valoreSuperiore})$, impostare *valoreSuperiore* = 1.

geomPdf(*p,ValX*)⇒numero se *ValX* è un numero, lista se *ValX* è una lista

Calcola una probabilità in corrispondenza di *ValX*, il numero della prova in cui si è verificato il primo caso favorevole, per la distribuzione geometrica discreta con la probabilità di esiti favorevoli *p* specificata.

Get**Menu Hub**

Get[*promptString,]var[, statusVar*]

Get[*promptString,] funz(*arg1, ...argn*)*
[, statusVar]

Istruzione di programmazione: recupera un valore da uno collegato TI-Innovator™ Hub e assegna il valore alla variabile *var*.

Il valore deve essere obbligatorio:

- In anticipo, attraverso un comando **Send "LEGGI ..."**.
 - oppure —
- Incorporando una richiesta "LEGGI ..." come argomento *promptString* opzionale. Questo metodo consente di utilizzare un solo comando per richiedere il valore e recuperarlo.

Avviene una semplificazione隐式. Per esempio, una stringa ricevuta di "123" viene interpretata come valore numerico. Per preservare la stringa, utilizzare **GetStr** invece di **Get**.

Se si include l'argomento opzionale *statusVar*, viene assegnato un valore basato sul successo dell'operazione. Un valore di zero significa che non è stato ricevuto nessun dato.

Esempio: richiedere il valore attuale del sensore di livello luminosità integrato dell'hub. Utilizzare **Get** per recuperare il valore e assegnarlo alla variabile *lightval*.

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	0.347922

Incorporare la richiesta LEGGI all'interno del comando **Get**.

Get "READ BRIGHTNESS", <i>lightval</i>	Done
<i>lightval</i>	0.378441

Nella seconda sintassi, l'argomento *funz*() consente a un programma di memorizzare la stringa ricevuta come definizione di una funzione. Questa sintassi opera come se il programma avesse eseguito il comando:

Definire *funz(arg1, ...argn) = stringa ricevuta*

Il programma può quindi usare la funzione definita *funz()*.

Nota: è possibile utilizzare il comando **Get** all'interno di un programma definito dall'utente, ma non di una funzione.

Nota: vedere anche **GetStr**, pagina 69 e **Send**, pagina 141.

getDenom() (Ottieni/restituisci denominatore)

Catalogo >

Trasforma l'argomento in un'espressione con denominatore comune ridotto e successivamente restituisce il denominatore.

getKey()

Catalogo >

getKey([0|1]) ⇒ **returnString**

Descrizione: **getKey()** - consente al programma TI-Basic di ottenere l'input della tastiera - palmare, desktop ed emulatore su desktop.

Esempio:

- **keypressed := getKey()** restituirà un codice o una stringa vuota, se non è stato premuto alcun tasto.
Questa chiamata verrà restituita immediatamente.
- **keypressed := getKey(1)** attenderà finché non verrà premuto un tasto.
Questa chiamata metterà in pausa l'esecuzione del programma finché non verrà premuto un tasto.

getKey()

Esempio:

```

1.1 1.2 *Doc ▾
"getkey_demo" stored s
Define getKey_demo()
Prgm
Local key
key:="1"
While key≠"esc"
key:=getKey(1)
Disp "Key: ",key
EndWhile
EndPrgm

```

Key: 1
Key: A
Key: =
Key: ^
Key: square
Key: var
Key: esc

Done

Gestione delle battute di tasto:

Tasto palmare/emulatore	Desktop	Restituire un valore
Esc	Esc	"esc"
Touchpad - Clic in alto	n/d	"up"
Acceso	n/d	"home"
Scratchapp	n/d	"scratchpad"
Touchpad - Clic a sinistra	n/d	"left"
Touchpad - Clic al centro	n/d	"center"
Touchpad - Clic a destra	n/d	"right"
Doc	n/d	"doc"
Tabulazione	Tabulazione	"tab"
Touchpad - Clic in basso	Freccia giù	"down"
Menu	n/d	"menu"
Ctrl	Ctrl	nessun valore restituito
Maiusc	Maiusc	nessun valore restituito
Var	n/d	"var"
Canc	n/d	"del"
=	=	"="
trig	n/d	"trig"
da 0 a 9	0-9	"0" ... "9"
Modelli	n/d	"template"
Catalogo	n/d	"cat"
^	^	"^"
X^2	n/d	"square"
/ (tasto per la divisione)	/	"/"
* (tasto per la moltiplicazione)	*	"*"

Tasto palmare/emulatore	Desktop	Restituire un valore
e^x	n/d	"exp"
10^x	n/d	"10power"
+	+	"+"
-	-	"_"
(("("
))	")"
.	.	"."
(-)	n/d	"-" (segno negativo)
Invio	Invio	"enter"
ee	n/d	"E" (notazione scientifica E)
a - z	a-z	alpha = lettera premuta (minuscola) ("a" - "z")
Maiusc a-z	Maiusc a-z	alpha = lettera premuta "A" - "Z"
		Nota: Ctrl-Maiusc attiva il blocco delle maiuscole
?!	n/d	"?!"
pi	n/d	"pi"
Contrassegno	n/d	nessun valore restituito
,	,	","
Return	n/d	"return"
spazio	spazio	" " (spazio)
Inaccessibile	Tasti con caratteri speciali @,!,&, ecc.	Il carattere viene restituito
n/d	Tasti funzione	Nessun carattere restituito
n/d	Tasti speciali di controllo desktop	Nessun carattere restituito
Inaccessibile	Non sono disponibili altri tasti desktop sulla	Lo stesso carattere restituito in Note (non in riquadro matematica)

Tasto palmare/emulatore	Desktop calcolatrice, mentre getKey() attende una battuta. ({, }, ;, :, ...)	Restituire un valore
--------------------------------	--	-----------------------------

Nota: è importante notare che la presenza di `getKey()` in un programma cambia il modo in cui certi eventi vengono gestiti dal sistema. Alcuni di questi sono descritti qui di seguito.

Terminare il programma e l'evento Handle - Proprio come se l'utente dovesse uscire dal programma premendo il tasto **ON**

"**Support**" qui sotto significa - il sistema funziona come previsto - l'esecuzione del programma continua.

Evento	Dispositivo	Desktop - Software TI-Nspire™ per studenti
Test rapido	Terminare il programma, l'evento handle	Come per il palmare (solo TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)
Gestione remota dei file (incl. inviare il file 'Exit Press 2 Test' da un altro palmare o desktop-palmare)	Terminare il programma, l'evento handle	Come per il palmare. (solo TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)
Fine classe	Terminare il programma, l'evento handle	Servizio di assistenza (solo TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)

Evento	Dispositivo	Desktop - Tutte le versioni di TI-Nspire™
TI-Innovator™ Hub connettere/disconnettere	Supporto - Può inviare comandi a TI-Innovator™ Hub. Dopo la disconnessione dal programma, TI-Innovator™ Hub continua a funzionare con il palmare.	Come per il palmare

getLangInfo() (Ottieni informazioni sulla lingua)

Catalogo > 

getLangInfo()=>stringa

Restituisce una stringa che corrisponde all'abbreviazione della lingua attiva corrente. Può essere utilizzato, ad esempio, in un programma o in una funzione per determinare la lingua corrente.

Inglese = "en"

Danese = "da"

Tedesco = "de"

Finlandese = "fi"

Francese = "fr"

Italiano = "it"

Olandese = "nl"

Olandese - Belgio = "nl_BE"

Norvegese = "no"

Portoghese = "pt"

Spagnolo = "es"

Svedese = "sv"

getLangInfo()

"en"

getLockInfo()

Catalogo > 

getLockInfo(*Var*)=>valore

Restituisce lo stato bloccato/sbloccato corrente della variabile *Var*.

valore =0: *Var* è sbloccata o non esiste.

valore =1: *Var* è bloccata e non può essere modificata o eliminata.

Vedere **Lock**, pagina 90 e **unLock**, pagina 173.

a:=65

65

Lock *a*

Done

getLockInfo(*a*)

1

a:=75 "Error: Variable is locked."

DelVar *a*

"Error: Variable is locked."

Unlock *a*

Done

a:=75

75

DelVar *a*

Done

getMode()

Catalogo > 

getMode(*interoNomeModo*)=>valore

getMode(0)⇒lista

getMode(interoNomeModo) restituisce un valore che rappresenta l'impostazione corrente del modo *internoNomeModo*.

getMode(0) restituisce una lista contenente coppie di numeri. Ciascuna coppia è costituita da un numero intero per il modo e da un numero intero per l'impostazione.

Per un elenco dei modi e delle relative impostazioni, vedere la tabella seguente.

Se si salvano le impostazioni con **getMode(0) → var**, è possibile utilizzare **setMode(var)** in una funzione o in un programma per ripristinare temporaneamente le impostazioni solo all'interno dell'esecuzione della funzione o del programma. Vedere **setMode()**, pagina 144.

Nome modo	Intero modo	Intere impostazioni
Mostra cifre (Display digits)	1	1 =Mobile, 2 =Mobile1, 3 =Mobile2, 4 =Mobile3, 5 =Mobile4, 6 =Mobile5, 7 =Mobile6, 8 =Mobile7, 9 =Mobile8, 10 =Mobile9, 11 =Mobile10, 12 =Mobile11, 13 =Mobile12, 14 =Fissa0, 15 =Fissa1, 16 =Fissa2, 17 =Fissa3, 18 =Fissa4, 19 =Fissa5, 20 =Fissa6, 21 =Fissa7, 22 =Fissa8, 23 =Fissa9, 24 =Fissa10, 25 =Fissa11, 26 =Fissa12
Angolo (Angle)	2	1 =Radianti, 2 =Gradi, 3 =Gradianti
Formato esponenziale (Exponential Format)	3	1 =Normale, 2 =Scientifico, 3 =Tecnico
Reale o Complesso (Real or Complex)	4	1 =Reale, 2 =Rettangolare, 3 =Polare
Auto o Approssimato (Auto or Approx.)	5	1 =Auto, 2 =Approssimato

Nome modo	Intero modo	Interi impostazioni
Formato vettoriale (Vector Format)	6	1=Rettangolare, 2=Cilindrico, 3=Sferico
Base	7	1=Decimale, 2=Esadecimale, 3=Binario

getNum() (Ottieni/restituisci numeratore)

Catalogo > 

Trasforma l'argomento in un'espressione con denominatore comune ridotto e successivamente restituisce il numeratore.

GetStr

Menu Hub

GetStr[*promptString*,] *var*[, *statusVar*]

Per degli esempi vedere **Get**.

GetStr[*promptString*,] **funz**(*arg1*, ...*argn*)
, *statusVar*]

Istruzione di programmazione: funziona allo stesso modo del comando **Get**, tranne per il fatto che il valore ricevuto viene sempre interpretato come una stringa. A differenza di questo, il comando **Get** interpreta la risposta come espressione a meno che non sia racchiusa tra virgolette ("").

Nota: vedere anche **Get**, pagina 62 e **Send**, pagina 141.

getType()

Catalogo > 

getType(*var*) \Rightarrow stringa

Restituisce una stringa che indica il tipo di dati della variabile *var*.

Se *var* non è stata definita, restituisce la stringa "NONE" (NESSUNA).

$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$
getType (<i>temp</i>)	"LIST"
$3 \cdot i \rightarrow temp$	$3 \cdot i$
getType (<i>temp</i>)	"EXPR"
DelVar <i>temp</i>	<i>Done</i>
getType (<i>temp</i>)	"NONE"

getVarInfo() \Rightarrow matrice o stringa

getVarInfo

$(StringaNomeLibreria) \Rightarrow$ matrice o stringa

getVarInfo() restituisce una matrice di informazioni (nome di variabile, tipo, accessibilità della libreria e stato bloccato/sbloccato) per tutte le variabili e gli oggetti libreria definiti nell'attività corrente.

Se non ci sono variabili definite, **getVarInfo()** restituisce la stringa "NONE".

getVarInfo

$(StringaNomeLibreria)$ restituisce una matrice di informazioni per tutti gli oggetti libreria definiti nella libreria *StringaNomeLibreria*.

StringaNomeLibreria deve essere una stringa (testo racchiuso tra virgolette) o una variabile stringa.

Se la libreria *StringaNomeLibreria* non esiste, si produce un errore.

Notare l'esempio sulla sinistra, in cui il risultato di **getVarInfo()** è assegnato alla variabile *vs*. Se si tenta di visualizzare la riga 2 o la riga 3 di *vs* viene restituito un errore "Invalid list or matrix (Lista o matrice non valida)" perché almeno uno degli elementi di queste righe (ad esempio, variabile *b*) viene ricalcolato in una matrice.

Questo errore potrebbe ripresentarsi quando si utilizza *Ans* per ricalcolare un risultato di **getVarInfo()**.

<code>getVarInfo()</code>	"NONE"
Define <i>x</i> =5	<i>Done</i>
Lock <i>x</i>	<i>Done</i>
Define LibPriv <i>y</i> ={1,2,3}	<i>Done</i>
Define LibPub <i>z</i> (<i>x</i>)=3· <i>x</i> ² - <i>x</i>	<i>Done</i>
<code>getVarInfo()</code>	$\begin{bmatrix} x & \text{"NUM"} & \begin{bmatrix} \square & 1 \\ \square & 0 \end{bmatrix} \\ y & \text{"LIST"} & \begin{bmatrix} \text{LibPriv} & 0 \\ \text{LibPub} & 0 \end{bmatrix} \\ z & \text{"FUNC"} & \begin{bmatrix} \text{LibPub} & 0 \\ \text{LibPub} & 0 \end{bmatrix} \end{bmatrix}$
<code>getVarInfo("tmp3")</code>	"Error: Argument must be a string"
<code>getVarInfo("tmp3")</code>	$\begin{bmatrix} \text{volcyl2} & \text{"NONE"} & \begin{bmatrix} \text{LibPub} & 0 \end{bmatrix} \end{bmatrix}$

<i>a</i> :=1	1
<i>b</i> := $\begin{bmatrix} 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 \end{bmatrix}$
<i>c</i> := $\begin{bmatrix} 1 & 3 & 7 \end{bmatrix}$	$\begin{bmatrix} 1 & 3 & 7 \end{bmatrix}$
<i>vs</i> := <code>getVarInfo()</code>	$\begin{bmatrix} a & \text{"NUM"} & \begin{bmatrix} \square & 0 \\ \square & 0 \end{bmatrix} \\ b & \text{"MAT"} & \begin{bmatrix} \square & 0 \\ \square & 0 \end{bmatrix} \\ c & \text{"MAT"} & \begin{bmatrix} \square & 0 \\ \square & 0 \end{bmatrix} \end{bmatrix}$
<i>vs</i> [1]	$\begin{bmatrix} 1 & \text{"NUM"} & \begin{bmatrix} \square & 0 \\ \square & 0 \end{bmatrix} \end{bmatrix}$
<i>vs</i> [1,1]	1
<i>vs</i> [2]	"Error: Invalid list or matrix"
<i>vs</i> [2,1]	$\begin{bmatrix} 1 & 2 \end{bmatrix}$

getVarInfo() (Ottieni informazioni variabile)

Catalogo > 

Questo errore viene generato perché la versione corrente del software non supporta una struttura di matrice generalizzata quando un elemento di una matrice può essere o una matrice o una lista.

Goto (Vai a)

Catalogo > 

Goto nomeEtichetta

Trasferisce il controllo all'etichetta *nomeEtichetta*.

nomeEtichetta deve essere definito nella stessa funzione mediante un'istruzione **Lbl**.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define $g() = \text{Func}$

Done

Local *temp,i*

$0 \rightarrow \text{temp}$

$1 \rightarrow i$

Lbl *top*

$\text{temp} + i \rightarrow \text{temp}$

If $i < 10$ Then

$i + 1 \rightarrow i$

Goto *top*

EndIf

Return *temp*

EndFunc

$g()$

55

►Grad (Gradienti o Gradi centesimali)

Catalogo > 

Espr1 ► Grad ⇒ espressione

In modalità angolo in gradi:

Converte *Espr1* in una misura di angolo in gradi.

(1.5) ► Grad

(1.66667)^g

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Grad.

In modalità angolo in radianti:

(1.5) ► Grad

(95.493)^g

/

identity()

Catalogo > 

identity(*Intero*) ⇒ matrice

identity(4)

Restituisce la matrice identità con la dimensione di *Intero*.

$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Intero deve essere un numero intero positivo.

If

Catalogo > 

If *EsprBooleana*
Istruzione

Define $g(x)=\text{Func}$
If $x < 0$ Then
Return x^2
EndIf
EndFunc

Done

If *EsprBooleana* Then
Blocco
EndIf

$g(-2)$ 4

Se il valore di *EsprBooleana* è vero, viene eseguita la singola istruzione *Istruzione* o il blocco di istruzioni *Blocco* prima di procedere con l'esecuzione.

Se il valore di *EsprBooleana* è falso, la funzione continua senza eseguire l'istruzione o il blocco di istruzioni.

Blocco può essere una singola espressione o una serie di espressioni separate dal carattere ":".

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

If *EsprBooleana* Then
Blocco1
Else
Blocco2
EndIf

Define $g(x)=\text{Func}$
If $x < 0$ Then
Return $-x$
Else
Return x
EndIf
EndFunc

Done

Se il valore di *EsprBooleana* è vero, viene eseguito *Blocco1* e successivamente viene saltato *Blocco2*.

$g(12)$ 12
 $g(-12)$ 12

Se il valore di *EsprBooleana* è falso, viene saltato *Blocco1* ma viene eseguito *Blocco2*.

Blocco1 e *Blocco2* possono essere una singola istruzione.

If**If** *EsprBooleana1* **Then** *Blocco1***If** *EsprBooleana2* **Then** *Blocco2***:****Elseif** *EsprBooleanaN* **Then** *BloccoN***EndIf**

Permette il passaggio a un'altra funzione. Se il valore di *EsprBooleana1* è vero, viene eseguito *Blocco1*. Se *EsprBooleana1* è falso, calcola *EsprBooleana2* e così via.

Define $g(x) = \text{Func}$ **If** $x < -5$ **Then**

Return 5

Elseif $x > -5$ and $x < 0$ **Then** Return $-x$ **Elseif** $x \geq 0$ and $x \neq 10$ **Then** Return x **Elseif** $x = 10$ **Then**

Return 3

EndIf **EndFunc****Done** $g(-4)$

4

 $g(10)$

3

iffn()

iffn(*EsprBooleana*, *Valore_se_vero*, [*Valore_se_falso*, [*Valore_se_sconosciuto*]]) \Rightarrow *espressione, lista o matrice*

Calcola l'espressione booleana *EsprBooleana* (o ciascun elemento di *EsprBooleana*) e produce un risultato sulla base delle seguenti regole:

- *EsprBooleana* può verificare un singolo valore, una lista o una matrice.
- Se un elemento di *EsprBooleana* dà come risultato una condizione vera, viene restituito l'elemento corrispondente di *Valore_se_vero*.
- Se un elemento di *EsprBooleana* dà come risultato una condizione falsa, viene restituito l'elemento corrispondente di *Valore_se_falso*. Se *Valore_se_falso* è omesso, viene restituito *undef*.
- Se un elemento di *EsprBooleana* non dà come risultato una condizione vera né una condizione falsa, viene restituito l'elemento corrispondente di *Valore_se_sconosciuto*. Se *Valore_se_sconosciuto* è omesso, viene restituito *undef*.

iffn($\{1,2,3\} < 2.5, \{5,6,7\}, \{8,9,10\}$)
 $\{5,6,10\}$

Il valore di verifica di **1** è minore di 2.5, così l'elemento

Valore_se_vero corrispondente di **5** viene copiato nella lista del risultato.

Il valore di verifica di **2** è minore di 2.5, così l'elemento

Valore_se_vero corrispondente di **6** viene copiato nella lista del risultato.

Il valore di verifica di **3** è minore di 2.5, così l'elemento *Valore_se_falso* corrispondente di **10** viene copiato nella lista del risultato.

iffn($\{1,2,3\} < 2.5, 4, \{8,9,10\}$)
 $\{4,4,10\}$

Valore_se_vero è un singolo valore e corrisponde a qualsiasi posizione selezionata.

iffn($\{1,2,3\} < 2.5, \{5,6,7\}$)
 $\{5,6,\text{undef}\}$

iffFn()

Catalogo > 

- Se il secondo, terzo o quarto argomento della funzione **iffFn()** è una singola espressione, la verifica booleana viene applicata a tutte le posizioni in *EsprBooleana*.

Nota: se l'istruzione *EsprBooleana* semplificata implica una lista o una matrice, tutti gli altri argomenti della lista o della matrice devono avere uguali dimensioni e il risultato avrà uguali dimensioni.

Valore_se_falso non è specificato. Viene utilizzato *Undef*.

iffFn($\{2, "a"\} < 2.5, \{6, 7\}, \{9, 10\}, "err"$)
 $\{6, "err"\}$

Un elemento selezionato da *Valore_se_vero*.
Un elemento selezionato da *Valore_se_sconosciuto*.

imag()

Catalogo > 

Restituisce la parte immaginaria dell'argomento.

imag(ListaI) \Rightarrow lista

imag($\{-3, 4 - i, i\}$)
 $\{0, -1, 1\}$

Restituisce una lista delle parti non reali degli elementi.

imag(MatriceI) \Rightarrow matrice

Restituisce una matrice delle parti immaginarie degli elementi.

Conversione indiretta

Vedere #(), pagina 198.

inString()

Catalogo > 

inString(stringaRicerca, sottoStringa[, Inizio]) \Rightarrow intero

inString("Hello there", "the")
inString("ABCEFG", "D")

Restituisce la posizione del carattere nella stringa *stringaRicerca* dal quale inizia per la prima volta la stringa *sottoStringa*.

Inizio, se incluso, specifica la posizione del carattere all'interno di *stringaRicerca* in cui comincia la stringa. L'impostazione predefinita è 1 (il primo carattere di *stringaRicerca*).

Se *stringaRicerca* non contiene *sottoStringa* o *Inizio* è $>$ della lunghezza di *stringaRicerca*, viene restituito zero.

int()

int(Lista1) \Rightarrow lista
int(Matrice1) \Rightarrow matrice

int(-2.5)	-3.
int([-1.234 0 0.37])	[-2. 0 0.]

Restituisce il più grande dei numeri interi che è minore o uguale all'argomento.
 Questa funzione è identica a **floor()**.

L'argomento può essere un numero reale o complesso.

Con una lista o con una matrice, restituisce il numero intero maggiore di ciascuno degli elementi.

intDiv()

intDiv(Numer01, Numer02) \Rightarrow intero
intDiv(Lista1, Lista2) \Rightarrow lista
intDiv(Matrice1, Matrice2) \Rightarrow matrice

intDiv(-7,2)	-3
intDiv(4,5)	0
intDiv({12,-14,-16},{5,4,3})	{2,-3,5}

Restituisce la parte intera con segno di $(Numer01 \div Numer02)$.

Con liste e matrici, restituisce la parte intera con segno di $(argomento1 \div argomento2)$ per ciascuna coppia di elementi.

interpolare()

interpolare(valoreX, xLista, yLista, listaYprime) \Rightarrow lista

Questa funzione consente di fare quanto segue:

Equazione differenziale:

$y' = -3 \cdot y + 6 \cdot t + 5$ e $y(0) = 5$

<i>rk</i> =rk23[-3·y+6·t+5,t,y,{0,10},5,1)	
[0. 1. 2. 3. 4. 5. 3.19499 5.00394 6.99957 9.00593 1c]	

Per vedere l'intero risultato, premere \blacktriangle , quindi utilizzare \blacktriangleleft e \triangleright per spostare il cursore.

interpolare()

Catalogo > 

Dati $listaX$, $listaY = f(listaX)$, e $listaYprime = f'(listaX)$ per una funzione f non nota, viene utilizzata una interpolante cubica per approssimare la funzione f al valore X . Si assume che $listaX$ è una lista di numeri monotonicamente crescenti o decrescenti, ma questa funzione può restituire un valore anche quando non c'è. Questa funzione scorre $listaX$ cercando un intervallo $[listaX[i], listaX[i+1]]$ che contenga $valoreX$. Se trova un intervallo di questo tipo, restituisce un valore interpolato per $f(valoreX)$; altrimenti, restituisce **undef**.

$listaX$, $listaY$ e $listaPrimiY$ devono avere la stessa dimensione ≥ 2 e contenere espressioni che vengono semplificate in numeri.

Utilizzare la funzione `interpolate()` per calcolare i valori della funzione per `listavalorix`:

```
xvaluelist:=seq(i,i,0,10,0.5)
{0,0.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,}
xlist:=mat►list({rk[1]})
{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.}
ylist:=mat►list({rk[2]})
{5.,3.19499,5.00394,6.99957,9.00593,10.9978
yprimelist:=-3·y+6·t+5|y=ylist and t=xlist
{-10.,1.41503,1.98819,2.00129,1.98221,2.006
interpolate(xvaluelist,xlist,ylist,yprimelist)
{5.,2.67062,3.19499,4.02782,5.00394,6.00018}
```

invχ²()

Catalogo > 

invχ²(Area,df)

invChi2(Area,gf)

Calcola la funzione della probabilità χ^2 (chi quadrato) cumulativa inversa specificata dal grado di libertà, df , per una data $Area$ sotto la curva.

invF()

Catalogo > 

invF(Area,glNumer,glDenom)

invF(Area,glNumer,glDenom)

Calcola la funzione della distribuzione F cumulativa inversa specificata da $glNumer$ e $glDenom$ per una data $Area$ sotto la curva.

invBinom

(CumulativeProb, numProve, Prob, OutputForm)⇒ scalare o matrice

Dato il numero di tentativi (*NumTrials*) e la probabilità di successo di ogni tentativo (*Prob*), questa funzione restituisce il numero minimo di successi *k*, in modo che la probabilità cumulativa di successi *k* sia maggiore o uguale alla probabilità cumulativa (*CumulativeProb*) data.

OutputForm=0 visualizza il risultato come un valore scalare (impostazione predefinita).

OutputForm=1, visualizza il risultato come una matrice.

Esempio: Mary e Kevin stanno giocando a dadi. Mary deve indovinare il numero massimo di volte che il numero 6 può apparire in 30 lanci. Mary vince se il numero 6 appare il numero di volte indicate. Inoltre, più piccolo è il numero indovinato, maggiori saranno le sue vincite. Qual è il più piccolo numero che Mary può indovinare se desidera che le probabilità di vincere siano maggiori del 77%?

$\text{invBinom}\left(0.77, 30, \frac{1}{6}\right)$	6
$\text{invBinom}\left(0.77, 30, \frac{1}{6}, 1\right)$	$\begin{bmatrix} 5 & 0.616447 \\ 6 & 0.776537 \end{bmatrix}$

invBinomN(*CumulativeProb, Prob, NumSuccess, OutputForm*)⇒ scalare o matrice

Data la probabilità di successo di ogni tentativo (*Prob*) e il numero di successi (*NumSuccess*), questa funzione restituisce il numero minimo di tentativi *N*, in modo che la probabilità cumulativa di successi *x* sia minore o uguale alla probabilità cumulativa (*CumulativeProb*) data.

OutputForm=0 visualizza il risultato come un valore scalare (impostazione predefinita).

OutputForm=1, visualizza il risultato come una matrice.

Esempio: Monique si sta allenando a tirare in porta a netball. Monique sa per esperienza di avere il 70% di possibilità di realizzare ogni tiro e decide di continuare ad allenarsi fino a aver segnato 50 goal. Quanti tiri deve tentare per essere sicura che le probabilità di segnare almeno 50 goal siano maggiori di 0,99?

$\text{invBinomN}(0.01, 0.7, 49)$	86
$\text{invBinomN}(0.01, 0.7, 49, 1)$	$\begin{bmatrix} 85 & 0.010451 \\ 86 & 0.00709 \end{bmatrix}$

invNorm(*Area[,μ[,σ]]*)

invNorm() (Distribuzione normale cumulativa inversa)

Catalogo > 

Calcola la funzione della distribuzione normale cumulativa inversa per una data *Area* sotto la curva della distribuzione normale specificata da μ e σ .

invt() (Funzione della probabilità t di Student)

Catalogo > 

invt(*Area,gl*)

Calcola la funzione della probabilità cumulativa inversa t di Student specificata dal grado di libertà, *gl* per una data *Area* sotto la curva.

iPart()

Catalogo > 

iPart(*Numero*) \Rightarrow intero

iPart(*Lista1*) \Rightarrow lista

iPart(*Matrice1*) \Rightarrow matrice

iPart(-1.234)

-1.

iPart($\left\{ \frac{3}{2}, -2.3, 7.003 \right\}$)

{1, 2, 7.}

Restituisce la parte intera dell'argomento.

Per una lista o per una matrice, restituisce la parte intera di ciascun argomento.

L'argomento può essere un numero reale o complesso.

irr()

Catalogo > 

irr(*CF0,CFLista [CFFreq]*) \Rightarrow valore

Funzione finanziaria che calcola l'indice di rendimento interno di un investimento (Internal Rate of Return).

CF0 è il cash flow iniziale al tempo 0; deve essere un numero reale.

CFLista è una lista di importi di cash flow dopo il cash flow iniziale *CF0*.

list1:={6000,-8000,2000,-3000}

{6000, -8000, 2000, -3000}

list2:={2,2,2,1}

{2, 2, 2, 1}

irr(5000,list1,list2)

-4.64484

CFReg è una lista opzionale in cui ciascun elemento specifica la frequenza di occorrenza di un importo di cash flow raggruppato (consecutivo), che è l'elemento corrispondente di *CFLista*. L'impostazione predefinita è 1; è possibile inserire valori che siano numeri interi positivi < 10000.

Nota: Vedere anche **mirr()**, pagina 99.

isPrime(*Numero*) \Rightarrow *espressione costante booleana*

Restituisce vero o falso per indicare se *numero* è un numero intero ≥ 2 divisibile solo per se stesso e per 1.

Se *Numero* ha più di 306 cifre e non ha fattori ≤ 1021 , **isPrime(*Numero*)** visualizza un messaggio di errore.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

isPrime(5)	true
isPrime(6)	false

Funzione per trovare il numero primo successivo dopo il numero specificato.

Define nextprim(<i>n</i>)=Func	Done
Loop	
<i>n</i> +1 \rightarrow <i>n</i>	
If isPrime(<i>n</i>)	
Return <i>n</i>	
EndLoop	
EndFunc	
nextprim(7)	11

isVoid(*Var*) \Rightarrow *espressione costante booleana*

isVoid(*Espr*) \Rightarrow *espressione costante booleana*

isVoid(*Lista*) \Rightarrow *lista di espressione costante booleana*

Restituisce vero o falso per indicare se l'argomento è un tipo di dati vuoto.

Per ulteriori informazioni sugli elementi vuoti, vedere pagina 222.

<i>a</i> := <u> </u>	=
isVoid(<i>a</i>)	true
isVoid({1,_,3})	{ false,true,false }

Lbl (Etichetta)**Catalogo > ****Lbl** *nomeEtichetta*Definisce un'etichetta chiamata *nomeEtichetta* in una funzione.È possibile utilizzare un'istruzione **Goto** *nomeEtichetta* per trasferire il controllo del programma all'istruzione immediatamente successiva all'etichetta.*nomeEtichetta* deve soddisfare gli stessi requisiti validi per i nomi delle variabili.**Nota per l'inserimento dell'esempio:** per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.Define *g()*=Func*Done*Local *temp,i*0→*temp*1→*i*Lbl *top**temp*+*i*→*temp*If *i*<10 Then*i*+1→*i*Goto *top*

EndIf

Return *temp*

EndFunc

g()

55

Icm() (Minimo comune multiplo)**Catalogo > ****Icm**(*Numero1, Numero2*)⇒*espressione***Icm**(*List1, Lista2*)⇒*lista***Icm**(*Matrice1, Matrice2*)⇒*matrice*Restituisce il minimo comune multiplo (lcm) di due argomenti. Il lcm di due frazioni è il **minimo comune multiplo** dei loro numeratori diviso per il **massimo comune divisor (gcd)** dei loro denominatori. Il lcm dei numeri frazionari a virgola mobile è il loro prodotto.

Per due liste o matrici, restituisce i minimi comuni multipli dei corrispondenti elementi.

lcm(6,9)

18

lcm($\left\{ \frac{1}{3}, -14, 16 \right\}, \left\{ \frac{2}{15}, 7, 5 \right\} \right\}$ $\left\{ \frac{2}{3}, 14, 80 \right\}$ **left() (Sinistra)****Catalogo > ****left**(*stringaOrigine*[, *Num*])⇒*stringa*

left("Hello",2)

"He"

Restituisce i caratteri *Num* più a sinistra contenuti nella stringa di caratteri *stringaOrigine*.

Se si omette *Num*, restituisce l'intera *stringaOrigine*.

left(Lista1[, Num])⇒*lista*

`left({1,3,2,4},3)`

$\{1,3,2\}$

Restituisce gli elementi *Num* più a sinistra contenuti in *Lista1*.

Se si omette *Num*, restituisce l'intera *Lista1*.

left(Confronto)⇒espressione

`left(x<3)`

x

Restituisce il primo membro di un'equazione o di una disequazione.

libShortcut() (Collegamento a libreria)

libShortcut(stringaNomeLibr, stringaNomeCollegamento[, LibPrivFlag])⇒*lista di variabili*

Crea una gruppo di variabili nell'attività corrente che contiene i riferimenti a tutti gli oggetti nel documento libreria specificato *stringaNomeLibr*. Aggiunge inoltre i membri del gruppo al menu Variables (Variabili). È quindi possibile fare riferimento a ciascun oggetto utilizzando la relativa *stringaNomeCollegamento*.

Impostare *LibPrivFlag=0* per escludere oggetti libreria privata (default)

Impostare *LibPrivFlag=1* per includere oggetti libreria privata

Per copiare un gruppo di variabili, vedere **CopyVar**, pagina 26.

Per eliminare un gruppo di variabili, vedere **DelVar**, pagina 39.

Questo esempio presuppone un documento libreria memorizzato e aggiornato, denominato **linalg2**, che contiene oggetti definiti come *clearmat*, *gauss1* e *gauss2*.

`getVarInfo("linalg2")`

$$\begin{bmatrix} clearmat & "FUNC" & "LibPub" \\ gauss1 & "PRGM" & "LibPriv" \\ gauss2 & "FUNC" & "LibPub" \end{bmatrix}$$

`libShortcut("linalg2","la")`

$\{la.clearmat,la.gauss2\}$

`libShortcut("linalg2","la",1)`

$\{la.clearmat,la.gauss1,la.gauss2\}$

LinRegBx *X, Y[,Freq[,Categoria,Includi]]*

Calcola la regressione lineare $y = a + b \cdot x$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a + b \cdot x$
stat.a, stat.b	Coefficienti di regressione
stat.r ²	coefficiente di determinazione
stat.r	Coefficiente di correlazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

LinRegMx X, Y[, Freq[, Categoria, Includi]]

Calcola la regressione lineare $y = m \cdot x + b$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $m \cdot x + b$
stat.m, stat.b	Coefficienti di regressione
stat.r ²	coefficiente di determinazione
stat.r	Coefficiente di correlazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

LinRegtIntervals $X, Y[, F[, 0$
 $, livelloConfidenza]]]$

Per pendenza Calcola un intervallo di confidenza di livello C per la pendenza.

LinRegtIntervals $X, Y[, F[, 1, ValX$
 $, livelloConfidenza]]]$

Per risposta. Calcola un valore y previsto, un intervallo di previsione del livello C per una singola osservazione e un intervallo di confidenza del livello C per la risposta media.

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere la stessa dimensione.

X e *Y* sono liste di variabili indipendenti e dipendenti.

F è una lista opzionale di valori di frequenza. Ciascun elemento di *F* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a+b \cdot x$
stat.a, stat.b	Coefficienti di regressione
stat.df	Gradi di libertà
stat.r ²	Coefficiente di determinazione
stat.r	Coefficiente di correlazione
stat.Resid	Residui della regressione

Solo tipo per pendenza

Variabile di output	Descrizione
[stat.CLower, stat.CUpper]	Intervallo di confidenza per la pendenza

Variabile di output	Descrizione
stat.ME	Margine di errore dell'intervallo di confidenza
stat.SESlope	Errore standard della pendenza
stat.s	Errore standard sulla linea

Solo tipo per risposta

Variabile di output	Descrizione
[stat.CLower, stat.CUpper]	Intervallo di confidenza per la risposta media
stat.ME	Margine di errore dell'intervallo di confidenza
stat.SE	Errore standard della risposta media
[stat.LowerPred, stat.UpperPred]	Intervallo di previsione per una singola osservazione
stat.MEPred	Margine di errore dell'intervallo di previsione
stat.SEPred	Errore standard per la previsione
stat. \hat{y}	$a + b \cdot \text{ValX}$

LinRegtTest (t Test regressione lineare)

Catalogo > 

LinRegtTest $X, Y[, Freq, Ipotesi]$

Calcola una regressione lineare sulle liste X e Y e un t test sul valore della pendenza β e il coefficiente di correlazione p per l'equazione $y=\alpha+\beta x$. Viene verificata l'ipotesi nulla $H_0: \beta=0$ (in modo equivalente, $p=0$) in relazione a una di tre ipotesi alternative.

Tutte le liste devono avere la stessa dimensione.

X e Y sono liste di variabili indipendenti e dipendenti.

$Freq$ è una lista opzionale di valori di frequenza. Ciascun elemento di $Freq$ specifica la frequenza di occorrenza di ogni dato corrispondente di X e Y . Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Ipotesi è un valore opzionale che specifica una di tre ipotesi alternative rispetto alla quale verrà testata l'ipotesi nulla ($H_0: \beta = p = 0$).

Per $H_a: \beta \neq 0$ e $p \neq 0$ (default), impostare *Ipotesi*=0

Per $H_a: \beta < 0$ e $p < 0$, impostare *Ipotesi*<0

Per $H_a: \beta > 0$ e $p > 0$, impostare *Ipotesi*>0

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a + b \cdot x$
stat.t	Statistica T per il test di significatività
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Gradi di libertà
stat.a, stat.b	Coefficienti di regressione
stat.s	Errore standard sulla linea
stat.SESlope	Errore standard della pendenza
stat.r ²	Coefficiente di determinazione
stat.r	Coefficiente di correlazione
stat.Resid	Residui della regressione

linSolve()**Catalogo > **

linSolve(*SistemaDiEqLineari, Var1, Var2, ...)***⇒lista**

linSolve(*EqLineare1 and EqLineare2 and ..., Var1, Var2, ...)***⇒lista**

linSolve(*{EqLineare1, EqLineare2, ...}, Var1, Var2, ...)***⇒lista**

linSolve(*SistemaDiEqLineari, {Var1, Var2, ...})***⇒lista**

linSolve(*EqLineare1 and EqLineare2 and ..., {Var1, Var2, ...})***⇒lista**

linSolve(*{EqLineare1, EqLineare2, ...}, {Var1, Var2, ...})***⇒lista**

Restituisce una lista di soluzioni per le variabili *Var1, Var2, ...*

Il primo argomento deve calcolare un sistema di equazioni lineari o una singola equazione lineare. Diversamente, si ottiene un argomento errato.

Ad esempio, calcolando `linSolve(x=1 e x=2,x)` si ottiene un risultato “Argomento errato”.

$$\text{linSolve}\left(\left\{\begin{array}{l} 2x+4y=3 \\ 5x-3y=7 \end{array}\right., \{x,y\}\right) \quad \left\{\begin{array}{l} \frac{37}{26}, \frac{1}{26} \end{array}\right\}$$

$$\text{linSolve}\left(\left\{\begin{array}{l} 2x=3 \\ 5x-3y=7 \end{array}\right., \{x,y\}\right) \quad \left\{\begin{array}{l} \frac{3}{2}, \frac{1}{6} \end{array}\right\}$$

$$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple}+4\cdot\text{pear}=23 \\ 5\cdot\text{apple}-\text{pear}=17 \end{array}\right., \{\text{apple},\text{pear}\}\right) \quad \left\{\begin{array}{l} \frac{13}{3}, \frac{14}{3} \end{array}\right\}$$

$$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple}+4\cdot\frac{\text{pear}}{3}=14 \\ -\text{apple}+\text{pear}=6 \end{array}\right., \{\text{apple},\text{pear}\}\right) \quad \left\{\begin{array}{l} \frac{36}{13}, \frac{114}{13} \end{array}\right\}$$

Δlist() (Differenza in una lista)**Catalogo > **

Δlist(*Lista1***)⇒lista**

ΔList(*{ 20,30,45,70 }***)** { 10,15,25 }

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando `deltaList(...)`.

Restituisce una lista contenente le differenze tra elementi consecutivi in *Lista1*. Ogni elemento di *Lista1* viene sottratto dal successivo elemento di *Lista1*. La lista risultante è sempre composta da un elemento in meno della *Lista1* originale.

list►mat() (Da lista a matrice)**Catalogo > ****list►mat(Lista [, elementiPerRiga])⇒matrice**Restituisce una matrice completata riga per riga con gli elementi di *Lista*.*elementiPerRiga*, se incluso, specifica il numero di elementi per riga.L'impostazione predefinita corrisponde al numero di elementi di *Lista* (una riga).Se *Lista* non completa la matrice risultante, viene aggiunta una serie di zeri.**Nota:** è possibile inserire questa funzione dalla tastiera del computer digitando **list@>mat(...)**.

list►mat({1,2,3})	[1 2 3]
list►mat({1,2,3,4,5},2)	[1 2 3 4 5 0]

ln() (Logaritmo naturale)**Tasti  ****ln(Lista1)⇒lista**

Restituisce il logaritmo naturale dell'argomento.

In una lista, restituisce i logaritmi naturali degli elementi.

ln(2.)	0.693147
--------	----------

Se la modalità del formato complesso è Reale:

ln({-3,1,2,5})	"Error: Non-real calculation"
----------------	-------------------------------

ln**(matriceQuadrata1)⇒matriceQuadrata**Restituisce il logaritmo naturale della matrice *matriceQuadrata1*. Ciò non equivale a calcolare il logaritmo naturale di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.*matriceQuadrata1* deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

Se la modalità del formato complesso è Rettangolare:

In modalità angolo in radianti e in modalità formato rettangolare complesso:

ln	$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$
	$1.83145+1.73485 \cdot i \quad 0.009193-1.49086$
	$0.448761-0.725533 \cdot i \quad 1.06491+0.623491 \cdot i$
	$-0.266891-2.08316 \cdot i \quad 1.12436+1.79018 \cdot i$

Per vedere l'intero risultato, premere **▲**, quindi utilizzare **◀** e **▶** per spostare il cursore.

LnReg *X, Y[, Freq [, Categoria, Includi]]*

Calcola la regressione logaritmica $y = a + b \cdot \ln(x)$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a + b \cdot \ln(x)$
stat.a, stat.b	Coefficienti di regressione
stat.r ²	Coefficiente di determinazione lineare di dati trasformati
stat.r	Coefficiente di correlazione per dati trasformati ($\ln(x)$, <i>y</i>)
stat.Resid	Residui associati al modello logaritmico
stat.ResidTrans	Residui associati all'adattamento lineare dei dati trasformati
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>

Variabile di output	Descrizione
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

Local (Variabile locale)

Catalogo > 

Local *Var1*[, *Var2*] [, *Var3*] ...

Definisce le *var* specificate come variabili locali. Tali variabili esistono solamente durante il calcolo di una funzione e vengono cancellate al termine dell'esecuzione di tale funzione.

Nota: le variabili locali permettono di risparmiare memoria in quanto esistono solo temporaneamente. Inoltre, esse non influiscono sui valori delle variabili globali esistenti. Le variabili locali devono essere utilizzate per i cicli **For** e per salvare in maniera provvisoria i valori in una funzione su diverse righe, poiché le modifiche sulle variabili globali non sono ammesse in una funzione.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define *rollcount*[]=Func

```

Local i
1 → i
Loop
If randInt(1,6)=randInt(1,6)
Goto end
i+1 → i
EndLoop
Lbl end
Return i
EndFunc
```

Done

<i>rollcount</i> []	16
---------------------	----

<i>rollcount</i> []	3
---------------------	---

Lock

Catalogo > 

Lock*Var1* [, *Var2*] [, *Var3*] ...

Lock*Var*.

Blocca le variabili o il gruppo di variabili specificate. Le variabili bloccate non possono essere modificate o eliminate.

Non è possibile bloccare o sbloccare la variabile di sistema *Ans*, inoltre non è possibile bloccare i gruppi di variabili di sistema *stat*. o *tvm*.

<i>a</i> :=65	65
---------------	----

Lock <i>a</i>	<i>Done</i>
---------------	-------------

getLockInfo(<i>a</i>)	1
-------------------------	---

<i>a</i> :=75	"Error: Variable is locked."
---------------	------------------------------

DelVar <i>a</i>	"Error: Variable is locked."
-----------------	------------------------------

Unlock <i>a</i>	<i>Done</i>
-----------------	-------------

<i>a</i> :=75	75
---------------	----

DelVar <i>a</i>	<i>Done</i>
-----------------	-------------

Nota: Il comando **Blocca (Lock)** cancella la cronologia di Annulla/Ripeti quando è applicato a variabili sbloccate.

Vedere **unLock**, pagina 173 e **getLockInfo()**, pagina 67.

log() (Logaritmo)

Tasti  

Nota: vedere anche **Modello di logaritmo**, pagina 2.

Se *Espr2* viene omesso, come base viene utilizzato 10.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

Se l'argomento base viene omesso, come base viene utilizzato 10.

Se la modalità del formato complesso è Reale:

Se la modalità del formato complesso è Rettangolare:

In modalità angolo in radianti e in modalità formato rettangolare complesso:

$$\log_{10} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$$

$$\begin{bmatrix} 0.795387 + 0.753438 \cdot i & 0.003993 - 0.6474 \cdot i \\ 0.194895 - 0.315095 \cdot i & 0.462485 + 0.2707 \cdot i \\ -0.115909 - 0.904706 \cdot i & 0.488304 + 0.7774 \cdot i \end{bmatrix}$$

Per vedere l'intero risultato, premere **▲**, quindi utilizzare **◀** e **▶** per spostare il cursore.

Logistic (Regressione logistica)

Catalogo > 

Logistic *X, Y[, Freq] [, Categoria, Includi]*]

Calcola la regressione logisticay = (c/(1+a · e^{-bx})) sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $c/(1+a \cdot e^{-bx})$
stat.a, stat.b, stat.c	Coefficienti di regressione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

LogisticD *X, Y* [, *[Iterazioni]*], *[Freq]* [, *Categoria*, *Includi*]

Calcola la regressione logistica $y = (c/(1+a \cdot e^{-bx})+d)$ sulle liste *X* e *Y* con frequenza *Freq*, utilizzando un numero specificato di *Iterazioni*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Iterazioni è un valore opzionale che specifica quante volte al massimo verrà tentata una soluzione. Se omesso, viene utilizzato 64. Di solito valori più alti danno una maggiore accuratezza ma richiedono tempi di esecuzione più lunghi, e viceversa.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $c/(1+a \cdot e^{-bx})+d)$
stat.a, stat.b, stat.c, stat.d	Coefficienti di regressione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

Loop

Blocco

EndLoop

Esegue ciclicamente le istruzioni di *Blocco*. Si noti che un ciclo viene eseguito infinite volte, se non si trovano istruzioni **Goto** o **Exit** all'interno di *Blocco*.

Blocco è una sequenza di istruzioni separate dal carattere.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

```
Define rollcount()=Func
  Local i
  1→i
  Loop
  If randInt(1,6)=randInt(1,6)
  Goto end
  i+1→i
  EndLoop
  Lbl end
  Return i
EndFunc
```

Done

rollcount()	16
rollcount()	3

LU (Scomposizione inferiore - superiore)

LU Matrice, MatriceL, MatriceU, MatriceP[,Tol]

Calcola la scomposizione LU (lower-upper, inferiore-superiore) di una matrice reale o complessa. La matrice triangolare inferiore è memorizzata in *MatriceL*, quella superiore in *MatriceU* e la matrice di permutazione (che descrive gli scambi di riga eseguiti durante i calcoli) in *MatriceP*.

$$\text{MatriceL} \cdot \text{MatriceU} = \text{MatriceP} \cdot \text{matrice}$$

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Toll*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Toll* viene ignorato.

- Se si usa **ctrl** **enter** oppure se si imposta la modalità **Auto** o

$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$
LU m1,lower,upper,perm	Done
<i>lower</i>	$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
<i>upper</i>	$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
<i>perm</i>	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Approssimato su Approssimato, i calcoli verranno eseguiti in virgola mobile.

- Se *Toll* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:
 $5E-14 \cdot \max(\dim(\text{Matrice})) \cdot \text{rowNorm}(\text{Matrice})$

L'algoritmo di scomposizione in fattori LU usa il pivoting parziale per lo scambio di righe.

M

matlist() (Da matrice a lista)

matlist(*Matrice*)⇒*lista*

Restituisce una lista completata con gli elementi di *Matrice*. Gli elementi sono copiati da *Matrice* riga per riga.

matlist([1 2 3])	{1,2,3}
[1 2 3] → <i>m1</i>	[1 2 3]
matlist(<i>m1</i>)	{1,2,3,4,5,6}

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando `mat@>list(...)`.

max() (Massimo)

max(*Lista1, Lista2*)⇒*lista*

max(*Matrice1, Matrice2*)⇒*matrice*

Restituisce il massimo di due argomenti; se questi sono due liste o matrici, restituisce una lista o matrice contenente il valore massimo di ciascuna coppia di elementi corrispondenti.

max(2.3,1.4)	2.3
max({1,2},{-4,3})	{1,3}

max(*Lista*)⇒*espressione*

Restituisce l'elemento massimo di *lista*.

max({0,1,-7,1.3,0.5})	1.3
-----------------------	-----

max(*Matrice1*)⇒*matrice*

Restituisce un vettore riga contenente l'elemento massimo di ciascuna colonna di *Matrice1*.

max([1 -3 7 -4 0 0.3])	[1 0 7]
---------------------------	---------

max() (Massimo)

Catalogo > 

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

Nota: vedere anche **min()**.

mean() Media

mean(*Lista*[, *listFreq*]) \Rightarrow espressione

Restituisce la media degli elementi di *Lista*.

Ogni elemento di *listFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

mean(*Matrice1*[, *matriceFreq*]) \Rightarrow matrice

Restituisce un vettore riga contenente le medie di tutte le colonne di *Matrice1*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Matrice1*.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

Catalogo > 

mean({0.2,0.1,-0.3,0.4})	0.26
mean({1,2,3},{3,2,1})	$\frac{5}{3}$

In modalità formato vettore rettangolare:

mean	$\begin{bmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{bmatrix}$	$[-0.133333 \quad 0.833333]$
mean	$\begin{bmatrix} \frac{1}{5} & 0 \\ -1 & 3 \\ \frac{2}{5} & -\frac{1}{2} \\ 5 & 2 \end{bmatrix}$	$\left[\frac{-2}{15} \quad \frac{5}{6} \right]$
mean	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \begin{bmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{bmatrix}$	$\left[\frac{47}{15} \quad \frac{11}{3} \right]$

median() (Mediana)

Catalogo > 

median(*Lista*[, *listFreq*]) \Rightarrow espressione

Restituisce la mediana degli elementi di *Lista*.

Ogni elemento di *listFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

median(*Matrice1*[, *matriceFreq*]) \Rightarrow matrice

Restituisce un vettore riga contenente le mediane degli elementi delle colonne di *Matrice1*.

median({0.2,0.1,-0.3,0.4})	0.2
----------------------------	-----

median	$\begin{bmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{bmatrix}$	$[0.4 \quad -0.3]$
--------	---	--------------------

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Matrice1*.

Note:

- Tutti gli elementi nella lista o nella matrice devono essere semplificati in numeri.
- Gli elementi vuoti (nulli) di una lista o matrice vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

MedMed (Linea mediana-mediana)

MedMed *X, Y [, Freq] [, Categoria, Includi]*]]

Calcola la linea mediana-medianay = $(m \cdot x + b)$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione linea mediana-mediana: $m \cdot x + b$
stat.m, stat.b	Coefficienti del modello
stat.Resid	Residui della linea mediana-mediana
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

mid() (In mezzo alla stringa)

Catalogo > 

mid(stringaOrigine, Inizio[, Cont])⇒stringa

Restituisce *Cont* caratteri dalla stringa di caratteri *stringaOrigine*, incominciando dal numero di carattere di *Inizio*.

Se *Cont* viene omesso o se è maggiore della dimensione di *stringaOrigine*, restituisce tutti i caratteri di *stringaOrigine*, incominciando dal numero di carattere di *Inizio*.

Cont deve essere ≥ 0 . Se *Cont* = 0, viene restituita una stringa vuota.

mid(listaOrigine, Inizio [, Cont])⇒lista

Restituisce *Cont* elementi da *listaOrigine*, iniziando dal numero di elemento di *Inizio*.

Se *Cont* viene omesso o è maggiore della dimensione di *listaOrigine*, restituisce tutti gli elementi di *listaOrigine*, incominciando dal numero di elemento di *Inizio*.

Cont deve essere ≥ 0 . Se *Cont* = 0, restituisce una lista vuota.

mid(listaStringaOrigine, Inizio[, Cont])⇒lista

mid("Hello there",2)	"ello there"
mid("Hello there",7,3)	"the"
mid("Hello there",1,5)	"Hello"
mid("Hello there",1,0)	"[]"

mid({9,8,7,6},3)	{7,6}
mid({9,8,7,6},2,2)	{8,7}
mid({9,8,7,6},1,2)	{9,8}
mid({9,8,7,6},1,0)	{[]}

mid({ "A", "B", "C", "D" },2,2)	{ "B", "C" }
---------------------------------	--------------

mid() (In mezzo alla stringa)

Catalogo > 

Restituisce *Cont* stringhe della lista delle stringhe *listaStringheOrigine*, incominciano dal numero di elemento di *Inizio*.

min() (Minimo)

Catalogo > 

min(Lista1, Lista2)⇒lista

$\min\{2,3,1,4\}$ 1.4

min(Matrice1, Matrice2)⇒matrice

$\min\{\{1,2\},\{-4,3\}\}$ {-4,2}

Restituisce il minimo di due argomenti. Se gli argomenti sono due liste o matrici, viene restituita una lista o una matrice contenente il valore minimo di ciascuna coppia di elementi corrispondenti.

min(Lista)⇒espressione

$\min\{\{0,1,-7,1,3,0,5\}\}$ -7

Restituisce l'elemento minimo di *Lista*.

min(Matrice1)⇒matrice

$\min\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}$ [-4 -3 0.3]

Restituisce un vettore di riga contenente l'elemento minimo di ciascuna colonna di *Matrice1*.

Nota: vedere anche **max()**.

mirr()

Catalogo > 

mirr
(
tassoFinanziamento,
tassoReinvestimento,*CF0*,*CFLista*
[,*CFFreq*])

$list1 := \{6000, -8000, 2000, -3000\}$
{6000, -8000, 2000, -3000}
 $list2 := \{2, 2, 2, 1\}$ {2,2,2,1}
 $\text{mirr}(4.65, 12, 5000, list1, list2)$ 13.41608607

Funzione finanziaria che restituisce l'indice interno di rendimento modificato di un investimento.

tassoFinanziamento è il tasso di interesse che si paga sugli importi di cash flow.

tassoReinvestimento è il tasso di interesse in corrispondenza del quale i cash flow vengono reinvestiti.

CFO è il cash flow iniziale al tempo 0; deve essere un numero reale.

$CFLista$ è una lista di importi di cash flow dopo il cash flow iniziale CFO .

$CFFreq$ è una lista opzionale in cui ciascun elemento specifica la frequenza di occorrenza di un importo di cash flow raggruppato (consecutivo), che è l'elemento corrispondente di $CFLista$. L'impostazione predefinita è 1; è possibile inserire valori che siano numeri interi positivi < 10000.

Nota: vedere anche **irr()**, pagina 78.

mod() (Modulo)

mod(Lista1, Lista2) \Rightarrow lista

mod(Matrice1, Matrice2) \Rightarrow matrice

Restituisce il primo argomento modulo secondo argomento, come definito dalle identità:

$$\text{mod}(x, 0) = x$$

$$\text{mod}(x, y) = x - y \text{ floor}(x/y)$$

Quando il secondo argomento è diverso da zero, il risultato è periodico in tale argomento. Il risultato può essere zero oppure ha lo stesso segno del secondo argomento.

Se gli argomenti sono due liste o due matrici, viene restituita una lista o una matrice contenente il modulo di ciascuna coppia di elementi corrispondenti.

Nota: vedere anche **remain()**, pagina 132

mod(7,0)	7
mod(7,3)	1
mod(-7,3)	2
mod(7,-3)	-2
mod(-7,-3)	-1
mod({12,-14,16},{9,7,-5})	{3,0,-4}

mRow() (Operazione con righe di matrice)**Catalogo > **

$$\text{mRow}\left(\frac{-1}{3}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right) \quad \begin{bmatrix} 1 & 2 \\ -1 & -4 \\ 3 & \end{bmatrix}$$

mRowAdd() (Moltiplicazione e somma di righe di matrice)**Catalogo > **

Restituisce una copia di *Matrice1* dove ciascun elemento della riga *Indice2* di *Matrice1* viene sostituito con:

*Indice2***MultReg (Regressione lineare multipla)****Catalogo > ****MultReg** *Y, X1[,X2[,X3,...[,X10]]]*

Calcola la regressione lineare multipla della lista *Y* sulle liste *X1, X2, ..., X10*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere la stessa dimensione.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $b0+b1 \cdot x1+b2 \cdot x2+ \dots$
stat.b0, stat.b1, ...	Coefficienti di regressione
stat.R ²	Coefficiente di determinazione multipla
stat.yList	$\hat{y}\text{List} = b0+b1 \cdot x1+ \dots$
stat.Resid	Residui della regressione

MultRegIntervals (Intervalli di confidenza della previsione di regressione multipla)**Catalogo > ****MultRegIntervals** *Y, X1[,X2[,X3,...*

MultRegIntervals (Intervalli di confidenza della previsione di regressione multipla)

Catalogo > 

[,X10]],listaValX[,livelloConfidenza]

Calcola un valore y previsto, un intervallo di previsione del livello C per una singola osservazione e un intervallo di confidenza del livello C per la risposta media.

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere la stessa dimensione.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $b0+b1 \cdot x1+b2 \cdot x2+ \dots$
stat.ŷ	Una stima del punto: $\hat{y} = b0 + b1 \cdot x1 + \dots$ per <i>listaValX</i>
stat.dfError	Gradi di libertà degli errori
stat.CLower, stat.CUpper	Intervallo di confidenza per una risposta media
stat.ME	Margine di errore dell'intervallo di confidenza
stat.SE	Errore standard della risposta media
stat.LowerPred, stat.UpperrPred	Intervallo di previsione per una singola osservazione
stat.MEPred	Margine di errore dell'intervallo di previsione
stat.SEPred	Errore standard per la previsione
stat.bList	Lista dei coefficienti di regressione, {b0,b1,b2,...}
stat.Resid	Residui della regressione

MultRegTests (Verifica sulla regressione lineare multipla)

Catalogo > 

MultRegTests Y, X1[,X2[,X3,...[,X10]]]

Il test di regressione lineare multipla calcola una regressione lineare multipla sui dati assegnati e fornisce la statistica F test globale e le statistiche t test per i coefficienti.

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Output

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $b0+b1 \cdot x1+b2 \cdot x2+ \dots$
stat.F	Statistica della verifica F globale
stat.PVal	Valore P associato alla statistica F globale
stat.R ²	Coefficiente di determinazione multipla
stat.AdjR ²	Coefficiente modificato di determinazione multipla
stat.s	Deviazione standard dell'errore
stat.DW	Statistica d Durbin-Watson; utilizzata per determinare se la correlazione automatica di primo ordine è presente nel modello
stat.dfReg	Gradi di liberà della regressione
stat.SSReg	Somma dei quadrati della regressione
stat.MSReg	Quadrato medio della regressione
stat.dfError	Gradi di libertà degli errori
stat.SSError	Somma dei quadrati degli errori
stat.MSError	Quadrato medio degli errori
stat.bList	{ $b0, b1, \dots$ } Lista dei coefficienti
stat.tList	Lista di statistiche t , una per ogni coefficiente di bList
stat.PList	Lista di valori P per ogni statistica t
stat.SEList	Lista di errori standard per coefficienti di bList

Variabile di output	Descrizione
stat.yList	lista \hat{y} = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residui della regressione
stat.sResid	Residui standardizzati; valore ottenuto dividendo un residuo per la sua deviazione standard
stat.CookDist	Distanza di Cook; misura dell'influenza di un'osservazione basata sui valori residui e di leverage
stat.Leverage	Misura della distanza dei valori della variabile indipendente dai rispettivi valori medi

N

nand

tasti

BooleanExpr1 nand BooleanExpr2
restuisce Boolean expression

$x \geq 3$ and $x \geq 4$	$x \geq 4$
$x \geq 3$ nand $x \geq 4$	$x < 4$

BooleanList1 nand BooleanList2
restuisce Boolean list

BooleanMatrix1 nand BooleanMatrix2
restuisce Boolean matrix

Restituisce la negazione di un'operazione **and** logica riguardo i due argomenti.
Restituisce true, false o una forma semplificata dell'equazione.

Nel caso di liste e matrici, restituisce un confronto elemento per elemento.

Integer1 nand Integer2 \Rightarrow integer

Confronta due interi reali bit per bit tramite un'operazione **nand**. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati i bit corrispondenti, il risultato sarà 0 se entrambi sono uguali a 1; in caso contrario, il risultato sarà 1. Il valore restituito rappresenta il risultato dei bit e viene visualizzato secondo la modalità base.

3 and 4	0
3 nand 4	-1
{1,2,3} and {3,2,1}	{1,2,1}
{1,2,3} nand {3,2,1}	{-2,-3,-2}

È possibile inserire gli interi in qualsiasi base numerica. Nel caso di un numero binario o esadecimale, è necessario utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli interi vengono considerati decimali (base 10)

nCr() (Combinazioni)

Catalogo > 

nCr(Lista1, Lista2)⇒lista

$nCr(\{5,4,3\}, \{2,4,2\})$ {10,1,3}

Restituisce una lista di combinazioni sulla base delle coppie di elementi corrispondenti nelle due liste. Le liste degli argomenti devono avere dimensioni uguali.

nCr(Matrice1, Matrice2)⇒matrice

$nCr(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix})$ $\begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$

Restituisce una matrice di combinazioni, sulla base delle coppie di elementi corrispondenti nelle due matrici. Le matrici degli argomenti devono avere le stesse dimensioni.

nDerivative()

Catalogo > 

nDerivative(Espr1, Var=Valore [,Ordine])⇒valore

$nDerivative(|x|, x=1)$ 1

nDerivative(Espr1, Var[,Ordine]) | Var=Valore⇒valore

$nDerivative(|x|, x)|x=0$ undef

$nDerivative(\sqrt{x-1}, x)|x=1$ undef

Restituisce la derivata numerica calcolata con metodi di differenziazione automatica.

Se il *Valore* è specificato, lo stesso sostituisce qualsiasi assegnazione di variabile precedente o qualsiasi sovrapposizione corrente “|” della variabile.

Ordine della derivata deve essere **1** o **2**.

newList() (Nuova lista)**Catalogo > ****newList(*numElementi*) \Rightarrow lista**

newList(4)

{0,0,0,0}

Restituisce una lista le cui dimensioni sono *numElementi*. Ciascun elemento è zero.

newMat() (Nuova matrice)**Catalogo > ****newMat(*numRighe*,
numColonne) \Rightarrow matrice**

newMat(2,3)

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Restituisce una matrice di zeri le cui dimensioni sono date da *numRighe* per *numColonne*.

nfMax() (Massimo di una funzione calcolato numericamente)**Catalogo > ****nfMax(*Espr*, *Var*) \Rightarrow valore**nfMax($-x^2 - 2 \cdot x - 1, x$)

-1.

nfMax(*Espr*, *Var*, *estremoInf*) \Rightarrow valorenfMax($0.5 \cdot x^3 - x - 2, x, -5, 5$)

5.

**nfMax(*Espr*, *Var*, *estremoInf*,
estremoSup) \Rightarrow valore**

**nfMax(*Espr*, *Var*) | *estremoInf* \leq *Var*
 \leq *estremoSup* \Rightarrow valore**

Restituisce un possibile valore calcolato numericamente della variabile *Var* che produce il massimo locale di *Espr*.

Se si fornisce *estremoInf* e *estremoSup*, la funzione controlla nell'ambito dell'intervallo chiuso [*estremoInf*,*estremoSup*] per il massimo locale.

nfMin() (Minimo di una funzione calcolato numericamente)**Catalogo > ****nfMin(*Espr*, *Var*) \Rightarrow valore**nfMin($x^2 + 2 \cdot x + 5, x$)

-1.

nfMin(*Espr*, *Var*, *estremoInf*) \Rightarrow valorenfMin($0.5 \cdot x^3 - x - 2, x, -5, 5$)

-5.

**nfMin(*Espr*, *Var*, *estremoInf*,
estremoSup) \Rightarrow valore**

nfMin() (Minimo di una funzione calcolato numericamente)

Catalogo > 

nfMin(Espr, Var) | $estremoInf \leq Var$
 $\leq estremoSup \Rightarrow valore$

Restituisce un possibile valore calcolato numericamente della variabile *Var* che produce il minimo locale di *Espr*.

Se si fornisce *estremoInf* e *estremoSup*, la funzione controlla nell'ambito dell'intervallo chiuso $[estremoInf, estremoSup]$ per il minimo locale.

nInt() (Integrale numerico)

Catalogo > 

nInt(Espr1, Var, Inferiore, Superiore) \Rightarrow espressione

$$\text{nInt}\left(e^{-x^2}, x, -1, 1\right)$$

1.49365

Se la funzione da integrare *Espr1* non contiene nessun'altra variabile oltre a *Var* e se *Inferiore* e *Superiore* sono costanti, ∞ positivo o ∞ negativo, allora **nInt()** restituisce un'approssimazione di $\int (Espr1, Var, Inferiore, Superiore)$. Tale approssimazione è una media pesata di alcuni valori esemplificativi della funzione da integrare nell'intervallo *Inferiore* $<$ *Var* $<$ *Superiore*.

L'obiettivo sono sei cifre significative. L'algoritmo adattivo termina quando sembra che l'obiettivo sia stato raggiunto, oppure quando sembra che ulteriori esempi non potrebbero portare alcun miglioramento significativo.

Viene visualizzato un avvertimento ("Accuratezza dubbia") quando sembra che l'obiettivo non sia stato raggiunto.

È possibile nidificare **nInt()** per un'integrazione numerica multipla. I limiti di integrazione possono dipendere da variabili di integrazione esterne ad essi.

$$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x\right), x, 0, 1\right) \quad 3.30423$$

nom()**nom(*tassoEffettivo*,*CpY*)**⇒*valore*

Funzione finanziaria che converte il tasso di interesse effettivo annuo *tassoEffettivo* in un tasso nominale, essendo *CpY* il numero di periodi di capitalizzazione per anno.

tassoEffettivo deve essere un numero reale e *CpY* deve essere un numero reale > 0 .

Nota: vedere anche **eff()**, pagina 45.

nom(5.90398,12)

5.75

nor**ctrl** **=** **tasti****BooleanExpr1**nor**BooleanExpr2**
restituisce Boolean expression $x \geq 3 \text{ or } x \geq 4$ $x \geq 3$ $x \geq 3 \text{ nor } x \geq 4$ $x < 3$ **BooleanList1**nor**BooleanList2**
restituisce Boolean list**BooleanMatrix1**nor**BooleanMatrix2**
restituisce Boolean matrix

Restituisce la negazione di un'operazione **or** logica riguardo i due argomenti.
Restituisce true, false o una forma semplificata dell'equazione.

Nel caso di liste e matrici, restituisce un confronto elemento per elemento.

Integer1nor**Integer2**⇒**integer**

Confronta due interi reali bit per bit tramite una operazione **nor**. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati i bit corrispondenti, il risultato sarà 1 se entrambi sono uguali a 1; in caso contrario, il risultato sarà 0. Il valore restituito rappresenta il risultato dei bit e viene visualizzato secondo la modalità base.

3 or 4	7
3 nor 4	-8
{1,2,3} or {3,2,1}	{3,2,3}
{1,2,3} nor {3,2,1}	{-4,-3,-4}

È possibile inserire gli integer in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli integer vengono considerati decimali (base 10)

norm() (Norma di Froebius)

Catalogo > 

norm(*Matrice*) \Rightarrow espressione

norm(*Vettore*) \Rightarrow espressione

Restituisce la norma di Frobenius.

normCdf() (Probabilità di distribuzione

normale)

Catalogo > 

normCdf(*valoreInferiore*,*valoreSuperiore*,
[, μ [, σ]]) \Rightarrow numero se *valoreInferiore* e
valoreSuperiore sono numeri, lista se
valoreInferiore e *valoreSuperiore* sono liste

Calcola la probabilità di distribuzione
normale tra *valoreInferiore* e
valoreSuperiore per μ (default=0) e σ
(default=1) specificati.

normPdf() (Densità di probabilità)

Catalogo > 

normPdf(*ValX*[, μ [, σ]]) \Rightarrow numero se *ValX* è
un numero, lista se *ValX* è una lista

Calcola la funzione della densità di
probabilità per la distribuzione normale in
un valore *valX* specificato per μ σ specificati.

not

Catalogo > 

not *EsprBooleana1* \Rightarrow espressione
booleana

Restituisce vero o falso o una forma
semplificata dell'argomento.

not *Interol* \Rightarrow intero

In modalità base Esadecimale:

Restituisce il complemento a uno di un intero reale. Internamente, *Intero1* viene convertito in numero binario a 64 bit con segno. Il valore di ciascun bit viene scambiato (0 diventa 1 e viceversa) per il complemento a uno. I risultati vengono visualizzati nella modalità Base che è stata impostata.

È possibile inserire l'intero in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, l'intero viene considerato decimale (base 10).

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Per ulteriori informazioni, vedere **►Base2**, pagina 17.

Importante: è zero, non la lettera O.

not 0h7AC36 0hFFFFFFFFFFFF853C9

In modalità base Bin:

Ob100101 ► Base10

37

not 0b100101

not 0b100101▶Base10

Per vedere l'intero risultato, premere ▲,
quindi utilizzare ▲ e ▼ per spostare il cursore.

Nota: un numero binario può contenere fino a 64 cifre (oltre al prefisso 0b). Un numero esadecimale può contenere fino ad 16 cifre.

nPr() (Disposizioni semplici)

Catalogo > [a-z]

nPr(Lista1, Lista2)⇒lista

$$\text{nPr}\{\{5,4,3\},\{2,4,2\}\} = \{20,24,6\}$$

Restituisce una lista delle disposizioni semplici sulla base delle coppie di elementi corrispondenti nelle due liste. Le liste degli argomenti devono avere dimensioni uguali.

nPr(Matrice1, Matrice2)⇒matrice

Restituisce una matrice di disposizioni semplici sulla base delle coppie di elementi corrispondenti nelle due matrici. Le matrici degli argomenti devono avere le stesse dimensioni.

$$nPr \left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \right) = \begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$$

**npv(*tassoInteresse,CFO,CFLista*,
[*CFFreq*])**

Funzione finanziaria che calcola il valore presente netto (Net Present Value), la somma dei valori presenti per i cash flow in entrata (somme ricevute) e in uscita (somme pagate). Un risultato positivo per npv indica un investimento proficuo.

tassoInteresse è il tasso a cui scontare i cash flow (il costo del denaro) di un dato periodo.

CF0 è il cash flow iniziale al tempo 0; deve essere un numero reale.

CFLista è una lista di importi di cash flow dopo il cash flow iniziale *CF0*.

CFFreq è una lista in cui ciascun elemento specifica la frequenza di occorrenza di un importo di cash flow raggruppato (consecutivo), che è l'elemento corrispondente di *CFLista*. L'impostazione predefinita è 1; è possibile inserire valori che siano numeri interi positivi < 10000.

<i>list1</i> := { 6000,-8000,2000,-3000 }	<i>list2</i> := { 6000,-8000,2000,-3000 }
npv(10,5000, <i>list1</i> , <i>list2</i>)	{ 2,2,2,1 }

nSolve() (Soluzione numerica)

**nSolve(*Equazione*,*Var*
[=Campione])**⇒numero o stringa_errore

**nSolve(*Equazione*,*Var*
[=Campione],*estremoInf*)**⇒numero o stringa_errore

**nSolve(*Equazione*,*Var*
[=Campione],*estremoInf*,*estremoSup*)**
⇒numero o stringa_errore

**nSolve(*Equazione*,*Var*[=Campione]) |
estremoInf≤*Var*≤*estremoSup***⇒numero o stringa_errore

nSolve($x^2+5 \cdot x-25=9, x$)	3.84429
nSolve($x^2=4, x=-1$)	-2.
nSolve($x^2=4, x=1$)	2.

Nota: se vi sono soluzioni multiple, è possibile usare un valore campione per trovare una soluzione particolare.

Ricerca iterativamente una soluzione numerica reale approssimata dell'*Equazione* per la sua variabile. Specificare la variabile come:

variabile

– o –

variabile = numero reale

Ad esempio, x è valido come pure x=3.

nSolve() cerca di determinare un punto in cui il resto sia zero oppure due punti relativamente vicini, nei quali il resto abbia segni opposti e la grandezza del resto non sia eccessiva. Se non è possibile ottenere ciò utilizzando un numero limitato di punti campione, viene restituita la stringa "Nessuna soluzione trovata."

$nSolve(x^2+5\cdot x-25=9,x) _{x<0}$	-8.84429
$nSolve\left(\frac{(1+r)^{24}-1}{r}=26,r\right) _{r>0 \text{ and } r<0.25}$	0.006886
$nSolve(x^2=-1,x)$	"No solution found"

O

OneVar [*1..XL*][*Freq*][*Categoria*,*Includi*]]

OneVar [*n..X1,X2,X3,...,X20*]]

Calcola le statistiche ad una variabile fino a 20 liste. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

Gli argomenti di X sono liste di dati.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di codici numerici di categoria dei dati corrispondenti di *X*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Un elemento vuoto (nullo) in qualsiasi lista *X*, *Freq* o *Categoria* produce un corrispondente elemento vuoto in tutte queste liste. Un elemento vuoto (nullo) in qualsiasi lista da *X1* a *X20* produce un corrispondente elemento vuoto in tutte queste liste. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

Variabile di output	Descrizione
stat. \bar{X}	Media dei valori X
Statistiche. Σx	Somma dei valori X
stat. Σx^2	Somma dei valori X^2
stat.sx	Deviazione standard del campione di X
stat. x	Deviazione standard della popolazione di X
stat.n	Numero dei punti di dati
stat.MinX	Minimo dei valori x
stat.Q ₁ X	1° quartile di x
stat.MedianX	Mediana di x
stat.Q ₃ X	3° quartile di x
stat.MaxX	Massimo dei valori x
stat.SSX	Somma dei quadrati delle deviazioni dalla media di x

or

BooleanExpr1 **or** *BooleanExpr2*
restituisce *Boolean expression*

Define $g(x)=$ Func
If $x \leq 0$ or $x \geq 5$

Done

Goto *end*

Return $x \cdot 3$

Lbl *end*

EndFunc

BooleanList1 **or** *BooleanList2* restituisce
Boolean list

BooleanMatrix1 **or** *BooleanMatrix2*
restituisce *Boolean matrix*

$g(3)$

9

$g(0)$

A function did not return a value

Restituisce vero o falso o una forma semplificata dell'espressione immessa originariamente.

Restituisce vero se la semplificazione di una o di entrambe le espressioni risulta vera. Restituisce falso solo se il calcolo di entrambe le espressioni risulta falso.

Nota: vedere xor.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Intero1 or *Intero2* \Rightarrow *intero*

Confronta due interi reali bit per bit tramite un'operazione or. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati bit corrispondenti, il risultato sarà 1 se uno dei due bit è uguale a 1; se entrambi i bit sono uguali a 0, il risultato sarà 0. Il valore restituito rappresenta il risultato dei bit e viene visualizzato nella modalità base che è stata impostata.

È possibile inserire gli interi in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli interi vengono considerati decimali (base 10).

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Per ulteriori informazioni, vedere **Base2**, pagina 17.

Nota: vedere xor.

In modalità base Esadecimale:

0h7AC36 or 0h3D5F 0h7BD7F

Importante: è zero, non la lettera O.

In modalità base Bin:

0b100101 or 0b100 0b100101

Nota: un numero binario può contenere fino a 64 cifre (oltre al prefisso 0b). Un numero esadecimale può contenere fino ad 16 cifre.

ord() Codice numerico di carattere**Catalogo > ****ord(Stringa)⇒intero****ord(Lista)⇒lista**

Restituisce il codice numerico del primo carattere nella stringa di caratteri *Stringa*, oppure una lista dei primi caratteri di ciascun elemento della lista.

ord("hello")	104
char(104)	"h"
ord(char(24))	24
ord({ "alpha", "beta" })	{ 97,98 }

P**P>Rx() (Coordinata x rettangolare)****Catalogo > ****P>Rx(*rEspr*, *θEspr*)⇒espressione**

In modalità angolo in radianti:

P>Rx(*rLista*, *θLista*)⇒lista**P>Rx(*rMatrice*, *θMatrice*)⇒matrice**

Restituisce la coordinata x equivalente della coppia (r, θ).

Nota: l'argomento θ viene interpretato come angolo in gradi, gradianti o radianti, conformemente alla modalità di misurazione degli angoli impostata. Se l'argomento è un'espressione, si può utilizzare il simbolo °, G o † per escludere tale impostazione provvisoriamente.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **P@>Rx** (...).

P>Ry() (Coordinata y rettangolare)**Catalogo > ****P>Ry(*rLista*, *θLista*)⇒lista**

In modalità angolo in radianti:

P>Ry(*rMatrice*, *θMatrice*)⇒matrice

Restituisce la coordinata y equivalente della coppia (r, θ).

Nota: l'argomento θ viene interpretato come angolo in gradi, gradianti o radianti, conformemente alla modalità di misurazione degli angoli impostata.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **Ry** (...).

PassErr**PassErr**

Passa un errore al livello successivo.

Se la variabile di sistema *errCode* è zero, **PassErr** non esegue alcuna azione.

L'istruzione **Else** del blocco **Try...Else...EndTry** dovrebbe utilizzare **ClrErr** o **PassErr**. Se l'errore deve essere elaborato o ignorato, utilizzare **ClrErr**. Se non si sa quale azione applicare all'errore, utilizzare **PassErr** per inviarlo al successivo blocco di gestione degli errori. Se non ci sono ulteriori blocchi di gestione degli errori **Try...Else...EndTry** in attesa di applicazione, la finestra di dialogo dell'errore viene visualizzata come normale.

Nota: Controlla anche **ClrErr**, pagina 24, e **Try**, pagina 166.

Nota per l'inserzione dell'esempio:

Nell'applicazione Calcolatrice sul palmare, è possibile inserire definizioni su più linee premendo **[Esc]** invece di **[enter]** alla fine di ogni riga. Sulla tastiera del computer, mantenere premuto **Alt** e premere **Invio**.

piecewise() (Funzione definita a tratti)

piecewise(*Espr1* [, *Condizione1* [, *Espr2* [, *Condizione2* [, ...]]]])

Restituisce definizioni di una funzione piecewise (definita a tratti) sotto forma di elenco. È inoltre possibile creare definizioni piecewise utilizzando un modello.

Nota: vedere anche **Modello di funzione piecewise** a pagina 2.

Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$	Done
$p(1)$	1
$p(-1)$	undef

poissCdf() (Probabilità cumulata per la distribuzione discreta di Poisson)

Catalogo > 

poissCdf

(

λ

,valoreInferiore,valoreSuperiore) \Rightarrow numero
se valoreInferiore e valoreSuperiore sono numeri, lista se valoreInferiore e valoreSuperiore sono liste

poissCdf(λ ,valoreSuperiore)(per $P(0 \leq X \leq valoreSuperiore)$) \Rightarrow numero se valoreSuperiore è un numero, lista se valoreSuperiore è una lista

Calcola una probabilità cumulata per la distribuzione discreta di Poisson con la media λ specificata.

Per $P(X \leq valoreSuperiore)$, impostare valoreInferiore=0

poissPdf() (Probabilità per la distribuzione discreta di Poisson)

Catalogo > 

poissPdf(λ ,ValX) \Rightarrow numero se ValX è un numero, lista se ValX è una lista

Calcola una probabilità per la distribuzione discreta di Poisson con la media λ specificata.

►Polar (Visualizza come vettore polare)

Catalogo > 

Vettore ►Polar

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Polar.

Visualizza vettore in forma polare [r∠ θ]. Il vettore deve essere bidimensionale e può essere sia una riga che una colonna.

Nota: ►Polar è un'istruzione in formato di visualizzazione, non una funzione di conversione. Tale istruzione può essere impiegata solo alla fine di una riga di introduzione e non aggiorna ans.

►Polar (Visualizza come vettore polare)

Catalogo > 

Nota: vedere anche ►Rect, pagina 129.

valoreComplesso ►Polar

Visualizza vettoreComplesso in forma polare.

- In modalità angolo in gradi, restituisce $(r\angle \theta)$.
- In modalità angolo in radianti, restituisce $re^{i\theta}$.

valoreComplesso può avere una forma complessa qualsiasi. Tuttavia una voce $re^{i\theta}$ causa un errore nella modalità di misurazione degli angoli in gradi.

Nota: è necessario usare le parentesi per un inserimento polare ($r\angle \theta$).

In modalità angolo in radianti:

In modalità angolo in gradi (gradi centesimali):

$(4\angle i)$ ►Polar

$(4\angle 100.)$

In modalità angolo in gradi:

polyEval() (Calcola polinomio)

Catalogo > 

polyEval(Lista1, Espr1)⇒espressione

polyEval(Lista1, Lista2)⇒espressione

Interpreta il primo argomento come coefficiente di un polinomio di grado decrescente e restituisce il polinomio calcolato per il valore del secondo argomento.

polyRoots()

Catalogo > 

polyRoots(Poli,Var)⇒lista

polyRoots(ListaDiCoeff)⇒lista

La prima sintassi, **polyRoots(Poli,Var)**, restituisce una lista di radici reali del polinomio *Poli* in funzione della variabile *Var*. Se non esistono radici reali, restituisce una lista vuota: `{ }`.

La seconda sintassi, **PolyRoots(ListaDiCoeff)**, restituisce una lista di radici reali per i coefficienti di *ListaDiCoeff*.

Nota: vedere anche **cPolyRoots()**, pagina 32.

PowerReg *X, Y [, Freq] [, Categoria, Includi]*

Calcola la regressione su potenza $y = (a \cdot (x)^b)$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot (x)^b$
stat.a, stat.b	Coefficienti di regressione
stat.r ²	Coefficiente di determinazione lineare di dati trasformati
stat.r	Coefficiente di correlazione per dati trasformati ($\ln(x), \ln(y)$)
stat.Resid	Residui associati al modello di potenza
stat.ResidTrans	Residui associati all'adattamento lineare dei dati trasformati
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>

Variabile di output	Descrizione
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

Prgm

Catalogo > 

Prgm
Blocco
EndPrgm

Modello per la creazione di un programma definito dall'utente. Deve essere utilizzato con il comando **Define**, **Define LibPub** o **Define LibPriv**.

Blocco può essere una singola istruzione, una serie di istruzioni separate dal carattere ":" o una serie di istruzioni su righe separate.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Calcolare il massimo comune divisore e visualizzare i risultati intermedi.

Define *proggcd*(*a,b*)=Prgm
 Local *d*
 While *b*≠0
d:=mod(*a,b*)
a:=*b*
b:=*d*
 Disp *a*, " ",*b*
 EndWhile
 Disp "GCD=",*a*
 EndPrgm

Done

proggcd(4560,450)

450 60

60 30

30 0

GCD=30

Done

prodSeq()

Vedere **P()**, pagina 194.

Product (P!) (Prodotto)

Vedere **P()**, pagina 194.

product() (Prodotto)

Catalogo > 

product(*Lista*[, *Inizio*[, *Fine*]])⇒*espressione*

Restituisce il prodotto degli elementi contenuti in *Lista*. *Inizio* e *Fine* sono opzionali e specificano un intervallo di elementi.

product(*Matrice1*[, *Inizio*[, *Fine*]]) \Rightarrow *matrice*

Restituisce un vettore di riga contenente i prodotti degli elementi nelle colonne di *Matrice1*. *Inizio* e *Fine* sono opzionali e specificano un intervallo di elementi.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

$$\text{product} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad [28 \ 80 \ 162]$$

$$\text{product} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}_{1,2} \quad [4 \ 10 \ 18]$$

propFrac(*numero_razionale*) restituisce *numero_razionale* sotto forma di somma di un numero intero e di una frazione, aventi lo stesso segno e denominatore di grandezza maggiore del numeratore.

propFrac(*espressione_razionale*, *Var*) restituisce la somma delle frazioni proprie ed un polinomio rispetto a *Var*. Il grado di *Var* nel denominatore supera il grado di *Var* nel numeratore di ciascuna frazione propria. Le potenze simili di *Var* sono ridotte. I termini ed i rispettivi fattori sono ordinati considerando *Var* la variabile principale.

Se *Var* è omesso, viene eseguita l'espansione della frazione propria rispetto alla variabile più significativa. I coefficienti della parte polinomiale vengono trasformati in propri rispetto alla prima variabile più significativa, e così di seguito.

$$\text{propFrac} \begin{pmatrix} 4 \\ 3 \end{pmatrix} \quad 1 \frac{1}{3}$$

$$\text{propFrac} \begin{pmatrix} -4 \\ 3 \end{pmatrix} \quad -1 \frac{1}{3}$$

$$\text{propFrac} \begin{pmatrix} x^2+x+1 \\ x+1 \end{pmatrix} + \frac{y^2+y+1}{y+1}, x \quad \frac{1}{x+1} + x + \frac{y^2+y+1}{y+1}$$

$$\text{propFrac}(\text{Ans}) \quad \frac{1}{x+1} + x + \frac{1}{y+1} + y$$

QR (Scomposizione QR)

Catalogo >

QR Matrice, MatriceQ, MatriceR[, Tol]

Calcola la scomposizione QR di Householder di una matrice reale o complessa. Le matrici Q ed R che si ottengono vengono memorizzate nei *Matrice* specificati. La matrice Q è unitaria. La matrice R è triangolare superiore.

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tol*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tol* viene ignorato.

- Se si usa **ctrl enter** oppure se si imposta la modalità **Auto o Approssimato** su Approssimato, i calcoli verranno eseguiti in virgola mobile.
- Se *Tol* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:
 $5E-14 \cdot \max(\dim(\text{Matrice})) \cdot \text{rowNorm}(\text{Matrice})$

La scomposizione QR viene calcolata in modo numerico tramite trasformazioni di Householder; la soluzione simbolica tramite Gram-Schmidt. Le colonne in *nomeMatriceQ* sono i vettori della base ortonormale con estensione pari allo spazio definito da *matrice*.

Il numero a virgola mobile (9.) in *m1* fa sì che i risultati vengano calcolati nella forma a virgola mobile.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix}$
QR <i>m1,qm,rm</i>	<i>Done</i>
<i>qm</i>	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
QR <i>m1,qm,rm</i>	<i>Done</i>
<i>qm</i>	$\begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} \frac{\sqrt{m^2+o^2}}{m} & \frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{ m \cdot p - n \cdot o }{\sqrt{m^2+o^2}} \end{bmatrix}$

QuadReg *X, Y [, Freq] [, Categoria, Includi]*

Calcola la regressione polinomiale quadraticay = a · x^2 +b · x+c sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: a · x^2 +b · x+c
stat.a, stat.b, stat.c	Coefficienti di regressione
stat.R ²	coefficiente di determinazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

QuartReg *X, Y [, Freq] [, Categoria, Includi]*]]

Calcola la regressione polinomiale quartica $y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di *X* e *Y*. Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Coefficienti di regressione
stat.R ²	coefficiente di determinazione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categorie</i>

Variabile di output	Descrizione
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoría</i> e <i>Includi Categoría</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

R

R►Pθ()

Catalogo > 

R►Pθ (*listaX, listaY*) \Rightarrow *lista*
R►Pθ (*matriceX, matriceY*) \Rightarrow *matrice*

Restituisce la coordinata θ equivalente della coppia di argomenti (x,y) .

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradi centesimali o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **R@>Ptheta** (...).

In modalità angolo in gradi:

In modalità angolo in gradi centesimali:

In modalità angolo in radianti:

R►Pr()

Catalogo > 

R►Pr (*listaX, listaY*) \Rightarrow *lista*
R►Pr (*matriceX, matriceY*) \Rightarrow *matrice*

Restituisce la coordinata r equivalente alla coppia di argomenti (x,y) .

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **R@>Pr** (...).

In modalità angolo in radianti:

► Rad

Catalogo > 

Converte l'argomento in una misura d'angolo in radianti.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando **@>Rad**.

In modalità angolo in gradi:

(1.5)►Rad $\quad (0.02618)^r$

In modalità angolo in gradi centesimali:

(1.5) ► Rad

(0.023562)^r**rand() (Numero casuale)**Catalogo > 

rand() \Rightarrow espressione
rand(*numTentativi*) \Rightarrow lista

rand() restituisce un numero casuale compreso tra 0 e 1.

rand(*numTentativi*) restituisce una lista contenente *numTentativi* valori casuali compresi tra 0 e 1.

Imposta il seme dei numeri casuali.

RandSeed 1147	Done
rand(2)	{ 0.158206, 0.717917 }

randBin() (Numero casuale da distribuzione binomiale)Catalogo > 

randBin(*n, p*) \Rightarrow espressione
randBin(*n, p, numTentativi*) \Rightarrow lista

randBin(*n, p*) restituisce un numero reale casuale da una distribuzione binomiale specificata.

randBin(*n, p, numTentativi*) restituisce una lista contenente *numTentativi* numeri reali casuali da una distribuzione binomiale specificata.

randInt() (Intero casuale)Catalogo > 

```
randInt
(
estremoInf
,estremoSup)  $\Rightarrow$ 
espressione
randInt
(
estremoInf
,estremoSup
, numTentativi)  $\Rightarrow$ 
lista
```

randInt() (Intero casuale)

Catalogo > 

randInt
(
estremoInf,
estremoSup)
restituisce un intero casuale all'interno dell'intervallo specificato dai numeri interi *estremoInf* e *estremoSup*.

randInt
(
estremoInf,
,
estremoSup,
numTentativi)
restituisce un elenco contenente *numTentativi* interi casuali all'interno dell'intervallo specificato.

randMat() (Matrice casuale)

Catalogo > 

randMat(*numRighe*, *numColonne*) \Rightarrow matrice

Restituisce una matrice di numeri interi compresi tra -9 e 9 della dimensione specificata.

Entrambi gli argomenti devono potere essere semplificati in numeri interi.

RandSeed 1147	Done
randMat(3,3)	$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$

Nota: i valori di questa matrice cambiano ogni volta che si preme **enter**.

randNorm() (Normale casuale)

Catalogo > 

randNorm(μ , σ) \Rightarrow espressione
randNorm(μ , σ , *numTentativi*) \Rightarrow lista

RandSeed 1147	Done
randNorm(0,1)	0.492541
randNorm(3,4.5)	-3.54356

randNorm() (Normale casuale)**Catalogo >** 

randNorm(μ, σ) restituisce un numero decimale dalla distribuzione normale specificata. Può essere qualsiasi numero reale, anche se con maggiore probabilità sarà compreso nell'intervallo $[\mu-3\sigma, \mu+3\sigma]$.

randNorm($\mu, \sigma, numTentativi$) restituisce una lista contenente $numTentativi$ valori decimali dalla distribuzione normale specificata.

randPoly() (Polinomio casuale)**Catalogo >** 

randPoly(*Var, Ordine*) \Rightarrow *espressione*

Restituisce un polinomio in *Var* dell'*Ordine* specificato. I coefficienti sono interi casuali compresi tra -9 e 9 . Il primo coefficiente non può essere zero.

Ordine deve essere tra 0 e 99 .

RandSeed 1147	Done
randPoly($x, 5$)	$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$

randSamp() (Campione casuale)**Catalogo >** 

randSamp(*Lista, numTentativi* [,*noSostituzione*]) \Rightarrow *lista*

Restituisce una lista contenente un campione casuale di *numTentativi* da *Lista* con l'opzione di sostituire il campione (*noSostituzione*=0) o meno (*noSostituzione*=1). L'impostazione predefinita prevede la sostituzione del campione.

RandSeed (Seme numero casuale)**Catalogo >** 

RandSeed *Numero*

Se *Numero* = 0 , imposta i semi ai valori predefiniti per il generatore di numeri casuali. Se *Numero* $\neq 0$, viene utilizzato per generare due semi, memorizzati nelle variabili di sistema *seed1* e *seed2*.

RandSeed 1147	Done
rand()	0.158206

Restituisce la parte reale dell'argomento.

real(Lista1) \Rightarrow lista

Restituisce le parti reali di tutti gli elementi.

real(Matrice1) \Rightarrow matrice

Restituisce le parti reali di tutti gli elementi.

► Rect

Vettore ► Rect

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Rect.

Visualizza *Vettore* nella forma rettangolare [x, y, z]. Le dimensioni del vettore devono essere 2 o 3 e il vettore può essere una riga o una colonna.

Nota: ► Rect è un'istruzione del formato di visualizzazione, non una funzione di conversione. Tale istruzione può essere impiegata solo alla fine di una riga di introduzione e non aggiorna ans.

Nota: Vedere anche ► Polar, pagina 117.

valoreComplesso ► Rect

In modalità angolo in radianti:

Visualizza *valoreComplesso* nella forma rettangolare a+bi. *valoreComplesso* può avere una forma complessa qualsiasi. Tuttavia una voce $rei\theta$ causa un errore nella modalità di misurazione degli angoli in gradi.

Nota: è necessario usare le parentesi per un inserimento polare ($r\angle \theta$).

In modalità angolo in gradi centesimali:

$\langle (1 \angle 100) \rangle$ ► Rect

i

In modalità angolo in gradi:

Nota: per inserire \angle dalla tastiera, selezionarlo nell'elenco dei simboli del Catalogo.

ref(*Matrice1*[, *Tol*]) \Rightarrow *matrice*

Restituisce la forma a scalini per righe di *Matrice1*.

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tol*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tol* viene ignorato.

$$\text{ref} \begin{pmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{pmatrix} \begin{pmatrix} 1 & -2 & -4 & 4 \\ 0 & 5 & 5 & 5 \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{pmatrix}$$

- Se si usa **ctrl enter** oppure se si imposta la modalità **Auto o Approssimato** su Approssimato, i calcoli saranno eseguiti in virgola mobile.
- Se *Tol* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:
 $5E-14 \cdot \max(\dim(\text{Matrice1}))$
 $\cdot \text{rowNorm}(\text{Matrice1})$

Evitare elementi indefiniti in *Matrice1*. Possono produrre risultati imprevisti.

Ad esempio, se *a* è indefinito nella seguente espressione, viene visualizzato un messaggio di avvertenza e il risultato viene mostrato come:

$$\text{ref} \begin{pmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Il messaggio di avvertenza appare perché l'elemento generalizzato $1/a$ non è valido per $a=0$.

È possibile evitare questa situazione memorizzando in anticipo un valore in *a* oppure utilizzando l'operatore di vincolo ("|"), come mostrato nell'esempio che segue.

$$\text{ref}\left[\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right] | a=0 \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Nota: vedere anche **rref()**, page 140.

RefreshProbeVars

RefreshProbeVars

Permette di accedere ai dati di tutti i sensori collegati in un programma TI-Basic.

Valore di varStato	Stato
varStato =0	Normale (continuare l'esecuzione del programma) L'applicazione Vernier DataQuest™ è in modalità di raccolta dati.
varStato =1	Nota: affinché questo comando funzioni, l'applicazione Vernier DataQuest™ deve essere in  modalità misura.
varStato =2	L'applicazione Vernier DataQuest™ non è stata avviata.
varStato =3	L'applicazione Vernier DataQuest™ è stata avviata, ma non sono presenti sensori collegati.

Esempio

```
Define temp()=
Prgm
  © Verificare se il sistema è
  pronto
  RefreshProbeVars stato
  If stato=0 Then
    Disp "pronto"
    For n,1,50
      RefreshProbeVars stato
      temperatura:=meter.temperatura
      Disp "Temperatura: ",temperatura
      If temperatura>30 Then
        Disp "Caldo eccessivo"
        EndIf
      © Attendere 1 secondo tra i
      campioni
      Wait 1
    EndFor
  Else
    Disp "Non pronto. Riprovare più
    tardi"
```

EndIf

EndPrgm

Nota: Può essere utilizzato anche con l'hub di TI-Innovator™.

remain() (Resto)Catalogo > 

remain(Lista1, Lista2) \Rightarrow lista
remain(Matrice1, Matrice2) \Rightarrow matrice

Restituisce il resto del primo argomento rispetto al secondo argomento, come definito dalle identità:

remain(x,0) \Rightarrow x
remain(x,y) \Rightarrow x - y * iPart(x/y)

Si noti che, di conseguenza **remain(-x,y) = -remain(x,y)**. Il risultato può essere zero oppure può avere lo stesso segno del primo argomento.

Nota: vedere anche **mod()**, pagina 100.

remain(7,0)	7
remain(7,3)	1
remain(-7,3)	-1
remain(7,-3)	1
remain(-7,-3)	-1
remain({12,-14,16},{9,7,-5})	{3,0,1}

$$\text{remain} \begin{bmatrix} 9 & -7 \\ 6 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 3 \\ 4 & -3 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$$

RequestCatalogo > 

Request stringaPrompt, var[, FlagDisp [, varState]]

Request promptString, funz(arg1, ...argn) [, FlagDisp [, statusVar]]

Istruzione di programmazione: Sospende il programma e visualizza una finestra di dialogo contenente il messaggio *stringaPrompt* e un riquadro di testo per la risposta dell'utente.

Dopo che l'utente ha digitato una risposta e ha fatto clic su **OK**, il contenuto del riquadro di testo viene assegnato alla variabile *var*.

Se l'utente fa clic su **Cancel** (Annulla), il programma procede senza accettare input. Il programma utilizza il precedente valore di *var* se *var* è già stata definita.

Definire un programma:

```
Define request_demo()=Prgm
  Request "Radius: ",r
  Disp "Area = ",pi*r^2
EndPrgm
```

Eseguire il programma e digitare una risposta:

request_demo()



Risultato dopo aver selezionato **OK**:

L'argomento opzionale *FlagDispl* può essere qualsiasi espressione.

- Se *FlagDispl* viene omesso o dà come risultato **1**, la richiesta e la risposta da parte dell'utente vengono visualizzate nella cronologia della calcolatrice.
- Se *FlagDispl* dà come risultato **0**, la richiesta e la risposta non vengono visualizzate nella cronologia.

L'argomento *varStato* opzionale offre al programma la possibilità di determinare la modalità di uscita dell'utente dalla finestra di dialogo. Notare che *varStato* richiede l'argomento *FlagDispl*.

- Se l'utente ha fatto clic su **OK** o ha premuto **Invio** oppure **Ctrl+Invio**, la variabile *varStato* viene impostata sul valore **1**.
- Altrimenti, la variabile *varStato* viene impostata sul valore **0**.

L'argomento *funz()* consente a un programma di memorizzare la risposta dell'utente come una definizione di funzione. Questa sintassi opera come se l'utente avesse eseguito il comando:

Define *funz(arg1, ...argn) = risposta utente*

Il programma può quindi usare la funzione definita *funz()*. La *stringaPrompt* dovrebbe portare l'utente ad inserire una *risposta utente* appropriata che completa la definizione di funzione.

Nota: è possibile utilizzare il comando **Request** all'interno di un programma definito dall'utente, ma non di una funzione.

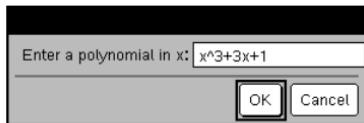
Per arrestare un programma che contiene un comando **Request** in un loop infinito:

Semiretta: 6/2
Area = 28.2743

Definire un programma:

```
Define polynomial()=Prgm
  Request "Enter a polynomial in
x:",p(x)
  Disp "Real roots are:",polyRoots(p
(x),x)
EndPrgm
```

Eseguire il programma e digitare una risposta:
polynomial()



Risultato dopo avere inserito x^3+3x+1 e selezionato **OK**:

Le radici reali sono: $\{-0.322185\}$

- **Palmare:** Tenere premuto il tasto  **on** e premere **enter** più volte.
- **Windows®:** Tenere premuto il tasto **F12** e premere **Invio** più volte.
- **Macintosh®:** Tenere premuto il tasto **F5** e premere **Invio** più volte.
- **iPad®:** L'app mostra un prompt. È possibile continuare ad attendere o annullare.

Nota: vedere anche **RequestStr**, page 134.

RequestStr

RequestStr *stringaPrompt, var[, DispFlag]*

Istruzione di programmazione: Opera in modo identico alla prima sintassi del comando **Request**, eccetto che la risposta dell'utente viene sempre interpretata come stringa. A differenza di questo, il comando **Request** interpreta la risposta come espressione a meno che l'utente non la racchiuda tra virgolette ("").

Nota: è possibile utilizzare il comando **RequestStr** all'interno di un programma definito dall'utente, ma non di una funzione.

Per arrestare un programma che contiene un comando **RequestStr** in un loop infinito:

- **Palmare:** Tenere premuto il tasto  **on** e premere **enter** più volte.
- **Windows®:** Tenere premuto il tasto **F12** e premere **Invio** più volte.
- **Macintosh®:** Tenere premuto il tasto **F5** e premere **Invio** più volte.
- **iPad®:** L'app mostra un prompt. È possibile continuare ad attendere o annullare.

Definire un programma:

```
Define requestStr_demo()=Prgm
  RequestStr "Your name:",name,0
  Disp "Response has ",dim(name)," characters."
EndPrgm
```

Eseguire il programma e digitare una risposta:

```
requestStr_demo()
```



Risultato dopo aver selezionato **OK** (si noti che l'argomento *FlagDisp* 0 omette la richiesta e la risposta dalla cronologia):

```
requestStr_demo()
```

La risposta ha 5 caratteri.

Nota: vedere anche **Request**, page 132.

Return

Return [Espr]

Restituisce *Espr* quale risultato della funzione. Questo comando viene utilizzato all'interno di un blocco **Func...EndFunc**.

Nota: utilizzare **Return** senza alcun argomento all'interno di un blocco **Prgm...EndPrgm** per uscire da un programma.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

```
Define factorial (nn)=
Func
Local answer,counter
1->answer
For counter,1,nn
answer· counter-> answer
EndFor
Return answer
EndFunc
```

factorial (3)

6

right() (Destra)

right(Lista1[, Num]) ⇒ lista

right({1,3,-2,4},3)

{3,2,4}

Restituisce i *Num* elementi più a destra contenuti in *Lista1*.

Se si omette *Num*, restituisce l'intera *Lista1*.

right(stringaOrigine[, Num]) ⇒ stringa

right("Hello",2)

"lo"

Restituisce i caratteri *Num* più a destra contenuti nella stringa di caratteri *stringaOrigine*.

Se si omette *Num*, restituisce l'intera *stringaOrigine*.

right(Confronto) ⇒ espressione

right(x<3)

3

Restituisce il secondo membro di un'equazione o di una disequazione.

rk23 ()

rk23(Espr, Var, varDipendente, {Var0,

Equazione differenziale:

VarMax}, varDipendente0, incrVar [, tollErrore]) \Rightarrow matrice

rk23(*SistemaDiEspr, Var, ListaDiVarDipendenti, {Var0, VarMax}, ListaDiVarDipendenti0, incrVar[, tollErrore])* \Rightarrow matrice

rk23(*ListaDiEspr, Var, ListaDiVarDipendenti, {Var0, VarMax}, ListaDiVarDipendenti0, incrVar[, tollErrore])* \Rightarrow matrice

Utilizza il metodo di Runge-Kutta per risolvere il sistema

$$\frac{d \text{ depVar}}{d \text{ Var}} = \text{Expr}(\text{Var}, \text{depVar})$$

con *varDipendente*

(Var0)=varDipendente0 nell'intervallo *[Var0,VarMax]*. Restituisce una matrice la cui prima riga definisce i valori ottenuti di *Var* come definito da *incrVar*. La seconda riga definisce il valore del primo componente della soluzione per i valori di *Var* corrispondenti, e così via.

Espr è il secondo membro che definisce l'equazione differenziale ordinaria (ODE).

SistemaDiEspr è un sistema di secondi membri che definiscono il sistema di ODE (corrisponde all'ordine di variabili dipendenti in *ListaDiVarDipendenti*).

ListaDiEspr è una lista di secondi membri che definisce il sistema di ODE (corrisponde all'ordine di variabili dipendenti in *ListaDiVarDipendenti*).

Var è la variabile indipendente.

ListaDiVarDipendenti è una lista di variabili dipendenti.

{Var0, VarMax} è una lista a due elementi che indica alla funzione di integrare da *Var0* a *VarMax*.

*y=0.001*y*(100-y) e y(0)=10*

rk23(*0.001·y·(100-y),t,y,{0,100},10,1*)

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9367 & 11.9493 & 13.042 & 14.2 \end{bmatrix}$$

Per vedere l'intero risultato, premere \blacktriangleleft , quindi utilizzare \blacktriangleleft e \blacktriangleright per spostare il cursore.

Stessa equazione con *tollErrore* impostata su **1.E-6**

rk23(*0.001·y·(100-y),t,y,{0,100},10,1,1.E-6*)

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9367 & 11.9495 & 13.0423 & 14.2189 \end{bmatrix}$$

Sistema di equazioni

$$\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$

con *y1(0)=2* e *y2(0)=5*

rk23(*{y1'+0.1·y1·y2, y2'+3·y2-y1·y2},t,{y1,y2},{0,5},{2,5},1*)

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 2. & 1.94103 & 4.78694 & 3.25253 & 1.82848 \\ 5. & 16.8311 & 12.3133 & 3.51112 & 6.27245 \end{bmatrix}$$

ListaDiVarDipendenti0 è una lista di valori iniziali di variabili dipendenti.

Se $incrVar$ dà come risultato un numero diverso da zero: $\text{sign}(incrVar) = \text{sign}(VarMax-Var0)$ e sono restituite soluzioni a $Var0+i*incrVar$ per tutti i valori di $i=0,1,2,\dots$ tali che $Var0+i*incrVar$ sia in $[var0,VarMax]$ (potrebbe non esserci un valore di soluzione a $VarMax$).

Se *incrVar* dà come risultato zero, le soluzioni sono restituite ai valori *Var* di "Runge-Kutta".

tollErrore è la tolleranza d'errore (predefinita a 0.001).

root() (Radice)

Nota: Vedere anche **Modello di radice ennesima a pagina, pagina 1.**

rotate() (Ruota)

rotate(*Interol*[, *numRotazioni*]) \Rightarrow *intero*

Ruota i bit di un numero intero binario. È possibile inserire *Interol* in qualsiasi base numerica; esso viene convertito automaticamente in forma binaria a 64 bit con segno. Se *Interol* è troppo grande per questa forma, un'operazione a modulo simmetrico lo porta all'interno dell'intervallo. Per ulteriori informazioni, vedere ► **Base2**, pagina 17.

Se *numRotazioni* è positivo, la rotazione avviene verso sinistra. Se *numRotazioni* è negativo, la rotazione avviene verso destra. L'impostazione predefinita è -1 (ruota a destra di un bit).

Ad esempio, in una rotazione a destra:

Ogni bit ruota verso destra.

0b00000000000001111010110000110101

In modalità base Bin:

Per vedere l'intero risultato, premere ▲,
quindi utilizzare ▲ e ▼ per spostare il
cursore.

In modalità base Esadecimale:

rotate(0h78E)	0h3C7
rotate(0h78E, -2)	0h8000000000000001E3
rotate(0h78E,2)	0h1E38

Importante: per inserire un numero binario o esadecimale, utilizzare sempre il prefisso 0b o 0h (è uno zero, non la lettera O).

L'ultimo bit a destra diventa il primo a sinistra.

dà:

0b100000000000000111101011000011010

I risultati vengono visualizzati nella modalità Base che è stata impostata.

rotate(Lista1[,numRotazioni]) ⇒ *lista*

Restituisce una copia di *Lista1* ruotata a destra o a sinistra di *numRotazioni* elementi. Non modifica *Lista1*.

Se *numRotazioni* è positivo, la rotazione avviene verso sinistra. Se *numRotazioni* è negativo, la rotazione avviene verso destra. L'impostazione predefinita è -1 (ruota a destra di un bit).

rotate(Stringa1[,numRotazioni]) ⇒ *stringa*

Restituisce una copia di *Stringa1* ruotata a destra o a sinistra di *numRotazioni* caratteri. Non modifica *Stringa1*.

Se *numRotazioni* è positivo, la rotazione avviene verso sinistra. Se *numRotazioni* è negativo, la rotazione avviene verso destra. L'impostazione predefinita è -1 (ruota a destra di un carattere).

In modalità base Dec:

rotate({1,2,3,4})	{4,1,2,3}
rotate({1,2,3,4},-2)	{3,4,1,2}
rotate({1,2,3,4},1)	{2,3,4,1}

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"bcda"

Restituisce l'argomento arrotondato ad un numero specifico di cifre dopo la virgola decimale.

cifre deve essere un numero intero compreso tra 0 e 12. Se *cifre* non è incluso, restituisce l'argomento arrotondato alle prime 12 cifre significative.

Nota: la visualizzazione dipende dalla modalità selezionata.

round(1.234567,3)

1.235

round() (Arrotondamento)**Catalogo > ****round(Lista1[, cifre])** \Rightarrow lista

Restituisce una lista degli elementi arrotondati al numero specifico di cifre.

round(Matrice1[, cifre]) \Rightarrow matrice

Restituisce una matrice degli elementi arrotondati al numero specifico di cifre.

round($\{\pi, \sqrt{2}, \ln(2)\}$, 4) $\{3.1416, 1.4142, 0.6931\}$ round($\begin{bmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{bmatrix}$, 1) $\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$ **rowAdd() (Somma di righe di matrice)****Catalogo > ****rowAdd(Matrice1, rIndice1, rIndice2)** \Rightarrow matriceRestituisce una copia di *Matrice1* nella quale la riga *rIndice2* è sostituita dalla somma delle righe *rIndice1* e *rIndice2*.**rowDim() (Dimensione righe matrice)****Catalogo > ****rowDim(Matrice)** \Rightarrow espressioneRestituisce il numero di righe di *Matrice*.**Nota:** vedere anche **colDim()**, pagina 25. $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1$ $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ rowDim(*m1*)

3

rowNorm() (Norma righe matrice)**Catalogo > ****rowNorm(Matrice)** \Rightarrow espressioneRestituisce il massimo delle somme dei valori assoluti degli elementi nelle righe di *Matrice*.**Nota:** tutti gli elementi della matrice devono potere essere semplificati in numeri. Vedere anche **colNorm()**, pagina 25.rowNorm($\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix}$)

25

rowSwap() (Inverti righe matrice)**Catalogo > ****rowSwap(*Matrice1*, *rIndice1*, *rIndice2*)**
 \Rightarrow matriceRestituisce *Matrice1* con le righe *rIndice1* e *rIndice2* scambiate.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
rowSwap(<i>mat</i> , 1, 3)	$\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$

rref() (Forma a scalini ridotta per righe)**Catalogo > ****rref(*Matrice1* [, *Tol*])** \Rightarrow matriceRestituisce la forma a scalini ridotta per righe di *Matrice1*.

$\text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$
--	---

 $\text{rref}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
--	--

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tol*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tol* viene ignorato.

- Se si usa **ctrl enter** oppure se si imposta la modalità **Auto o Approssimato** su Approssimato, i calcoli saranno eseguiti in virgola mobile.
- Se *Tol* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come:
5E-14 •max(dim(*Matrice1*))
•rowNorm(*Matrice1*)

Nota: vedere anche **ref()**, page 130.**S****sec() (Secante)****Tasto ****sec(*Lista1*)** \Rightarrow lista

In modalità angolo in gradi:

sec() (Secante)

Tasto 

Nota: conformemente alla modalità di misurazione degli angoli impostata, l'argomento viene interpretato come angolo in gradi, gradianti o radianti. È possibile utilizzare °, G o r per escludere provvisoriamente la modalità d'angolo selezionata.

sec⁻¹() (Secante inversa)

Tasto 

$\text{sec}^{-1}(\text{Lista1}) \Rightarrow \text{lista}$

In modalità angolo in gradi:

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradianti o radianti.

In modalità angolo in gradi (gradi centesimali):

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arcsec** (...).

In modalità angolo in radianti:

sech() (Secante iperbolica)

Catalogo > 

$\text{sech}(\text{Lista1}) \Rightarrow \text{lista}$

sech⁻¹() (Secante iperbolica inversa)

Catalogo > 

$\text{sech}^{-1}(\text{Lista1}) \Rightarrow \text{lista}$

In modalità angolo in radianti e in modalità formato rettangolare complesso:

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arcsech** (...).

Send

Menu Hub

Send *exprOrString1[, exprOrString2] ...*

Esempio: attivare l'elemento blu del LED RGB integrato per 0,5 secondi.

Send "SET COLOR.BLUE ON TIME .5"

Done

Istruzione di programmazione: invia uno o più TI-Innovator™ Hub comandi a un hub collegato.

exprOrString deve essere un TI-Innovator™ Hub comando valido. Normalmente, *exprOrString* contiene un comando "SET ..." per controllare un dispositivo o un comando "READ ..." per richiedere i dati.

Esempio: richiedere il valore attuale del sensore di livello luminosità integrato dell'hub. Un comando **Get** recupera il valore e lo assegna alla variabile *lightval*.

Send

Menu Hub

Gli argomenti vengono inviati in successione all'hub.

Nota: è possibile utilizzare il comando **Send** all'interno di un programma definito dall'utente ma non di una funzione.

Nota: vedere anche **Get** (pagina 62), **GetStr** (pagina 69) e **eval()** (pagina 49).

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	0.347922

Esempio: inviare una frequenza calcolata al diffusore integrato dell'hub. Utilizzare la variabile speciale *iostr.SendAns* per mostrare il comando hub con l'espressione valutata.

<i>n</i> :=50	50
<i>m</i> :=4	4
Send "SET SOUND eval(<i>m</i> · <i>n</i>)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

seq() (Sequenza)

Catalogo >

seq(*Espr*, *Var*, *Basso*, *Alto*[, *Incr*]) \Rightarrow *lista*

Aumenta *Var* da *Basso* a *Alto* con incrementi di *Incr*, calcola *Espr* e restituisce i risultati in forma di lista. Il contenuto originale di *Var* è intatto dopo l'esecuzione di **seq()**.

Il valore predefinito di *Incr* è 1.

$\text{seq}\left(n^2, n, 1, 6\right)$	$\{1, 4, 9, 16, 25, 36\}$
$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$	$\left\{\frac{1}{1}, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	$\frac{1968329}{1270080}$

Nota: Per forzare un risultato approssimativo,

Palmare: Premere **ctrl** **enter**.

Windows®: Premere **Ctrl+Invio**.

Macintosh®: Premere **⌘+Invio**.

iPad®: Tenere premuto **Invio** e selezionare **≈**

$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	1.54977
--	---------

seqGen()

Catalogo >

seqGen(*Espr*, *Var*, *varDipendente*, {*Var0*, *VarMax*}[, *ListaDiTermIniziali* [, *incrVar* [, *ValArrotPerEccesso*]]]) \Rightarrow *lista*

Generare i primi 5 termini della sequenza $u(n) = u(n-1)^2/2$, con $u(1)=2$ e $incrVar=1$.

seqGen()

Genera una lista di termini per la successione $varDipendente(Var)=Espr$ come segue: Incrementa la variabile indipendente Var da $Var0$ fino a $VarMax$ di $incrVar$, calcola $varDipendente(Var)$ per i valori corrispondenti di Var utilizzando la formula $Espr$ e $ListaDiTermIniziali$ e restituisce i risultati in forma di lista.

seqGen{*ListaOSistemaDiEspr*, *Var*, *ListaDiVarDipendenti*, {*Var0*, *VarMax* [, *MatriceDiTermIniziali* [, *incrVar* [, *ValArrotPerEccesso*]]]} \Rightarrow *matrice*

Genera una matrice di termini per un sistema (o lista) di successioni *ListaDiVarDipendenti* (*Var*)=*ListaOSistemaDiEspr* come segue: Incrementa la variabile indipendente Var da $Var0$ fino a $VarMax$ di $incrVar$, calcola *ListaDiVarDipendenti*(*Var*) per i valori corrispondenti di *Var* utilizzando la formula *ListaOSistemaDiEspr* e *MatriceDiTermIniziali* e restituisce i risultati in forma di matrice.

Il contenuto originale di *Var* è intatto dopo l'esecuzione di **seqGen()**.

Il valore predefinito di *incrVar* è 1.

$$\begin{aligned} & \text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right) \\ & \left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\} \end{aligned}$$

Esempio in cui $Var0=2$:

$$\begin{aligned} & \text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right) \\ & \left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\} \end{aligned}$$

Sistema di due successioni:

$$\begin{aligned} & \text{seqGen}\left[\left\{\frac{1}{n}, \frac{u2(n-1)}{2} + u1(n-1)\right\}, n, \{u1, u2\}, \{1, 5\}\right] \begin{bmatrix} _ \\ 2 \end{bmatrix} \\ & \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix} \end{aligned}$$

Nota: il simbolo di nullo (_) nella matrice di termini iniziali sopra viene utilizzato per indicare che il termine iniziale per $u1(n)$ è calcolato utilizzando la formula di sequenza esplicita $u1(n)=1/n$.

seqn()

seqn{*Espr*{*u*, *n* [, *ListaDiTermIniziali* [, *nMax* [, *ValArrotPerEccesso*]]]} \Rightarrow *lista*

Genera una lista di termini per una successione $u(n)=Espr(u, n)$ come segue: incrementa *n* da 1 a *nMax* di 1, calcola *u*(*n*) per i valori di *n* corrispondenti utilizzando la formula *Espr*(*u*, *n*) e *ListaDiTermIniziali* e restituisce i risultati in forma di lista.

seqn{*Espr*{*n* [, *nMax* [, *ValArrotPerEccesso*]]]} \Rightarrow *lista*

Genera i primi 6 termini della successione $u(n)=u(n-1)/2$, con $u(1)=2$.

$$\begin{aligned} & \text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right) \\ & \left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\} \end{aligned}$$

$$\begin{aligned} & \text{seqn}\left(\frac{1}{n^2}, 6\right) \\ & \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\} \end{aligned}$$

Genera una lista di termini per una successione non ricorsiva $u(n)=\text{Espr}(n)$ come segue: Incrementa n da 1 a $nMax$ di 1, calcola $u(n)$ per i valori di n corrispondenti utilizzando la formula $\text{Espr}(n)$ e restituisce i risultati in forma di lista.

Se manca $nMax$, $nMax$ viene impostata su 2500

Se $nMax=0$, $nMax$ viene impostato su 2500

Nota: `seqn()` chiama `seqGen()` con $n0=1$ e $nincr=1$

setMode()

**setMode(*interoNomeModo*,
interoImpostazione)** \Rightarrow *intero*

setMode(*lista*) \Rightarrow *lista interi*

Valido solo all'interno di una funzione o di un programma.

**setMode(*interoNomeModo*,
interoImpostazione)** imposta temporaneamente il modo *interoNomeModo* sulla nuova impostazione *interoImpostazione* e restituisce un intero corrispondente all'impostazione originale di quel modo. La modifica è limitata alla durata dell'esecuzione del programma o della funzione.

interoNomeModo specifica il modo da impostare. Deve essere uno degli interi dei modi della tabella seguente.

interoImpostazione specifica la nuova impostazione per il modo. Deve essere uno dei numeri interi per le impostazioni elencati di seguito per lo specifico modo che si sta impostando.

Visualizzare il valore approssimato di π utilizzando l'impostazione predefinita di Mostra cifre (Display Digits), quindi visualizzare π con l'impostazione Fissa2. Verificare che l'impostazione predefinita venga ripristinata dopo l'esecuzione del programma.

setMode(*lista*) consente di modificare impostazioni multiple. *lista* contiene coppie di interi dei modi e di interi delle impostazioni. **setMode(*lista*)** restituisce una lista simile le cui coppie di interi rappresentano i modi e le impostazioni originali.

Se si salvano tutte le impostazioni di modo con **getMode(0) → var**, è possibile utilizzare **setMode(var)** per ripristinare tali impostazioni fintantoché la funzione o il programma esistono. Vedere **getMode()**, pagina 67.

Nota: le impostazioni di modo correnti vengono passate alle sottoroutine chiamate. Se una sottoroutine cambia un'impostazione di modo, tale modifica viene perduta quando il controllo torna alla routine di chiamata.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Nome modo	Intero modo	Interi impostazioni
Mostra cifre (Display digits)	1	1 =Mobile, 2 =Mobile1, 3 =Mobile2, 4 =Mobile3, 5 =Mobile4, 6 =Mobile5, 7 =Mobile6, 8 =Mobile7, 9 =Mobile8, 10 =Mobile9, 11 =Mobile10, 12 =Mobile11, 13 =Mobile12, 14 =Fissa0, 15 =Fissa1, 16 =Fissa2, 17 =Fissa3, 18 =Fissa4, 19 =Fissa5, 20 =Fissa6, 21 =Fissa7, 22 =Fissa8, 23 =Fissa9, 24 =Fissa10, 25 =Fissa11, 26 =Fissa12
Angolo (Angle)	2	1 =Radianti, 2 =Gradi, 3 =Gradanti
Formato esponenziale (Exponential Format)	3	1 =Normale, 2 =Scientifico, 3 =Tecnico
Reale o Complesso (Real or Complex)	4	1 =Reale, 2 =Rettangolare, 3 =Polare

Nome modo	Intero modo	Interi impostazioni
Auto o Approssimato (Auto or Approx.)	5	1 =Auto, 2 =Approssimato
Formato vettoriale (Vector Format)	6	1 =Rettangolare, 2 =Cilindrico, 3 =Sferico
Base	7	1 =Decimale, 2 =Esa, 3 =Binario

shift() (Sposta)

Catalogo > 

shift(Intero1[,numSpostamenti])⇒intero

Sposta i bit di un numero intero binario. È possibile inserire *Intero1* in qualsiasi base numerica; esso viene convertito automaticamente in forma binaria a 64 bit con segno. Se *Intero1* è troppo grande per questa forma, un'operazione a modulo simmetrico lo porta all'interno dell'intervallo. Per ulteriori informazioni, vedere **Base2**, pagina 17.

Se *numSpostamenti* è positivo, lo spostamento avviene verso sinistra. Se *numSpostamenti* è negativo, lo spostamento avviene verso destra. L'impostazione predefinita è -1 (sposta a destra di un bit).

In uno spostamento a destra, il bit più a destra viene eliminato e al suo posto viene inserito 0 o 1, in modo che corrisponda al bit più a sinistra. In uno spostamento a sinistra, il bit più a sinistra viene eliminato e viene inserito 0 come bit più a destra.

Ad esempio, in uno spostamento a destra:

ogni bit viene spostato verso destra.

0b0000000000000011101011000011010

Inserisce 0 se il bit più a sinistra è 0,

oppure 1 se il bit più a sinistra è 1.

Dà:

In modalità base Bin:

shift(0b1111010110000110101)	0b111101011000011010
shift(256,1)	0b1000000000

In modalità base Esadecimale:

shift(0h78E)	0h3C7
shift(0h78E,-2)	0h1E3
shift(0h78E,2)	0h1E38

Importante: per inserire un numero binario o esadecimale, utilizzare sempre il prefisso 0b o 0h (è uno zero, non la lettera O).

0b00000000000000111101011000011010

I risultati vengono visualizzati nella modalità Base che è stata impostata. Gli zeri iniziali non vengono visualizzati.

shift(Lista1 [,numSpostamenti])⇒lista

Restituisce una copia di *Lista1* spostata a destra o a sinistra di *numSpostamenti* elementi. Non modifica *Lista1*.

Se *numSpostamenti* è positivo, lo spostamento avviene verso sinistra. Se *numSpostamenti* è negativo, lo spostamento avviene verso destra. L'impostazione predefinita è -1 (sposta a destra di un elemento).

Gli elementi introdotti all'inizio o alla fine di *lista* a seguito dello spostamento sono contrassegnati con il simbolo "undef".

shift(Stringa1 [,numSpostamenti])⇒stringa

Restituisce una copia di *Stringa1* spostata a sinistra o a destra di *numSpostamenti* caratteri. Non modifica *Stringa1*.

Se *numSpostamenti* è positivo, lo spostamento avviene verso sinistra. Se *numSpostamenti* è negativo, lo spostamento avviene verso destra. L'impostazione predefinita è -1 (sposta a destra di un carattere).

I caratteri introdotti all'inizio o alla fine di *stringa* a seguito dello spostamento sono costituiti da uno spazio.

In modalità base Dec:

<code>shift({1,2,3,4})</code>	{ undef,1,2,3 }
<code>shift({1,2,3,4},-2)</code>	{ undef,undef,1,2 }
<code>shift({1,2,3,4},2)</code>	{ 3,4,undef,undef }

<code>shift("abcd")</code>	" abc"
<code>shift("abcd",-2)</code>	" ab"
<code>shift("abcd",1)</code>	"bcd "

sign(Lista1)⇒lista

sign(Matrice1)⇒matrice

sign(0) restituisce ± 1 se la modalità Formato complesso è Reale; altrimenti restituisce se stesso.

Se la modalità del formato complesso è Reale:

sign(0) rappresenta la circonferenza unitaria del dominio complesso.

Con liste o matrici, restituisce i segni di tutti gli elementi.

simult() Sistema di equazioni simultanee

simult(*matriceCoeff*, *vettoreCost*, *Tol*)⇒*matrice*

Restituisce un vettore colonna che contiene le soluzioni di un sistema di equazioni lineari.

Nota: vedere anche **linSolve()**, pagina 87.

matriceCoeff deve essere una matrice quadrata contenente i coefficienti delle equazioni.

vettoreCost deve avere lo stesso numero di righe (stesse dimensioni) di *matriceCoeff* e deve contenere i termini noti.

In alternativa, un elemento qualsiasi della matrice viene considerato zero se il suo valore assoluto è minore di *Tol*. Tale tolleranza viene utilizzata solo se la matrice contiene elementi a virgola mobile e non contiene variabili simboliche alle quali non sia stato assegnato un valore. In caso contrario, *Tol* viene ignorato.

- Se si imposta la modalità **Auto** o **Approssimato** su Approssimato, i calcoli verranno eseguiti in virgola mobile.
- Se *Tol* viene omesso o non è utilizzato, la tolleranza predefinita viene calcolata come: $5E-14 \cdot \max(\dim(\text{matriceCoeff})) \cdot \text{rowNorm}(\text{matriceCoeff})$

simult(*matriceCoeff*, *matriceCost*[, *Tol*])⇒*matrice*

Risolvere rispetto a x e y:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) \quad \begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

La soluzione è x=-3 e y=2.

Risolvere:

$$ax + by = 1$$

$$cx + dy = 2$$

Risolve sistemi multipli di equazioni lineari in cui ogni sistema ha coefficienti delle equazioni uguali ma termini noti diversi.

Ogni colonna di *matriceCost* deve contenere i termini noti per un sistema di equazioni. Ogni colonna della matrice risultante contiene la soluzione per il sistema corrispondente.

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right) = \begin{bmatrix} -3 & -7 \\ 2 & 9/2 \end{bmatrix}$$

Per il primo sistema, $x=-3$ e $y=2$. Per il secondo sistema, $x=-7$ e $y=9/2$.

sin() (Seno)

Tasto 

sin(Lista1)⇒*lista*

In modalità angolo in gradi:

sin(Lista1) restituisce una lista di seni di tutti gli elementi di *Lista1*.

In modalità angolo in gradi (gradi centesimali):

Nota: l'argomento è interpretato come angolo in gradi, gradi in gradi o radianti, a seconda della modalità angolo correntemente impostata. Si può usare $^{\circ}$, G o r per escludere tale impostazione provvisoriamente.

In modalità angolo in radianti:

sin
(*matriceQuadrata1*)⇒*matriceQuadrata*

In modalità angolo in radianti:

Restituisce il seno della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il seno di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

$$\text{sin}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

Tasto 

sin⁻¹⁽⁾ (Arcoseno)

sin⁻¹(Lista1)⇒*lista*

In modalità angolo in gradi:

sin⁻¹(Lista1) restituisce una lista contenente l'inversa del seno di ciascun elemento di *Lista1*.

In modalità angolo in gradi (gradi centesimali):

sin⁻¹() (Arcoseno)

Tasto 

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradianti o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arcsin(...)**.

sin⁻¹

(matriceQuadrata1)⇒matriceQuadrata

Restituisce l'inversa del seno della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare l'inversa del seno di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti:

In modalità angolo in radianti e in modalità formato rettangolare complesso:

$$\begin{aligned} \sin^{-1}\left(\begin{pmatrix} 1 & 5 \\ 4 & 2 \end{pmatrix}\right) \\ \begin{bmatrix} -0.174533-0.12198 \cdot i & 1.74533-2.35591 \cdot i \\ 1.39626-1.88473 \cdot i & 0.174533-0.593162 \cdot i \end{bmatrix} \end{aligned}$$

sinh() (Seno iperbolico)

Catalogo > 

sinh(Lista1)⇒lista

sinh (Lista1) restituisce una lista del seno iperbolico di ciascun elemento di *Lista1*.

sinh

(matriceQuadrata1)⇒matriceQuadrata

Restituisce il seno iperbolico della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il seno iperbolico di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti:

$$\begin{aligned} \sinh\left(\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}\right) \\ \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix} \end{aligned}$$

sinh⁻¹() (Arcoseno iperbolico)

Catalogo > 

sinh⁻¹(Lista1)⇒lista

sinh⁻¹(Lista1) restituisce una lista contenente l'inversa del seno iperbolico di ciascun elemento di *Lista1*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arcsinh(...)**.

sinh⁻¹**(matriceQuadrata1)⇒matriceQuadrata**

Restituisce l'inversa del seno iperbolico della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare l'inversa del seno iperbolico di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti:

$$\sinh^{-1} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

SinReg (Regressione sinusoidale)**SinReg X, Y [, [Iterazioni], [Periodo] [, Categoria, Includi]]**

Calcola la regressione sinusoidale sulle liste *X* e *Y*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e *Y* sono liste di variabili indipendenti e dipendenti.

Iterazioni è un valore che specifica quante volte al massimo (da 1 a 6) verrà tentata una soluzione. Se omesso, viene utilizzato 8. Di solito valori più alti danno una maggiore accuratezza ma richiedono tempi di esecuzione più lunghi, e viceversa.

Periodo indica il periodo stimato. Se viene omesso, la differenza tra i valori di *X* deve essere uguale e in ordine sequenziale. Se invece *Periodo* viene specificato, le differenze tra valori x possono non essere uguali.

Categoria è una lista di categoria dei dati corrispondenti *X* e *Y*.

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

L'output di **SinReg** è sempre espresso in radianti, indipendentemente dall'impostazione corrente della modalità di misurazione degli angoli.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Coefficienti di regressione
stat.Resid	Residui della regressione
stat.XReg	Lista di punti dati della <i>Lista X</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.YReg	Lista di punti dati della <i>Lista Y</i> modificata attualmente usata nella regressione secondo le restrizioni di <i>Freq</i> , <i>Lista Categoria</i> e <i>Includi Categoria</i>
stat.FreqReg	Lista di frequenze corrispondenti a <i>stat.XReg</i> e <i>stat.YReg</i>

SortA (Ordinamento ascendente)

SortA *Lista1*[, *Lista2*] [, *Lista3*] ...

SortA *Vettore1*[, *Vettore2*] [, *Vettore3*]

...

Questo comando permette di ordinare in modo ascendente gli elementi del primo argomento.

Se vengono inclusi altri argomenti, il comando permette di ordinare gli elementi di ciascuno di essi in modo che le loro nuove posizioni coincidano con le nuove posizioni degli elementi del primo argomento.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
SortA <i>list1</i>	<i>Done</i>
<i>list1</i>	$\{1,2,3,4\}$
$\{4,3,2,1\} \rightarrow list2$	$\{4,3,2,1\}$
SortA <i>list2</i> , <i>list1</i>	<i>Done</i>
<i>list2</i>	$\{1,2,3,4\}$
<i>list1</i>	$\{4,3,2,1\}$

SortA (Ordinamento ascendente)

Catalogo > 

Tutti gli argomenti devono essere nomi di liste o di vettori e devono avere le stesse dimensioni.

Gli elementi vuoti (nulli) nel primo argomento si spostano in fondo. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

SortD (Ordinamento discendente)

Catalogo > 

SortD *Lista1[, Lista2] [, Lista3] ...*

SortD *Vettore1[, Vettore2] [, Vettore3] ...*

Questo comando è identico a **SortA**, con la differenza che **SortD** ordina gli elementi in modo discendente.

Gli elementi vuoti (nulli) nel primo argomento si spostano in fondo. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
SortD <i>list1, list2</i>	<i>Done</i>
<i>list1</i>	$\{4,3,2,1\}$
<i>list2</i>	$\{3,4,1,2\}$

►Sphere (Visualizza come vettore sferico)

Catalogo > 

Vettore ►Sphere

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @>Sphere.

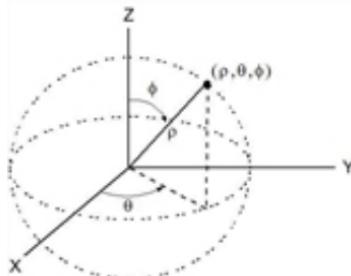
Visualizza il vettore di riga o colonna nel formato sferico $[\rho \angle\theta \angle\phi]$.

Vettore deve avere dimensione 3 e può essere un vettore di riga o colonna.

Nota: ►Sphere è un'istruzione nel formato di visualizzazione, ma non è una funzione di conversione. Tale istruzione può essere impiegata solo alla fine di una riga di introduzione.

$[1 \ 2 \ 3] \blacktriangleright \text{Sphere}$
 $[3.74166 \ \angle 1.10715 \ \angle 0.640522]$

$\left(2 \ \angle \frac{\pi}{4} \ 3\right) \blacktriangleright \text{Sphere}$
 $[3.60555 \ \angle 0.785398 \ \angle 0.588003]$



sqrt() (Radice quadrata)

sqrt(Lista1)⇒lista

Restituisce la radice quadrata dell'argomento.

In una lista, restituisce le radici quadrate di tutti gli elementi di *Lista1*.

Nota: vedere anche **Modello di radice quadrata**, pagina 1.

stat.results

Visualizza risultati di un calcolo statistico.

I risultati sono visualizzati sotto forma di coppie nome-valore. I nomi specifici mostrati dipendono dall'ultima funzione o comando statistico applicato.

È possibile copiare un nome o un valore e incollarlo in altre posizioni.

Nota: evitare di definire variabili con gli stessi nomi delle variabili utilizzate per l'analisi statistica. In alcuni casi, potrebbe prodursi un errore. I nomi di variabile utilizzati per l'analisi statistica sono elencati nella seguente tabella.

<i>xlist:=</i> { 1,2,3,4,5 }	{ 1,2,3,4,5 }
<i>ylist:=</i> { 4,8,11,14,17 }	{ 4,8,11,14,17 }

LinRegMx *xlist,ylist,1: stat.results*

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r ² "	0.996109
"r"	0.998053
"Resid"	"{...}"

<i>stat.values</i>	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	"{-0.4,0.4,0.2,0,-0.2}"

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBlock
stat.AdjR ²	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r ²	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.ox	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.oy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.ox1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.ox2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Sx	stat.X
stat.b9	stat.FBlock	stat.ŷ	stat.Sx ²	stat.X̄1
stat.b10	stat.Fcol	stat.ŷ1	stat.Sxy	stat.X̄2
stat.bList	stat.FInteract	stat.ŷ2	stat.Σy	stat.X̄Diff
stat.χ ²	stat.FreqReg	stat.ŷDiff	stat.Σy ²	stat.X̄List
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg

stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat. \bar{y}
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat. \hat{y}
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat. \hat{y} List
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

Nota: ogni volta che Foglio elettronico calcola risultati statistici, l'applicazione copia le variabili di gruppo "stat." in un gruppo "stat#", dove # è un numero che viene incrementato automaticamente. Questa funzione consente di mantenere i risultati precedenti mentre si eseguono calcoli multipli.

stat.values (Valori dei risultati)

Catalogo > 

stat.values

Per un esempio vedere **stat.results**.

Visualizza una matrice dei valori calcolati per l'ultima funzione o comando statistico calcolato.

Diversamente da **stat.results**, **stat.values** omette i nomi associati ai valori.

È possibile copiare un valore e incollarlo in altre posizioni.

stDevPop() (Deviazione standard della popolazione)

Catalogo > 

stDevPop(*Lista*[, *listaFreq*]) \Rightarrow espressione

In modalità angolo in radianti:

Restituisce la deviazione standard degli elementi di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

Nota: *Lista* deve avere almeno due elementi. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

stDevPop() (Deviazione standard della popolazione)

Catalogo > 

stDevPop(*Matrice1[, matriceFreq]*) \Rightarrow *matrice*

Restituisce un vettore di riga delle deviazioni standard della popolazione delle colonne di *Matrice1*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Matrice1*.

Nota: *Matrice1* deve avere almeno due righe. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

stDevSamp() (Deviazione standard del campione)

Catalogo > 

stDevSamp(*Lista[, listaFreq]*) \Rightarrow *espressione*

Restituisce la deviazione standard del campione di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

Nota: *Lista* deve avere almeno due elementi. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

stDevSamp(*Matrice1[, matriceFreq]*) \Rightarrow *matrice*

Restituisce un vettore di riga delle deviazioni standard del campione delle colonne di *Matrice1*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Matrice1*.

Nota: *Matrice1* deve avere almeno due righe. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

Stop**Catalogo > ****Stop**

Istruzione di programmazione: termina il programma il programma.

Stop non è ammesso nelle funzioni.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

<i>i:=0</i>	0
Define <i>prog1()</i> =Prgm	<i>Done</i>
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1()</i>	<i>Done</i>
<i>i</i>	5

Store (Memorizza)**Vedere → (memorizza), pagina 203.****string() (Da espressione a stringa)****Catalogo > **

string(*Espr*)⇒*stringa*

Semplifica *Espr* e restituisce il risultato come una stringa di caratteri.

subMat() (Sottomatrice)**Catalogo > **

subMat(*Matrice1*[, *rigaInizio*] [, *collInizio*] [, *rigaFine*] [, *collFine*])
⇒*matrice*

Restituisce la sottomatrice specificata di *Matrice1*.

Impostazioni predefinite: *rigaInizio*=1, *collInizio*=1, *rigaFine*=ultima riga, *collFine*=ultima colonna.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
subMat(<i>m1</i> ,2,1,3,2)	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
subMat(<i>m1</i> ,2,2)	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)**Vedere $\Sigma()$, pagina 195.****sum() (Somma)****Catalogo > **

sum(*Lista*[, *Inizio*[, *Fine*]])⇒*espressione*

Restituisce la somma degli elementi di *Lista*.

Inizio e *Fine* sono opzionali e specificano un intervallo di elementi.

Un argomento vuoto produce un risultato vuoto. Gli elementi vuoti (nulli) di *Lista* vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

sum(*Matrice1*[, *Inizio*[, *Fine*]])⇒*matrice*

Restituisce un vettore di riga contenente le somme degli elementi nelle colonne della *Matrice1*.

Inizio e *Fine* sono opzionali e specificano un intervallo di elementi.

Un argomento vuoto produce un risultato vuoto. Gli elementi vuoti (nulli) di *Matrice1* vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

sum $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$	[5 7 9]
sum $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	[12 15 18]
sum $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, 2, 3\right)$	[11 13 15]

sumIf(*Lista*, *Criteri*[, *listaSomma*])⇒*valore*

Restituisce la somma cumulata di tutti gli elementi di *Lista* che soddisfano i *Criteri* specificati. Opzionalmente è possibile specificare una lista alternativa, *ListaSomma*, per fornire gli elementi da accumulare.

Lista può essere un'espressione, una lista o una matrice. *listaSomma*, se specificata, deve avere le stesse dimensioni di *Lista*.

Criteri può essere:

- Un valore, un'espressione o una stringa. Ad esempio, **34** accumula solo quegli elementi di *Lista* che sono semplificati nel valore **34**.
- Un'espressione booleana contenente il simbolo **?** come segnaposto di ciascun

elemento. Ad esempio, `?<10` accumula solo quegli elementi di *Lista* che sono minori di 10.

Quando un elemento di *Lista* soddisfa i *Criteri*, esso viene aggiunto alla somma accumulata. Se si include *listaSomma*, l'elemento corrispondente di *listaSomma* viene aggiunto invece alla somma.

In Foglio elettronico, è possibile utilizzare un intervallo di celle al posto di *Lista* e di *listaSomma*.

Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

Nota: vedere anche **countIf()**, pagina 31.

Restituisce un sistema di equazioni, formattato come una lista. È inoltre possibile creare un sistema utilizzando un modello.

$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right) \quad x=4 \text{ and } y=-4$$

Matrice1 $\text{T} \Rightarrow$ *matrice*

Restituisce la trasposta dei complessi coniugati di *Matrice1*.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando `@t`.

`tan(Lista1) \Rightarrow lista`

In modalità angolo in gradi:

tan(Lista1) restituisce una lista delle tangenti di tutti gli elementi di *Lista1*.

Nota: l'argomento è interpretato come angolo in gradi, gradi o radianti, a seconda della modalità angolo correntemente impostata. Si può usare $^{\circ}$, G o r per escludere tale impostazione provvisoriamente.

tan

(matriceQuadrata1) \Rightarrow *matriceQuadrata*

Restituisce la tangente della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare la tangente di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in gradi (gradi centesimali):

In modalità angolo in radianti:

In modalità angolo in radianti:

$$\tan \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

tan⁻¹() (Arcotangente)

tan⁻¹(Lista1) \Rightarrow *lista*

tan⁻¹(Lista1) restituisce una lista dell'inversa della tangente di ciascun elemento di *Lista1*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, il risultato è in gradi, gradi o radianti.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arctan(...)**.

In modalità angolo in gradi:

$$\tan^{-1}(1) \quad 45$$

In modalità angolo in gradi (gradi centesimali):

$$\tan^{-1}(1) \quad 50$$

In modalità angolo in radianti:

$$\tan^{-1}(\{0,0.2,0.5\}) \quad \{0,0.197396,0.463648\}$$

In modalità angolo in radianti:

$$\tan^{-1} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

tan⁻¹

(matriceQuadrata1) \Rightarrow *matriceQuadrata*

Restituisce la tangente inversa della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare la tangente inversa di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

tanh() (Tangente iperbolica)Catalogo > **tanh(Lista1)⇒lista**

tanh(Lista1) restituisce una lista delle tangenti iperboliche di ciascun elemento di *Lista1*.

tanh**(matriceQuadrata1)⇒matriceQuadrata**

Restituisce la tangente iperbolica della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare la tangente iperbolica di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

In modalità angolo in radianti:

$$\tanh \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

tanh⁻¹() (Arcotangente iperbolica)Catalogo > **tanh⁻¹(Lista1)⇒lista**

In modalità formato rettangolare complesso:

tanh⁻¹(Lista1) restituisce una lista dell'inversa della tangente iperbolica di ciascun elemento di *Lista1*.

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **arctanh(...)**.

tanh⁻¹**(matriceQuadrata1)⇒matriceQuadrata**

Restituisce la tangente iperbolica inversa della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare la tangente iperbolica inversa di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

In modalità angolo in radianti e in modalità formato rettangolare complesso:

$$\tanh^{-1} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \begin{bmatrix} -0.099353+0.164058 \cdot i & 0.267834-1.4908 \\ -0.087596-0.725533 \cdot i & 0.479679-0.9473 \cdot i \\ 0.511463-2.08316 \cdot i & -0.878563+1.7901 \end{bmatrix}$$

tanh⁻¹() (Arcotangente iperbolica)

Catalogo > 

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

Per vedere l'intero risultato, premere **▲**, quindi utilizzare **◀** e **▶** per spostare il cursore.

tCdf() (Probabilità di distribuzione t di Student)

Catalogo > 

tCdf(*estremoInf*,*estremoSup*,*gl*) \Rightarrow numero se *estremoInf* e *estremoSup* sono numeri, lista se *estremoInf* e *estremoSup* sono liste

Calcola la probabilità della distribuzione *t* di Student tra il *estremoInf* e il *estremoSup* per i gradi di libertà *gl* specificati.

Text

Catalogo > 

Text*stringaPrompt*[, *FlagDispl*]

Istruzione di programmazione: Sospende il programma e visualizza la stringa di caratteri *stringaPrompt* in una finestra di dialogo.

Selezionando **OK**, l'esecuzione del programma continua.

L'argomento opzionale *flag* può essere un'espressione.

- Se *FlagDispl* viene omesso o dà come risultato **1**, il messaggio testuale viene aggiunto alla cronologia di Calcolatrice.
- Se *FlagDispl* dà come risultato **0**, il messaggio testuale non viene aggiunto alla cronologia.

Se il programma richiede di digitare una risposta, vedere **Request, pagina 132** oppure **RequestStr, pagina 134**.

Nota: è possibile utilizzare questo comando all'interno di un programma definito dall'utente, ma non di una funzione.

Definire un programma che si arresta momentaneamente per visualizzare ciascuno dei cinque numeri casuali in una finestra di dialogo.

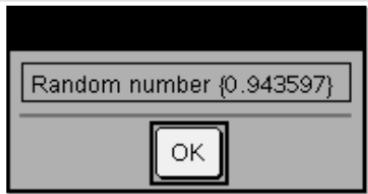
All'interno del modello **Prgm...EndPrgm**, completare ciascuna riga premendo **↵** invece di **enter**. Sulla tastiera del computer, mantenere premuto **Alt** e premere **Invio**.

```
Define text_demo()=Prgm
  For i,1,5
    strinfo:="Random number " & string
    (rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Eseguire il programma:

```
text_demo()
```

Esempio di finestra di dialogo:



Then

Vedere If, pagina 72.

tInterval (Intervallo di confidenza t)Catalogo > **tInterval** *Lista[,Freq[,livelloConfidenza]]*

(Input lista dati)

tInterval *\bar{x} ,sx,n[,livelloConfidenza]*

(Input statistiche riepilogo)

Calcola un intervallo di confidenza *t*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza per una media non nota di una popolazione
stat. \bar{x}	Media del campione della successione di dati dalla distribuzione casuale normale
stat.ME	Margine di errore
stat.df	Gradi di libertà
stat.sx	Deviazione standard del campione
stat.n	Lunghezza della successione di dati con media del campione

**tInterval_2Samp (Intervallo di
confidenza t su due campioni)****Catalogo > ****tInterval_2Samp** *List1,Lista2[,Freq1
[,Freq2[,livelloConfidenza[,Aggregata]]]]*

(Input lista dati)

tInterval_2Samp *Ȑx1,sx1,n1,Ȑx2,sx2,n2
[,livelloConfidenza[,Aggregata]]*

(Input statistiche riepilogo)

Calcola un intervallo di confidenza *t* su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Aggregata=1 aggrega le varianze;
Aggregata=0 non aggrega le varianze.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza contenente la probabilità di distribuzione del livello di confidenza
stat.Ȑx1-Ȑx2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat.ME	Margine di errore
stat.df	Gradi di libertà
stat.Ȑx1, stat.Ȑx2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat.Ȑx1, stat.Ȑx2	Deviazioni standard dei campioni di <i>List1</i> e <i>List2</i>
stat.n1, stat.n2	Numero di campioni nelle successioni di dati
stat.sp	Deviazione standard aggregata. Calcola quando <i>Aggregata</i> = sì.

**tPdf() (Densità di probabilità t di
Student)****Catalogo > ****tPdf**(*ValX,gl*) \Rightarrow numero se *ValX* è un numero, lista se *ValX* è una lista

tPdf() (Densità di probabilità t di Student)

Catalogo > 

Calcola la funzione della densità di probabilità (pdf) per la distribuzione *t* di Student in corrispondenza di un valore *x* specificato con i gradi di libertà *gl* specificati.

trace() (Traccia)

Catalogo > 

Restituisce la traccia (somma di tutti gli elementi sulla diagonale principale) di matriceQuadrata.

Try (Tentativo)

Catalogo > 

Try

```
Define prog1()=Prgm
  Try
  z:=z+1
  Disp "z incremented."
  Else
  Disp "Sorry, z undefined."
  EndTry
EndPrgm
```

Done

z:=1;prog1()

z incremented.

Done

DelVar z;prog1()

Sorry, z undefined.

Done

blocco1

Else

blocco2

EndTry

Esegue blocco1 a meno che non si verifichi un errore. In questo caso, l'esecuzione del programma viene trasferita a blocco2. La variabile di sistema *errCode* contiene il codice di errore che consente al programma di eseguire il ripristino dell'errore. Per un elenco dei codici di errore, vedere "Codici di errore e messaggi", pagina 232.

blocco1 e blocco2 possono essere o una singola istruzione o una serie di istruzioni separate dal carattere ":".

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Esempio 2

```
Define eigenvals(a,b)=Prgm
```

© Questo programma visualizza gli autovalori di A·B

Per vedere i comandi **Try**, **ClrErr** e **PassErr** in funzione, il programma **eigenvals()** riportato sulla destra. Avviare il programma eseguendo ciascuna delle seguenti espressioni.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

Nota: vedere anche **ClrErr**, pagina 24 e **PassErr**, pagina 116.

Try

Disp "A= ",a

Disp "B= ",b

Disp " "

Disp "Eigenvalues of A·B are:",eigVl(a*b)

Else

If errCode=230 Then

Disp "Error: Product of A·B must be a square matrix"

ClrErr

Else

PassErr

EndIf

EndTry

EndPrgm

tTest $\mu0$,*Lista* [,*Freq* [,*Ipotesi*]]

(Input lista dati)

tTest $\mu0$, \bar{x} ,*sx*,*n* [,*Ipotesi*]

(Input statistiche riepilogo)

Esegue una verifica dell'ipotesi su un'unica media μ non nota di una popolazione quando la deviazione standard σ della popolazione non è nota. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Viene verificata l'ipotesi nulla $H_0: \mu = \mu0$ in contrapposizione a una delle alternative seguenti:

Per $H_a: \mu < \mu0$, impostare *Ipotesi*<0Per $H_a: \mu \neq \mu0$ (default), impostare *Ipotesi*=0

Per $H_a: \mu > \mu_0$, impostare *Ipotesi>0*

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.t	$(\bar{x} - \mu_0) / (stdev / \sqrt{n})$
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Gradi di libertà
stat. \bar{x}	Media del campione della sequenza di dati in <i>Lista</i>
stat.sx	Deviazione standard del campione della sequenza di dati
stat.n	Dimensione dei campioni

tTest_2Samp (Verifica t su due campioni)

tTest_2Samp *Lista1*,*Lista2*[,*Freq1*[,*Freq2*[,*Ipotesi*[,*Aggregata*]]]]

(Input lista dati)

tTest_2Samp $\bar{x}_1, sx_1, n_1, \bar{x}_2, sx_2, n_2$ [,*Ipotesi*[,*Aggregata*]]

(Input statistiche riepilogo)

Esegue una verifica *t* su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Viene verificata l'ipotesi nulla $H_0: \mu_1 = \mu_2$ in contrapposizione a una delle alternative seguenti:

Per $H_a: \mu_1 < \mu_2$, impostare *Ipotesi>0*

Per $H_a: \mu_1 \neq \mu_2$ (default), impostare *Ipotesi=0*

Per $H_a: \mu_1 > \mu_2$, impostare *Ipotesi>0*

Aggregata=1 aggrega le varianze

tTest_2Samp (Verifica t su due campioni)

Catalogo > 

Aggregata=0 non aggrega le varianze

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.t	Valore normale standard calcolato per la differenze delle medie
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat.df	Gradi di libertà della statistica t
stat.Ȑx1, stat.Ȑx2	Medie dei campioni delle sequenze di dati di <i>Lista1</i> e <i>Lista2</i>
stat.sx1, stat.sx2	Deviazioni standard dei campioni delle sequenze di dati di <i>Lista1</i> e <i>Lista2</i>
stat.n1, stat.n2	Dimensione dei campioni
stat.sp	Deviazione standard aggregata. Calcolato quando <i>Aggregata</i> =1.

tvmFV()

Catalogo > 

tvmFV(*N,I,PV,Pmt,[Pp Y],[Cp Y],[PmtAt]*) \Rightarrow valore

tvmFV(120,5,0,-500,12,12)

77641.1

Funzione finanziaria che calcola il valore futuro del denaro.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 170.
Vedere anche **amortTbl()**, pagina 7.

tvmI()

Catalogo > 

tvmI(*N,PV,Pmt,FV,[Pp Y],[Cp Y],[PmtAt]*) \Rightarrow valore

tvmI(240,100000,-1000,0,12,12)

10.5241

Funzione finanziaria che calcola il tasso di interesse annuo.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 170.
Vedere anche **amortTbl()**, pagina 7.

tvmN()**Catalogo > ****tvmN($I, PV, Pmt, FV, [PpY], [CpY], [PmtAt]$)** \Rightarrow valore

tvmN(5,0,-500,77641,12,12)

120.

Funzione finanziaria che calcola il numero di periodi di pagamento.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 170. Vedere anche **amortTbl()**, pagina 7.

tvmPmt()**Catalogo > ****tvmPmt($N, I, PV, FV, [PpY], [CpY], [PmtAt]$)** \Rightarrow valore

tvmPmt(60,4,30000,0,12,12)

-552.496

Funzione finanziaria che calcola l'importo di ciascuna rata.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 170. Vedere anche **amortTbl()**, pagina 7.

tvmPV()**Catalogo > ****tvmPV($N, I, Pmt, FV, [PpY], [CpY], [PmtAt]$)** \Rightarrow valore

tvmPV(48,4,-500,30000,12,12)

-3426.7

Funzione finanziaria che calcola il valore presente.

Nota: gli argomenti utilizzati nelle funzioni TVM sono descritti nella tabella degli argomenti TVM, pagina 170. Vedere anche **amortTbl()**, pagina 7.

Argomento TVM *	Descrizione	Tipo di dati
N	Numero di periodi di pagamento	Numero reale
I	Tasso di interesse annuo	Numero reale
PV	Valore presente	Numero reale
Pmt	Importo della rata	Numero reale
FV	Valore futuro	Numero reale

Argomento TVM *	Descrizione	Tipo di dati
PpY	Rate all'anno, impostazione predefinita=1	Numero intero > 0
CpY	Periodi di capitalizzazione all'anno, impostazione predefinita=1	Numero intero > 0
$PmtAt$	Pagamento versato alla fine o all'inizio di ogni periodo di pagamento, impostazione predefinita=fine	Numero intero (0=fine, 1=inizio)

* Questi nomi di argomenti TVM sono simili ai nomi delle variabili TVM (come ad esempio **tvm.pv** e **tvm.pmt**) che sono utilizzate dal Risolutore finanziario dell'applicazione *Calculator*. Le funzioni finanziarie, tuttavia, non memorizzano automaticamente i valori degli argomenti o i risultati in variabili TVM.

TwoVar (Risultati a due variabili)

Catalogo > 

TwoVar $X, Y[, Freq] [, Categoria, Includi]$

Calcola le statistiche TwoVar. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Tutte le liste devono avere le stesse dimensioni, ad eccezione di *Includi*.

X e Y sono liste di variabili indipendenti e dipendenti.

Freq è una lista opzionale di valori di frequenza. Ciascun elemento di *Freq* specifica la frequenza di occorrenza di ogni dato corrispondente di X e Y . Il valore predefinito è 1. Tutti gli elementi devono essere numeri interi ≥ 0 .

Categoria è una lista di categoria dei dati corrispondenti X e Y .

Includi è una lista di uno o più codici di categoria. Solo quei dati il cui codice di categoria è inserito in questa lista vengono inclusi nel calcolo.

Un elemento vuoto (nullo) in qualsiasi lista *X*, *Freq* o *Categoria* produce un corrispondente elemento vuoto in tutte queste liste. Un elemento vuoto (nullo) in qualsiasi lista da *X1* a *X20* produce un corrispondente elemento vuoto in tutte queste liste. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

Variabile di output	Descrizione
stat. \bar{X}	Media dei valori X
stat. x	Somma dei valori X
stat. x2	Somma dei valori x2
stat.sx	Deviazione standard del campione di X
stat. x	Deviazione standard della popolazione di X
stat.n	Numero dei punti di dati
stat. \bar{y}	Media dei valori y
stat. y	Somma dei valori y
stat. y ²	Somma dei valori y ²
stat.sy	Deviazione standard del campione di y
stat. y	Deviazione standard della popolazione di y
stat. xy	Somma dei valori x · y
stat.r	Coefficiente di correlazione
stat.MinX	Minimo dei valori x
stat.Q ₁ X	1° quartile di x
stat.MedianX	Mediana di x
stat.Q ₃ X	3° quartile di x
stat.MaxX	Massimo dei valori x
stat.MinY	Minimo dei valori y
stat.Q ₁ Y	1° quartile di x
stat.MedY	Mediana di y
stat.Q ₃ Y	3° quartile di y

Variabile di output	Descrizione
stat.MaxY	Massimo dei valori Y
stat. (x-) ²	Somma dei quadrati delle deviazioni dalla media di x
stat. (y-) ²	Somma dei quadrati delle deviazioni dalla media di y

U

unitV() (Vettore unità)

Catalogo > 

unitV(*Vettore1*)⇒*vettore*

Restituisce un vettore unità riga o colonna, a seconda della forma di *Vettore1*.

Vettore1 deve essere una matrice composta da una sola riga o da una sola colonna.

unLock

Catalogo > 

unLockVar1 [, *Var2*] [, *Var3*] ...

unLockVar.

Sblocca le variabili o il gruppo di variabili specificate. Le variabili bloccate non possono essere modificate o eliminate.

Vedere **Lock**, pagina 90 e **getLockInfo()**, pagina 67.

<i>a:=65</i>	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a:=75</i>	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a:=75</i>	75
DelVar <i>a</i>	Done

V

varPop() (Varianza della popolazione)

Catalogo > 

varPop(*Lista*[, *listaFreq*])⇒*espressione*

Restituisce la varianza della popolazione di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

Nota: *Lista* deve contenere almeno due elementi.

Se un elemento di una lista è vuoto (nullo), l'elemento viene ignorato, come pure l'elemento corrispondente nell'altra lista. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

varSamp() (Varianza campione)

varSamp(*Lista*[, *listaFreq*]) \Rightarrow espressione

Restituisce la varianza campione di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

Nota: *Lista* deve contenere almeno due elementi.

Se un elemento di una lista è vuoto (nullo), l'elemento viene ignorato, come pure l'elemento corrispondente nell'altra lista. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

varSamp(*MatriceI*[, *matriceFreq*]) \Rightarrow matrice

Restituisce un vettore riga contenente la varianza campione di ciascuna colonna di *MatriceI*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *MatriceI*.

Nota: *MatriceI* deve contenere almeno due righe.

Se un elemento di una matrice è vuoto (nullo), l'elemento viene ignorato, come pure l'elemento corrispondente nell'altra matrice. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 222.

varSamp	$\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{pmatrix}$	$\begin{bmatrix} 4.75 & 1.03 & 4 \end{bmatrix}$
varSamp	$\begin{pmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{pmatrix}$	$\begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}$

$$\begin{bmatrix} 3.91731 & 2.08411 \end{bmatrix}$$

Wait**Catalogo > ****Wait *tempoInSecondi***

Sospende l'esecuzione per un periodo di *tempoInSecondi* secondi.

Wait è particolarmente utile in un programma che necessita di un breve ritardo per consentire che i dati richiesti diventino disponibili.

L'argomento *tempoInSecondi* deve essere un'espressione che viene semplificata a un valore decimale da 0 a 100. Il comando arrotonda questo valore al decimo di secondo più vicino.

Per annullare un **Wait** in corso,

- **Palmare:** Tenere premuto il tasto **Home** e premere **Enter** più volte.
- **Windows®:** Tenere premuto il tasto **F12** e premere **Invio** più volte.
- **Macintosh®:** Tenere premuto il tasto **F5** e premere **Invio** più volte.
- **iPad®:** L'app mostra un prompt. È possibile continuare ad attendere o annullare.

Nota: È possibile utilizzare il comando **Wait** all'interno di un programma definito dall'utente, ma non di una funzione.

Per attendere 4 secondi:

Wait 4

Per attendere 1/2 secondo:

Wait 0.5

Per attendere 1,3 secondi utilizzando la variabile *seccount*:

seccount:=1.3

Wait seccount

Questo esempio fa accendere un LED verde per 0,5 secondi e quindi lo spegne.

Send "SET GREEN 1 ON"

Wait 0.5

Send "SET GREEN 1 OFF"

warnCodes ()**Catalogo > **

**warnCodes(*Espr1*,
varStato)**⇒*espressione*

Calcola l'espressione *Espr1*, restituisce il risultato e memorizza i codici di tutte le avvertenze generate nella variabile di lista *varStato*. Se non sono generate avvertenze, questa funzione assegna a *varStato* una lista vuota.

Espr1 può essere qualsiasi espressione matematica valida di TI-Nspire™ o TI-Nspire™ CAS. Non è possibile utilizzare un comando o una assegnazione come *Espr1*.

varStato deve essere un nome di variabile valido.

Per un elenco dei codici di avvertenza e dei messaggi associati, vedere pagina 241.

when() (Quando)

**Catalogo > **

when(*Condizione*, *risultatoVero* [, *risultatoFalso*][, *risultatoSconosciuto*])
⇒ *espressione*

Restituisce *risultatoVero*, *risultatoFalso* o *risultatoSconosciuto*, a seconda che *Condizione* sia vera, falsa o sconosciuta. Restituisce la voce inserita se gli argomenti immessi sono insufficienti per specificare un risultato appropriato.

Omettere sia *risultatoFalso* che *risultatoSconosciuto* se si desidera che l'espressione venga definita solamente nella regione in cui *Condizione* è vera.

Utilizzare **undef** come *risultatoFalso* per definire un'espressione che esegua una rappresentazione grafica solo in un intervallo.

when() è utile per la definizione di funzioni ricorsive.

when $\{x < 0, x + 3\}$	$ x = 5$	Done
when $\{n > 0, n \cdot factorial(n - 1), 1\}$	$\rightarrow factorial(n)$	
$factorial(3)$		6
3!		6

While Condizione*Blocco***EndWhile**

Esegue le istruzioni di *Blocco* fino a quando *Condizione* è vera.

Blocco può essere una singola espressione o una serie di espressioni separate dal carattere ":".

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define *sum_of_recip(n)*=FuncLocal *i,tempsum*1→*i*0→*tempsum*While *i*≤*n**tempsum*+ $\frac{1}{i}$ →*tempsum**i*+1→*i*

EndWhile

Return *tempsum*

EndFunc

Done

sum_of_recip(3) $\frac{11}{6}$

6

X**xor**

BooleanExpr1 xor BooleanExpr2
restituisce Boolean expression

true xor true	false
5>3 xor 3>5	true

BooleanList1 xor BooleanList2
restituisce Boolean list

BooleanMatrix1 xor BooleanMatrix2
restituisce Boolean matrix

Restituisce true se *Espressione booleana1* è vera e *Espressione booleana2* è falsa, o viceversa.

Restituisce false se gli argomenti sono entrambi veri o falsi. Restituisce un'espressione booleana semplificata se nessuno degli argomenti può essere risolto in vero o falso.

Nota: vedere **or**, pagina 113.

Intero1 xor Intero2 ⇒ *intero*

In modalità base Esadecimale:

Importante: è zero, non la lettera O.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

In modalità base Bin:

Confronta due interi reali bit per bit tramite un'operazione **xor**. Internamente, entrambi gli interi vengono convertiti in numeri binari a 64 bit con segno. Quando vengono confrontati bit corrispondenti, il risultato sarà 1 se uno dei due bit (ma non entrambi) è uguale a 1; se entrambi i bit sono uguali a 0 o a 1, il risultato sarà 0. Il valore restituito rappresenta il risultato dei bit e viene visualizzato nel modo Base che è stato impostato.

È possibile inserire gli interi in qualsiasi base numerica. Se si tratta di un numero binario o esadecimale, utilizzare rispettivamente il prefisso 0b o 0h. Senza prefisso, gli interi vengono considerati decimali (base 10).

Se viene indicato un intero decimale troppo grande per una forma binaria con segno a 64 bit, verrà utilizzata un'operazione a modulo simmetrico per portare il valore nell'intervallo appropriato. Per ulteriori informazioni, vedere **►Base2**, pagina 17.

Nota: vedere **or**, pagina 113.

Z

zInterval (Intervallo di confidenza Z)

zInterval $\sigma, \text{Lista}[, \text{Freq}[, \text{livelloConfidenza}]]$

(Input lista dati)

zInterval $\sigma, \bar{x}, n [, \text{livelloConfidenza}]$

(Input statistiche riepilogo)

Calcola un intervallo di confidenza z . Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155)

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

0b100101 xor 0b100

0b100001

Nota: un numero binario può contenere fino a 64 cifre (oltre al prefisso 0b). Un numero esadecimale può contenere fino ad 16 cifre.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza per una media non nota di una popolazione
stat. \bar{x}	Media del campione della successione di dati dalla distribuzione casuale normale
stat.ME	Margine di errore
stat.sx	Deviazione standard del campione
stat.n	Lunghezza della successione di dati con media del campione
stat. σ	Deviazione standard nota della popolazione per la sequenza di dati <i>Lista</i>

zInterval_1Prop (Intervallo di confidenza z per una proporzione)

Catalogo > 

zInterval_1Prop $x, n [, livelloConfidenza]$

Calcola un intervallo di confidenza z per una proporzione. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

x è un intero non negativo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza contenente la probabilità di distribuzione del livello di confidenza
stat. \hat{p}	Proporzione calcolata di casi favorevoli
stat.ME	Margine di errore
stat.n	Numero di campioni nelle successioni di dati

zInterval_2Prop (Intervallo di confidenza z per due proporzioni)

Catalogo > 

zInterval_2Prop $x1, n1, x2, n2 [, livelloConfidenza]$

**zInterval_2Prop (Intervallo di
confidenza z per due proporzioni)****Catalogo > **

Calcola l'intervallo di confidenza z per due proporzioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

$x1$ e $x2$ sono interi non negativi.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza contenente la probabilità di distribuzione del livello di confidenza
stat. \hat{p} Diff	Differenza calcolata tra le proporzioni
stat.ME	Margine di errore
stat. $\hat{p}1$	Stima della proporzione del primo campione
stat. $\hat{p}2$	Stima della proporzione del secondo campione
stat.n1	Dimensione del campione nella successione di dati uno
stat.n2	Dimensione del campione nella successione di dati due

**zInterval_2Samp (Intervallo di
confidenza z su due campioni)****Catalogo > **

zInterval_2Samp $\sigma_1, \sigma_2, List1, Lista2[, Freq1$
 $, Freq2, [livelloConfidenza]]$

(Input lista dati)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2$
 $, [livelloConfidenza]$

(Input statistiche riepilogo)

Calcola l'intervallo di confidenza z su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.CLower, stat.CUpper	Intervallo di confidenza contenente la probabilità di distribuzione del livello di confidenza
stat. \bar{x}_1 - \bar{x}_2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat.ME	Margine di errore
stat. \bar{x}_1 , stat. \bar{x}_2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat. σx_1 , stat. σx_2	Deviazioni standard dei campioni di <i>Lista 1</i> e <i>Lista 2</i>
stat.n1, stat.n2	Numero di campioni nelle successioni di dati
stat.r1, stat.r2	Deviazioni standard note della popolazione per la successione di dati <i>Lista1</i> e <i>Lista2</i>

zTest (Verifica z)

Catalogo > 

zTest $\mu 0, \sigma, Lista, [Freq, Ipotesi]$

(Input lista dati)

zTest $\mu 0, \sigma, \bar{x}, n, [Ipotesi]$

(Input statistiche riepilogo)

Esegue una verifica *z* con frequenza *listaFreq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Viene verificata l'ipotesi nulla $H_0: \mu = \mu_0$ in contrapposizione a una delle alternative seguenti:

Per $H_a: \mu < \mu_0$, impostare *Ipotesi*<0

Per $H_a: \mu \neq \mu_0$ (default), impostare *Ipotesi*=0

Per $H_a: \mu > \mu_0$, impostare *Ipotesi*>0

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$

Variabile di output	Descrizione
stat.P Valore	Probabilità minima in cui l'ipotesi nulla può essere rifiutata
stat. \bar{x}	Media del campione della sequenza di dati in <i>Lista</i>
stat.sx	Deviazione standard del campione della sequenza di dati. Restituita solo per input di <i>Dati</i> .
stat.n	Dimensione dei campioni

zTest_1Prop (Verifica z per una proporzione)

Catalogo > 

zTest_1Prop $p0,x,n,[Ipotesi]$

Esegue una verifica z per una proporzione. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

x è un intero non negativo.

Viene verificata l'ipotesi nulla $H_0: p = p0$ in contrapposizione a una delle alternative seguenti:

Per $H_a: p > p0$, impostare *Ipotesi>0*

Per $H_a: p \neq p0$ (*default*), impostare *Ipotesi=0*

Per $H_a: p < p0$, impostare *Ipotesi<0*

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.p0	Proporzione della popolazione ipotizzata
stat.z	Valore normale standard calcolato per la proporzione
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat. \hat{p}	Proporzione attesa del campione
stat.n	Dimensione dei campioni

zTest_2Prop (Verifica z per due proporzioni)

Catalogo > 

zTest_2Prop $x1, n1, x2, n2, [Ipotesi]$

Esegue una verifica z per due proporzioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

$x1$ e $x2$ sono interi non negativi.

Viene verificata l'ipotesi nulla $H_0: p1 = p2$ in contrapposizione a una delle alternative seguenti:

Per $H_a: p1 > p2$, impostare *Ipotesi*>0

Per $H_a: p1 \neq p2$ (default), impostare *Ipotesi*=0

Per $H_a: p1 < p2$, impostare *Ipotesi*<0

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.z	Valore normale standard calcolato per la differenza delle proporzioni
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat. $\hat{p}1$	Stima della proporzione del primo campione
stat. $\hat{p}2$	Stima della proporzione del secondo campione
stat. \hat{p}	Stima delle proporzioni del campione aggregate
stat.n1, stat.n2	Numero di campioni presi nelle prove 1 e 2

zTest_2Samp (Verifica su due campioni)

Catalogo > 

zTest_2Samp $\sigma_1, \sigma_2, Lista1, Lista2, [Freq1, Freq2, [Ipotesi]]$

(Input lista dati)

zTest_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2, [Ipotesi]$

(Input statistiche riepilogo)

Esegue una verifica z su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 155).

Viene verificata l'ipotesi nulla $H_0: \mu_1 = \mu_2$ in contrapposizione a una delle alternative seguenti:

Per $H_a: \mu_1 < \mu_2$, impostare *Ipotesi*<0

Per $H_a: \mu_1 \neq \mu_2$ (default), impostare *Ipotesi*=0

Per $H_a: \mu_1 > \mu_2$, *Ipotesi*>0

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 222.

Variabile di output	Descrizione
stat.z	Valore normale standard calcolato per la differenza delle medie
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata
stat. \bar{x}_1 , stat. \bar{x}_2	Medie dei campioni delle sequenze di dati di <i>Lista1</i> e <i>Lista2</i>
stat.sx1, stat.sx2	Deviazioni standard dei campioni delle sequenze di dati di <i>Lista1</i> e <i>Lista2</i>
stat.n1, stat.n2	Dimensione dei campioni

Simboli

+ (addizione)

Restituisce la somma dei due argomenti.

	Tasto 
56	56
56+4	60
60+4	64
64+4	68
68+4	72

Lista1 + Lista2⇒*lista*

Matrice1 + Matrice2⇒*matrice*

Restituisce una lista (o matrice) contenente le somme degli elementi corrispondenti di *Lista1* e *Lista2* (oppure di *Matrice1* e *Matrice2*).

Gli argomenti devono avere uguali dimensioni.

$15 + \{10, 15, 20\}$	$\{25, 30, 35\}$
$\{10, 15, 20\} + 15$	$\{25, 30, 35\}$
$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$

Nota: utilizzare **+** (punto più) per aggiungere un'espressione a ciascun elemento.

-(sottrazione)

Tasto 

Lista1 - Lista2⇒*lista*

Matrice1 - Matrice2⇒*matrice*

Sottrae ciascun elemento di *Lista2* (o di *Matrice2*) dal corrispondente elemento di *Lista1* (o di *Matrice1*) e restituisce il risultato.

Gli argomenti devono avere uguali dimensioni.

$15 - \{10, 15, 20\}$	$\{5, 0, -5\}$
$\{10, 15, 20\} - 15$	$\{-5, 0, 5\}$

-(sottrazione)**Tasto**

Nota: Utilizzare $\cdot-$ (punto meno) per sottrarre un'espressione da ciascun elemento.

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

·(moltiplicazione)**Tasto**

Restituisce il prodotto di due argomenti.

Lista1 · *Lista2* ⇒ *lista*

Restituisce una lista contenente i prodotti degli elementi corrispondenti di *Lista1* e *Lista2*.

Le liste devono avere uguali dimensioni.

Matrice1 · *Matrice2* ⇒ *matrice*

Restituisce il prodotto di matrice di *Matrice1* per *Matrice2*.

Il numero di colonne di *Matrice1* deve essere uguale al numero di righe di *Matrice2*.

Nota: Utilizzare \cdot (punto moltiplicazione) per moltiplicare un'espressione per ciascun elemento.

/ (divisione)**Tasto**

Nota: vedere anche **Modello di frazione**, pagina 1.

Lista1 / *Lista2* ⇒ *lista*

Restituisce una lista contenente i quozienti di *Lista1* diviso per *Lista2*.

Le liste devono avere uguali dimensioni.

Nota: utilizzare $\cdot /$ (punto divisione) per dividere un'espressione per ciascun elemento.

$$\frac{\{1,2,3\}}{\{4,5,6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

^ (elevamento a potenza)**Tasto**

Lista1 ^ *Lista2* ⇒ *lista*

^ (elevamento a potenza)

Tasto 

Restituisce il primo argomento elevato alla potenza del secondo argomento.

Nota: vedere anche **Modello di esponente**, pagina 1.

In una lista, restituisce gli elementi di *Lista1* elevati alla potenza dei corrispondenti elementi di *Lista2*.

Nel dominio reale, le potenze frazionarie con esponenti ridotti e denominatori dispari utilizzano l'ambito reale rispetto a quello principale, impiegata nel modo complesso.

$$\left\{1,2,3,4\right\}^{-2} \quad \left\{1,\frac{1}{4},\frac{1}{9},\frac{1}{16}\right\}$$

matriceQuadrata1 ^ *intero* \Rightarrow *matrice*

Restituisce *matriceQuadrata1* elevata alla potenza di *intero*.

matriceQuadrata1 deve essere una matrice quadrata.

Se *intero* = -1, viene calcolata l'inversa della matrice.

Se *intero* < -1, viene calcolata l'inversa della matrice a una potenza positiva appropriata.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \quad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} \frac{11}{4} & -\frac{5}{4} \\ 2 & 2 \\ -\frac{15}{4} & \frac{7}{4} \end{bmatrix}$$

x² (quadrato)

Tasto 

Restituisce il quadrato dell'argomento.

Lista1 ^ 2 \Rightarrow *lista*

Restituisce una lista contenente i quadrati degli elementi di *Lista1*.

matriceQuadrata1 ^ 2 \Rightarrow *matrice*

$$4^2 \quad 16$$

$$\left\{2,4,6\right\}^2 \quad \left\{4,16,36\right\}$$

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2 \quad \begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \cdot^2 \quad \begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$$

x² (quadrato)Tasto 

Restituisce la matrice *matriceQuadrata1* elevata al quadrato. Ciò non equivale a calcolare il quadrato di ogni elemento. Utilizzare $.^2$ per calcolare il quadrato di ogni elemento.

.+ (punto addizione)Tasti 

Matrice1 .+ *Matrice2* \Rightarrow matrice

Matrice1 .+ *Matrice2* restituisce una matrice che è la somma di ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

.- (punto sottrazione).Tasti 

Matrice1 .- *Matrice2* \Rightarrow matrice

Matrice1 .- *Matrice2* restituisce una matrice che è la differenza tra ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

.· (punto moltiplicazione)Tasti 

Matrice1 .· *Matrice2* \Rightarrow matrice

Matrice1 .· *Matrice2* restituisce una matrice che è il prodotto di ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

. / (punto divisione)Tasti 

Matrice1 ./ *Matrice2* \Rightarrow matrice

Matrice1 ./ *Matrice2* restituisce una matrice che è il quoziente di ciascuna coppia di elementi corrispondenti di *Matrice1* e di *Matrice2*.

= (uguale)

Tasto  =

$Matrice1 = Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere uguale a $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere uguale ad $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define $g(x)=$ Func

If $x \leq -5$ Then

Return 5

ElseIf $x > -5$ and $x < 0$ Then

Return $-x$

ElseIf $x \geq 0$ and $x < 10$ Then

Return x

ElseIf $x = 10$ Then

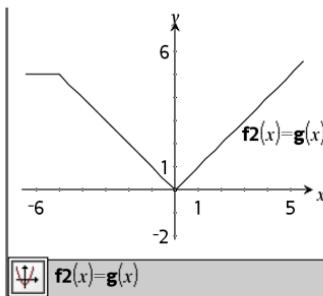
Return 3

EndIf

EndFunc

Done

Risultato della rappresentazione grafica $g(x)$



≠ (diverso)

Tasti  =

$Espr1 \neq Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di "=" (uguale).

$Lista1 \neq Lista2 \Rightarrow$ lista booleana

$Matrice1 \neq Matrice2 \Rightarrow$ Matrice booleana

Restituisce true se $Espr1$ è determinata in modo da non essere uguale a $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da essere uguale a $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando `/=`

< (minore di)

$Espr1 < Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di `=` (uguale).

$Lista1 < Lista2 \Rightarrow$ lista booleana

$Matrice1 < Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere minore di $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere maggiore o uguale a $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

≤ (minore di o uguale a)

$Espr1 \leq Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di `=` (uguale).

$Lista1 \leq Lista2 \Rightarrow$ lista booleana

$Matrice1 \leq Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere minore o uguale a $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere maggiore di $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando `<=`

> (maggiore di)

Tasti

$Espr1 > Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di “=” (uguale).

$Lista1 > Lista2 \Rightarrow$ lista booleana

$Matrice1 > Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere maggiore di $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere minore o uguale a $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

\geq (maggiore di o uguale a)

Tasti

$Espr1 \geq Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di “=” (uguale).

$Lista1 \geq Lista2 \Rightarrow$ lista booleana

$Matrice1 \geq Matrice2 \Rightarrow$ matrice booleana

Restituisce true se $Espr1$ è determinata in modo da essere maggiore o uguale a $Espr2$.

Restituisce false se $Espr1$ è determinata in modo da non essere minore o uguale a $Espr2$.

Negli altri casi, restituisce una forma semplificata dell'equazione.

Nelle liste e nelle matrici, restituisce un confronto elemento per elemento.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando \geq

\Rightarrow (implicazione logica)

ctrl = tasti

$BooleanExpr1 \Rightarrow BooleanExpr2$
restituisce Boolean expression

$BooleanList1 \Rightarrow BooleanList2$
restituisce Boolean list

$BooleanMatrix1 \Rightarrow BooleanMatrix2$
restituisce Boolean matrix

$Integer1 \Rightarrow Integer2$ restituisce Integer

$5 > 3$ or $3 > 5$	true
$5 > 3 \Rightarrow 3 > 5$	false
3 or 4	7
$3 \Rightarrow 4$	-4
$\{1, 2, 3\}$ or $\{3, 2, 1\}$	$\{3, 2, 3\}$
$\{1, 2, 3\} \Rightarrow \{3, 2, 1\}$	$\{-1, -1, -3\}$

Valuta l'espressione **not** <argomento1>
or <argomento2> e restituisce true, false
oppure una forma semplificata
dell'equazione.

Nel caso di liste e matrici, restituisce un
confronto elemento per elemento.

Nota: È possibile inserire questo
operatore dalla tastiera digitando \Rightarrow

\Leftrightarrow (doppia implicazione logica, XNOR)

ctrl = tasti

$BooleanExpr1 \Leftrightarrow BooleanExpr2$
restituisce Boolean expression

$BooleanList1 \Leftrightarrow BooleanList2$
restituisce Boolean list

$BooleanMatrix1 \Leftrightarrow BooleanMatrix2$
restituisce Boolean matrix

$Integer1 \Leftrightarrow Integer2$ restituisce Integer

$5 > 3$ xor $3 > 5$	true
$5 > 3 \Leftrightarrow 3 > 5$	false
3 xor 4	7
$3 \Leftrightarrow 4$	-8
$\{1, 2, 3\}$ xor $\{3, 2, 1\}$	$\{2, 0, 2\}$
$\{1, 2, 3\} \Leftrightarrow \{3, 2, 1\}$	$\{-3, -1, -3\}$

Restituisce la negazione di un'operazione
XOR logica riguardo i due argomenti.
Restituisce true, false oppure una forma
semplificata dell'equazione.

Nel caso di liste e matrici, restituisce un
confronto elemento per elemento.

Nota: È possibile inserire questo
operatore dalla tastiera digitando \Leftrightarrow

! (fattoriale)**Tasto**  $List1! \Rightarrow lista$ $Matrice1! \Rightarrow matrice$

Restituisce il fattoriale dell'argomento.

In una lista o in una matrice, restituisce rispettivamente una lista o una matrice dei fattoriali di ciascun elemento.

5!	120
$\{\{5,4,3\}\}!$	$\{120,24,6\}$
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}!$	$\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

& (aggiunge)**Tasti**   $Stringa1 \& Stringa2 \Rightarrow stringa$ Restituisce una stringa di testo, aggiungendo $Stringa2$ a $Stringa1$.

"Hello "&"Nick"

"Hello Nick"

d() (derivata)**Catalogo** > **Nota:** è possibile inserire questa funzione dalla tastiera del computer digitando **derivative** (...). **$\sqrt()$ (radice quadrata)****Tasti**   $\sqrt(List1) \Rightarrow lista$

Restituisce la radice quadrata dell'argomento.

In una lista, restituisce le radici quadrate di tutti gli elementi di $List1$.**Nota:** è possibile inserire questa funzione dalla tastiera del computer digitando **sqrt** (...)**Nota:** vedere anche **Modello di radice quadrata**, pagina 1. **$\prod()$ (prodSeq)****Catalogo** >  $\prod(Espr1, Var, Basso, Alto) \Rightarrow espressione$

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **prodSeq**(...).

Calcola *Espr1* per ciascun valore di *Var* da *Basso* ad *Alto* e restituisce il prodotto dei risultati.

Nota: vedere anche **Modello di prodotto** (Π), pagina 5.

$\Pi\{\text{Espr1}, \text{Var}, \text{Basso}, \text{Basso-1}\} \Rightarrow 1$

$\Pi\{\text{Espr1}, \text{Var}, \text{Basso}, \text{Alto}\} \Rightarrow 1 / \Pi\{\text{Espr1}, \text{Var}, \text{Alto+1}, \text{Basso-1}\}$ if *Alto* < *Basso-1*

Le formule di prodotto utilizzate sono derivate dal seguente riferimento:

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\prod_{k=4}^3 \langle k \rangle \quad 1$$

$$\prod_{k=4}^1 \left\langle \frac{1}{k} \right\rangle \quad 6$$

$$\prod_{k=4}^1 \left\langle \frac{1}{k} \right\rangle \cdot \prod_{k=2}^4 \left\langle \frac{1}{k} \right\rangle \quad \frac{1}{4}$$

$\Sigma\{\text{Espr1}, \text{Var}, \text{Basso}, \text{Alto}\} \Rightarrow \text{espressione}$

Nota: è possibile inserire questa funzione dalla tastiera del computer digitando **sumSeq**(...).

Calcola *Espr1* per ciascun valore di *Var* da *Basso* ad *Alto* e restituisce la somma dei risultati.

Nota: vedere anche **Modello di somma**, pagina 5.

$\Sigma\{\text{Espr1}, \text{Var}, \text{Basso}, \text{Basso-1}\} \Rightarrow 0$

$\Sigma\{\text{Espr1}, \text{Var}, \text{Basso}, \text{Alto}\} \Rightarrow -\Sigma\{\text{Espr1}, \text{Var}, \text{Alto+1}, \text{Basso-1}\}$ if *Alto* < *Basso-1*

$$\sum_{k=4}^3 \langle k \rangle \quad 0$$

$\Sigma()$ (sumSeq)

Catalogo >

Le formule di sommatoria utilizzate sono derivate dal seguente riferimento:

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\begin{array}{c} \sum_{k=4}^1 (k) \\ \hline \sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) \end{array}$$

$\Sigma\text{Int}()$

Catalogo >

$\Sigma\text{Int}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [valoreArrotondato]) \Rightarrow valore$

ΣInt

```
(  
NPmt1  
,
```

$NPmt2, \text{tabellaAmmortamento}) \Rightarrow valore$

Funzione di ammortamento che calcola la somma dell'interesse pagato durante un intervallo specifico di pagamenti.

$NPmt1$ e $NPmt2$ definiscono l'inizio e la fine dell'intervallo dei pagamenti.

$N, I, PV, Pmt, FV, PpY, CpY$ e $PmtAt$ sono descritti nella tabella degli argomenti TVM, pagina 170.

- Se si omette Pmt , viene utilizzata l'impostazione predefinita $Pmt=\text{tvmPmt}$ ($N, I, PV, FV, PpY, CpY, PmtAt$).
- Se si omette FV , viene utilizzata l'impostazione predefinita $FV=0$.
- Le impostazioni predefinite di PpY , CpY e $PmtAt$ sono le stesse delle funzioni TVM.

$valoreArrotondato$ specifica il numero di cifre decimali di arrotondamento. Impostazione predefinita=2.

$tbl:=\text{amortTbl}(12, 12, 4.75, 20000, , 12, 12)$				
0	0.	0.	20000.	
1	-79.17	-1630.69	18369.3	
2	-72.71	-1637.15	16732.2	
3	-66.23	-1643.63	15088.5	
4	-59.73	-1650.13	13438.4	
5	-53.19	-1656.67	11781.7	
6	-46.64	-1663.22	10118.5	
7	-40.05	-1669.81	8448.7	
8	-33.44	-1676.42	6772.28	
9	-26.81	-1683.05	5089.23	
10	-20.14	-1689.72	3399.51	
11	-13.46	-1696.4	1703.11	
12	-6.74	-1703.12	-0.01	

$\Sigma\text{Int}(1, 3, tbl)$ -218.11

ΣInt

(NPmt1, NPmt2, tabellaAmmortamento)
 calcola la somma degli interessi sulla
 base della tabella di ammortamento
tabellaAmmortamento. L'argomento
tabellaAmmortamento deve essere una
 matrice avente la forma descritta in
amortTbl(), pagina 7.

Nota: vedere anche **ΣPrn()**, sotto, e **Bal()**,
 pagina 16.

ΣPrn()

ΣPrn(NPmt1, NPmt2, N, I, PV, [Pmt],
[FV], [PpY], [CpY], [PmtAt],
[valoreArrotondato])⇒*valore*

ΣPrn

(
NPmt1

,
NPmt2, tabellaAmmortamento)⇒*valore*

Funzione di ammortamento che calcola
 la somma del capitale versato durante
 un intervallo specifico di pagamenti.

NPmt1 e *NPmt2* definiscono l'inizio e la
 fine dell'intervallo dei pagamenti.

N, I, PV, Pmt, FV, PpY, CpY e PmtAt
 sono descritti nella tabella degli
 argomenti TVM, pagina 170.

- Se si omette *Pmt*, viene utilizzata
 l'impostazione predefinita
Pmt=tvmPmt
 $(N, I, PV, FV, PpY, CpY, PmtAt)$.
- Se si omette *FV*, viene utilizzata
 l'impostazione predefinita *FV=0*.
- Le impostazioni predefinite di *PpY*,
CpY e *PmtAt* sono le stesse delle
 funzioni TVM.

valoreArrotondato specifica il numero di
 cifre decimali di arrotondamento.
 Impostazione predefinita=2.

ΣPrn(1,3,12,4.75,20000,,12,12) -4911.47

<i>tbl:=amortTbl(12,12,4.75,20000,,12,12)</i>				
0	0.	0.	20000.	
1	-79.17	-1630.69	18369.3	
2	-72.71	-1637.15	16732.2	
3	-66.23	-1643.63	15088.5	
4	-59.73	-1650.13	13438.4	
5	-53.19	-1656.67	11781.7	
6	-46.64	-1663.22	10118.5	
7	-40.05	-1669.81	8448.7	
8	-33.44	-1676.42	6772.28	
9	-26.81	-1683.05	5089.23	
10	-20.14	-1689.72	3399.51	
11	-13.46	-1696.4	1703.11	
12	-6.74	-1703.12	-0.01	

ΣPrn(1,3,tbl) -4911.47

ΣPrn

$(NPmt1, NPmt2, tabellaAmmortamento)$
 calcola la somma del capitale versato
 sulla base della tabella di
 ammortamento *tabellaAmmortamento*.
 L'argomento *tabellaAmmortamento*
 deve essere una matrice avente la forma
 descritta in **amortTbl()**, pagina 7.

Nota: vedere anche $\Sigma Int()$, sopra, e $Bal()$,
 pagina 16.

(conversione indiretta)

Tasti  

stringaNomeVariabile

Crea o fa riferimento alla variabile xyz.

Questo operatore, che si riferisce alla variabile chiamata *stringaNomeVariabile*, permette di creare e di modificare le variabili dall'interno di una funzione mediante le stringhe.

$10 \rightarrow r$	10
$"r" \rightarrow s1$	$"r"$
$#s1$	10

Restituisce il valore della variabile (r) il cui nome è memorizzato nella variabile s1.

E (notazione scientifica)

Tasto 

mantissaEesponente

23000.	23000.
2300000000.+4.1E15	4.1E15
$3 \cdot 10^4$	30000

Permette di inserire un numero in notazione scientifica. Il numero viene interpretato come *mantissa* \times $10^{\text{esponente}}$.

Suggerimento: se si desidera inserire una potenza di 10 senza ottenere un valore decimale per risultato, utilizzare 10^{intero} .

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando @E. Per esempio, digitare $2.3@E4$ per inserire 2.3E4.

g (gradanti)

Tasto 1

Espr1g \Rightarrow espressione

In modalità di misurazione degli angoli in gradi, gradanti o radianti:

Espr1g \Rightarrow espressione

Lista1g \Rightarrow lista

Matrice1g \Rightarrow matrice

Questa funzione consente di utilizzare un angolo espresso in gradanti pure essendo attiva la modalità di misurazione degli angoli in gradi o in radianti.

In modalità angolo in radianti, moltiplica *Espr1* per $\pi/200$.

In modalità angolo in gradi, moltiplica *Espr1* per $g/100$.

In modalità gradanti, restituisce *Espr1* senza modifiche.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando @g.

r (radianti)

Tasto 1

Lista1r \Rightarrow lista

In modalità angolo in gradi, gradanti o radianti:

Matrice1r \Rightarrow matrice

Questa funzione consente di utilizzare un angolo espresso in radianti pur essendo attiva la modalità di misurazione degli angoli in gradi o in gradanti.

In modalità angolo in gradi, moltiplica l'argomento per $180/\pi$.

In modalità angolo in radianti, restituisce l'argomento immutato.

In modalità angolo in gradanti, moltiplica l'argomento per $200/\pi$.

Suggerimento: ricorrere al simbolo r se si desidera forzare l'uso dei radianti in una definizione, indipendentemente dalla modalità prevalente nella funzione.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando **€r**.

° (gradi)

Lista 1°⇒lista

Matrice 1°⇒matrice

Questa funzione consente di utilizzare un angolo espresso in gradi pur essendo attiva la modalità di misurazione degli angoli in gradi o in radianti.

In modalità angolo in radianti, moltiplica l'argomento per $\pi/180$.

In modalità angolo in gradi, restituisce l'argomento immutato.

In modalità angolo in gradi, moltiplica l'argomento per $10/9$.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando **€d**.

°, ', " (gradi/primi/secondi)

gg°pp'ss.ss"⇒espressione

ggun numero positivo o negativo

ppun numero non negativo

ss.ssun numero non negativo

Restituisce $gg+(pp/60)+(ss.ss/3600)$.

Questo formato di introduzione in base -60 consente di:

- Inserire un angolo in gradi/primi/secondi indipendentemente dalla modalità di misurazione degli angoli corrente.
- Inserire un orario nella forma ore/minuti/secondi.

Nota: far seguire ss.ss da due apostrofi (') e non dal simbolo di virgolette (").

In modalità angolo in gradi, gradi o radianti:

In modalità angolo in radianti:

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right)$$

$$\{1., 0.707107, 0., 0.864976\}$$

[*Raggio*, $\angle\theta$ *Angolo*] \Rightarrow vettore
(formato polare)

In modalità angolo in radianti e con il formato vettore impostata su:

[*Raggio*, $\angle\theta$ *Angolo*,*Z*_{Coordinate}] \Rightarrow vettore

rettangolare

(formato cilindrico)

cilindrico

[*Raggio*, $\angle\theta$ *Angolo*, $\angle\theta$ *Angolo*] \Rightarrow vettore

sferico

(formato sferico)

Restituisce le coordinate nella forma di un vettore a seconda dell'impostazione della modalità formato vettore: rettangolare, cilindrico o sferico.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando @<.

(*Grandezza* \angle *Angolo*) \Rightarrow valoreComplesso

In modalità angolo in radianti e in modalità formato rettangolare complesso:

(formato polare)

$$5+3 \cdot i \cdot \left(10 \angle \frac{\pi}{4}\right) \quad -2.07107 - 4.07107 \cdot i$$

Inserisce un valore complesso in formato polare ($r\angle\theta$). *Angolo* viene interpretato conformemente alla modalità di misurazione degli angoli impostata.

_ (trattino basso come elemento vuoto)

**Vedere “Elementi vuoti (nulli)”,
pagina 222.**

_ (trattino basso come designatore di unità)

Tasti

*Variabile*_

Assumendo che *z* non sia definito:

Se *Variabile* non ha un valore, essa verrà trattata come se indicasse un numero complesso. Per impostazione predefinita, senza _, la variabile verrà considerata un numero reale.

$\text{real}(z)$	<i>z</i>
$\text{real}(z_)$	$\text{real}(z_)$
$\text{imag}(z)$	0
$\text{imag}(z_)$	$\text{imag}(z_)$

Se a *Variabile* è associato un valore, il carattere _ viene ignorato e *Variabile* conserverà il tipo di dati originale.

10^ (List1)⇒list

Restituisce 10 elevato alla potenza dell'argomento.

In una lista, restituisce 10 elevato alla potenza degli elementi di *List1*.

10^**(matriceQuadrata1)⇒matriceQuadrata**

Restituisce 10 elevato alla potenza di *matriceQuadrata1*. Ciò non equivale a calcolare 10 elevato alla potenza di ciascun elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

$$\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1.14336\text{e}7 & 8.17155\text{e}6 & 6.67589\text{e}6 \\ 9.95651\text{e}6 & 7.11587\text{e}6 & 5.81342\text{e}6 \\ 7.65298\text{e}6 & 5.46952\text{e}6 & 4.46845\text{e}6 \end{bmatrix}$$

matriceQuadrata1 deve essere diagonalizzabile. Il risultato contiene sempre numeri a virgola mobile.

^-1(reciproco)**List1^-1⇒list**

Restituisce il reciproco dell'argomento.

In una lista, restituisce i reciproci degli elementi di *List1*.

matriceQuadrata1^-1⇒matriceQuadrata

Restituisce l'inversa di *matriceQuadrata1*.

matriceQuadrata1 deve essere una matrice quadrata non singolare.

| (operatore di vincolo)

Expr | *BooleanExpr1*
[**and***BooleanExpr2*]...

Expr | *BooleanExpr1*
[**or***BooleanExpr2*]...

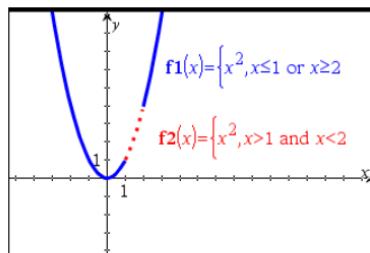
Il simbolo di vincolo ("|") serve da operatore binario. L'operando a sinistra di | è un'espressione. L'operando a destra di | definisce una o più relazioni destinate ad avere un effetto sulla semplificazione dell'espressione. Le relazioni multiple dopo | devono essere collegate da operatori logici **"and"** o **"or"**.

L'operatore di vincolo fornisce tre tipi primari di funzionalità:

- Sostituzioni
- Vincoli d'intervallo
- Esclusioni

Le sostituzioni si presentano nella forma di equazione, come per esempio $x=3$ o $y=\sin(x)$. I migliori risultati si ottengono quando la parte sinistra è una variabile semplice. *Expr | Variabile = valore* sostituirà *valore* per ogni ricorrenza di *Variabile* in *Expr*.

I vincoli d'intervallo si presentano nella forma di una o più disequazioni, collegate da operatori logici **"and"** o **"or"**. I vincoli d'intervallo permettono inoltre la semplificazione, la quale sarebbe altrimenti non valida o non calcolabile.



$$\text{solve}(x^2 - 1 = 0, x) |_{x \neq 1} \quad x = 1$$

Le esclusioni utilizzano l'operatore relazionale "diverso da" (\neq) per escludere dalla valutazione un valore specifico.

→ (memorizza)

Tasti

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando `=:` come scelta rapida. Per esempio, digitare `pi/4 =: Miavar.`

*Var := Lista**Var := Matrice**Funzione(Param1,...) := Espr**Funzione(Param1,...) := Lista**Funzione(Param1,...) := Matrice***© (commento)****© [testo]**

© considera *testo* come una riga di commento che consente di annotare le funzioni e i programmi creati.

© può trovarsi all'inizio o in qualsiasi altro punto della riga. Tutto quanto si trova a destra del segno © fino alla fine della riga viene considerato come commento.

Nota per l'inserimento dell'esempio: per istruzioni sull'inserimento di definizioni di programmi e funzioni costituite da più righe, consultare la sezione Calcolatrice del manuale del prodotto.

Define $g(n) = \text{Func}$

© Declare variables

Local $i, result$ $result := 0$ For $i, 1, n, 1$ ©Loop n times $result := result + i^2$

EndFor

Return $result$

EndFunc

Done

g(3)

14

0b, 0h**0b numeroBinario**

In modalità base Dec:

0b10+0hF+10

27

0h numeroEsadecimale

Indica, rispettivamente, un numero binario o un numero esadecimale. Per inserire un numero binario o esadecimale, inserire il prefisso 0b o 0h indipendentemente dalla modalità Base che è stata impostata. Senza prefisso, un numero viene considerato decimale (base 10).

I risultati vengono visualizzati nella modalità Base che è stata impostata.

In modalità base Bin:

0b10+0hF+10

0b11011

In modalità base Esadecimale:

0b10+0hF+10

0h1B

TI-Nspire™ CX II - Comandi di Disegna

Si tratta di un documento integrativo della Guida di riferimento di TI-Nspire™ e della Guida di riferimento di TI-Nspire™ CAS. Tutti i comandi di TI-Nspire™ CX II verranno integrati e pubblicati nella versione 5.1 della Guida di riferimento di TI-Nspire™ e della Guida di riferimento di TI-Nspire™ CAS.

Programmazione grafica

Ai palmari TI-Nspire™ CX II e alle applicazioni da desktop TI-Nspire™ sono stati aggiunti nuovi comandi per la programmazione grafica.

I palmari TI-Nspire™ CX II passeranno a questa modalità grafico durante l'esecuzione dei comandi grafici e ritorneranno alla modalità di esecuzione del programma precedente una volta completato il programma.

Durante l'esecuzione del programma, sulla barra superiore verrà visualizzato il messaggio "In esecuzione...". Al completamento del programma, verrà mostrato il messaggio "Finito". La modalità grafico viene disattivata con la pressione di qualsiasi tasto.

- Il passaggio alla modalità grafico viene attivato automaticamente quando il programma TI-Basic individua uno dei comandi (grafico) di Disegna durante l'esecuzione.
- Questo passaggio avviene solo quando si esegue un programma dalla Calcolatrice; in un documento o una Calcolatrice nel Blocco note.
- Il passaggio alla modalità grafico avviene al termine del programma.
- La modalità grafico è disponibile solo nella Vista palmare di TI-Nspire™ CX II e nella Vista palmare di TI-Nspire™ CX II per desktop. Ciò significa che non è disponibile nella vista documento del computer sul desktop o su iOS.
 - Se durante l'esecuzione di un programma TI-Basic nella modalità errata viene individuato un comando grafico, verrà visualizzato un messaggio di errore e il programma TI-Basic verrà chiuso.

Schermata grafica

La schermata grafica presenta un'intestazione nella parte superiore dello schermo che non può essere modificata dai comandi grafici.

L'area di disegno della schermata grafica viene cancellata (colore = 255.255.255) quando si inizializza la schermata grafica.

Schermata grafica	Predefinito
Altezza	212
Larghezza	318
Colore	bianco: 255.255.255

Impostazioni e vista predefinite

- Le icone di stato nella barra superiore (stato della batteria, stato Premere per Test, indicatore di rete, ecc.) non sono visibili durante l'esecuzione di un programma grafico.
- Colore predefinito per il disegno: nero (0,0,0)
- Stile della penna predefinito: normale, uniforme
 - Spessore: 1 (sottile), 2 (normale), 3 (molto spesso)
 - Stile 1 (uniforme), 2 (punteggiato), 3 (tratteggiato)
- Tutti i comandi di disegno utilizzano le impostazioni del colore e della penna correnti, vale a dire i valori predefiniti oppure i valori impostati mediante i comandi di TI-Basic.
- Il carattere del testo è fisso e non può essere modificato.
- Qualsiasi disegno eseguito nella schermata grafica viene tracciato all'interno di una finestra di ritaglio che presenta le stesse dimensioni dell'area di disegno della schermata grafica. Qualsiasi disegno che fuoriesce da quest'area di disegno ritagliata della schermata grafica non verrà tracciato. Non verrà visualizzato alcun messaggio di errore.
- Tutte le coordinate x,y specificate per i comandi di disegno vengono definite in modo che 0,0 si trovi nell'angolo superiore sinistro dell'area di disegno della schermata grafica.
 - **Eccezioni:**
 - **DrawText** utilizza le coordinate nell'angolo inferiore sinistro del riquadro di delimitazione per il testo.
 - **SetWindow** utilizza l'angolo inferiore sinistro della schermata.
- È possibile fornire tutti i parametri dei comandi come espressioni che restituiscono un numero che viene arrotondato al numero intero più vicino.

Messaggi di errore della schermata grafica

Se la convalida non riesce, viene visualizzato un messaggio di errore.

Messaggio di errore	Descrizione	Visualizza
Errore Sintassi	Se vengono rilevati eventuali errori di sintassi, l'applicazione di controllo visualizza un messaggio di errore e tenta di portare il cursore in prossimità del primo errore in modo che sia possibile correggerlo.	
Errore Argomenti mancanti	Nella funzione o nel comando mancano uno o più argomenti	Error Too few arguments The function or command is missing one or more arguments. OK
Errore Troppi argomenti	La funzione o il comando contiene un numero eccessivo di argomenti e non è possibile eseguirne il calcolo.	Error Too many arguments The function or command contains an excessive number of arguments and cannot be evaluated. OK
Errore Tipo di dati non valido	Un argomento è del tipo di dati errato.	Error Invalid data type An argument is of the wrong data type. OK

Comandi non validi in modalità grafico

Alcuni comandi vengono disattivati quando il programma passa alla modalità grafico. Se questi comandi vengono individuati durante la modalità grafico, verrà visualizzato un messaggio di errore e il programma verrà chiuso.

Comando non consentito	Messaggio di errore
Request	Impossibile eseguire Request in modalità grafico
RequestStr	Impossibile eseguire RequestStr in modalità grafico
Testo	Impossibile eseguire Text in modalità grafico

I comandi che eseguono la stampa del testo sulla Calcolatrice, **disp** e **dispAt**, saranno comandi supportati nella modalità grafico. Il testo immesso con questi comandi viene inviato alla schermata della Calcolatrice (non alla schermata grafica) e risulta visibile una volta che si chiude il programma e il sistema ritorna all'app Calcolatrice.

Cancellare**Cancella $x, y, \text{larghezza}, \text{altezza}$**

Cancella l'intera schermata se non viene specificato alcun parametro.

Se vengono specificati $x, y, \text{larghezza}$ e altezza , verrà cancellato il rettangolo definito dai parametri.

Cancellare

Cancella l'intera schermata

Cancella $10, 10, 100, 50$

Cancella l'area di un rettangolo con l'angolo in alto a sinistra su (10, 10) e con larghezza 100, altezza 50

DrawArcCatalogo > 
CXII**DrawArc** *x, y, larghezza, altezza, startAngle, arcAngle*

Disegna un arco all'interno del rettangolo di delimitazione definito con gli angoli di inizio e dell'arco forniti.

x, y: coordinata superiore sinistra del rettangolo di selezione*larghezza, altezza*: dimensioni del rettangolo di delimitazione

L'"angolo dell'arco" definisce l'estensione dell'arco.

Questi parametri possono essere forniti come espressioni che restituiscono un numero che viene in seguito arrotondato al numero intero più vicino.

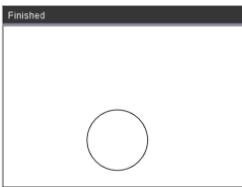
DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180

**Vedere anche:** [FillArc](#)**DrawCircle**Catalogo > 
CXII**DrawCircle** *x, y, raggio**x, y*: coordinata del centro*raggio*: raggio della circonferenza

DrawCircle 150,150,40

**Vedere anche:** [FillCircle](#)

DrawLine

Catalogo >  CXII

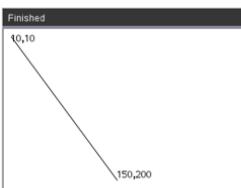
DrawLine $x1, y1, x2, y2$

Disegna una linea da $x1, y1, x2, y2$.

Espressioni che restituiscono un numero che viene in seguito arrotondato al numero intero più vicino.

Limiti della schermata: se in base alle coordinate specificate qualsiasi parte della linea viene tracciata all'esterno della schermata grafica, tale parte verrà tagliata e non verrà visualizzato un messaggio di errore.

DrawLine 10,10,150,200



DrawPoly

Catalogo >  CXII

I comandi presentano due varianti:

DrawPoly $xlist, ylist$

oppure

DrawPoly $x1, y1, x2, y2, x3, y3 \dots xn, yn$

Nota: la forma **DrawPoly** $xlist, ylist$ collegherà $x1, y1$ a $x2, y2$, $x2, y2$ a $x3, y3$ e così via.

Nota: **DrawPoly** $x1, y1, x2, y2, x3, y3 \dots xn, yn$

xn, yn **NON** verrà automaticamente collegata a $x1, y1$.

Espressioni che restituiscono un elenco di valori mobili reali
 $xlist, ylist$

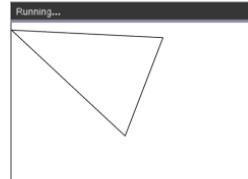
Espressioni che restituiscono un unico valore mobile reale

$x1, y1 \dots xn, yn$ = coordinate per i vertici di un poligono

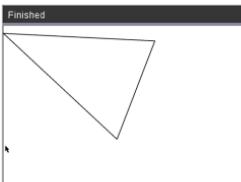
$xlist := \{0, 200, 150, 0\}$

$ylist := \{10, 20, 150, 20\}$

DrawPoly $xlist, ylist$



DrawPoly 0,10,200,20,150,150,0,10



Nota: **DrawPoly**: dimensioni dell'input (larghezza/altezza) relative alle linee disegnate.

Le linee vengono tracciate in un riquadro di delimitazione attorno alla coordinata e alle dimensioni specificate in modo che le dimensioni effettive del poligono disegnato siano superiori a quelle della larghezza e dell'altezza.

Vedere anche: [FillPoly](#)

DrawRect

DrawRect *x, y, larghezza, altezza*

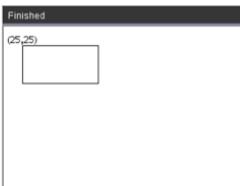
x, y: coordinata superiore sinistra del rettangolo

larghezza, altezza: larghezza e altezza del rettangolo (rettangolo tracciato verso il basso e a destra rispetto alla coordinata iniziale).

Nota: le linee vengono tracciate in un riquadro di delimitazione attorno alla coordinata e alle dimensioni specificate in modo che le dimensioni effettive del rettangolo disegnato siano superiori a quelle indicate dalla larghezza e dall'altezza.

Vedere anche: [FillRect](#)

DrawRect 25,25,100,50



DrawText

DrawText *x, y, exprOrString1*
[,exprOrString2]...

x, y: coordinata dell'output del testo

Disegna il testo in *exprOrString* in corrispondenza delle coordinate *x, y* specificate.

DrawText 50,50,"Hello World"



Le regole per *exprOrString* sono identiche a quelle per **Disp** - **DrawText** può contenere più argomenti.

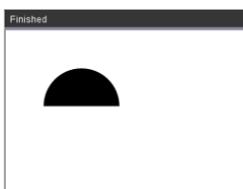
FillArcCatalogo >  CXII**FillArc** $x, y, \text{larghezza, altezza}$ $\text{startAngle, arcAngle}$ x, y : coordinata superiore sinistra del rettangolo di selezione

Disegna e riempie un arco all'interno del rettangolo di delimitazione definito con gli angoli di inizio e dell'arco forniti.

Il colore di riempimento predefinito è il nero. È possibile impostare il colore di riempimento con il comando [SetColor](#).

L'“angolo dell'arco” definisce l'estensione dell'arco.

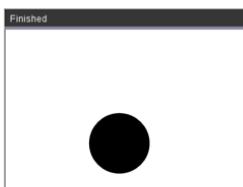
FillArc 50,50,100,100,0,180

**FillCircle**Catalogo >  CXII**FillCircle** x, y, raggio x, y : coordinata del centro

Disegna e riempie una circonferenza in corrispondenza del centro specificato con il raggio specificato.

Il colore di riempimento predefinito è il nero. È possibile impostare il colore di riempimento con il comando [SetColor](#).

FillCircle 150,150,40



Here!

FillPolyCatalogo >  CXII**FillPoly** $xlist, ylist$

xlist:={0,200,150,0}

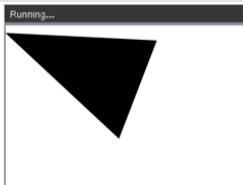
oppure

ylist:={10,20,150,10}

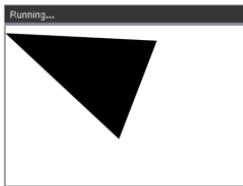
FillPoly $x1, y1, x2, y2, x3, y3 \dots xn, yn$

FillPoly xlist,ylist

Nota: utilizzare [SetColor](#) e [SetPen](#) per specificare la linea e il colore.



FillPoly 0,10,200,20,150,150,0,10



FillRect

FillRect *x, y, larghezza, altezza*

x, y: coordinata superiore sinistra del rettangolo

larghezza, altezza: larghezza e altezza del rettangolo

Disegna e riempie un rettangolo il cui angolo superiore sinistro si trova in corrispondenza della coordinata specificata da (x, y)

Il colore di riempimento predefinito è il nero.
È possibile impostare il colore di riempimento con il comando [SetColor](#).

Nota: utilizzare [SetColor](#) e [SetPen](#) per specificare la linea e il colore.

FillRect 25,25,100,50



getPlatform()**Catalogo >  CXII****getPlatform()**

getPlatform()

"dt"

Restituisce:

"dt" sulle applicazioni software da desktop

"hh" sui palmari TI-Nspire™ CX

"ios" sull'app TI-Nspire™ CX iPad®

PaintBuffer**Catalogo >** 
CXII**PaintBuffer**

Disegna il buffer grafico sulla schermata.

Questo comando viene utilizzato insieme a UseBuffer per aumentare la velocità di visualizzazione su schermo quando il programma genera più oggetti grafici.

UseBuffer

```
For n,1,10
x:=randInt(0,300)
y:=randInt(0,200)
raggio:=randInt(10,50)
Wait 0,5
```

```
DrawCircle x,y,raggio
```

```
EndFor
```

PaintBuffer

Questo programma visualizzerà tutti i 10 cerchi contemporaneamente.

Se il comando “UseBuffer” viene rimosso, ciascun cerchio sarà visualizzato come è disegnato.

Vedere anche: [UseBuffer](#)

PlotXY *x, y, forma**x, y*: coordinata per tracciare la forma*forma* : un numero compreso tra 1 e 13 che specifica la forma

1 - Circonferenza piena

2 - Circonferenza vuota

3 - Quadrato pieno

4 - Quadrato vuoto

5 - Croce

6 - Più

7 - Sottile

8 - Punto medio, pieno

9 - Punto medio, vuoto

10 - Punto più grande, pieno

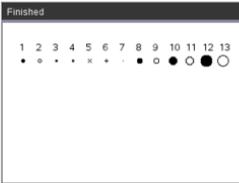
11 - Punto più grande, vuoto

12 - Il punto più grande, pieno

13 - Il punto più grande, vuoto

PlotXY *100,100,1*For *n*,1,13DrawText 1+22*n,40,*n*PlotXY 5+22*n,50,*n*

EndFor



SetColor

Catalogo >  CXII

SetColor

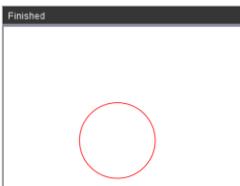
Valore rosso, verde, blu

I valori validi per il rosso, il verde e il blu sono compresi tra 0 e 255.

Imposta i colori per i comandi di Disegna successivi.

SetColor 255,0,0

DrawCircle 150,150,100

**SetPen**

Catalogo >  CXII

SetPen

spessore, stile

spessore: $1 \leq \text{spessore} \leq 3$ | 1 è il più sottile, 3 è il più spesso

stile: 1 = Uniforme, 2 = Punteggiato, 3 = Tratteggiato

Imposta lo stile della penna per i comandi di Disegna successivi.

SetPen 3,3

DrawCircle 150,150,50

**SetWindow**

Catalogo >  CXII

SetWindow

xMin, xMax, yMin, yMax

Definisce una finestra logica che esegue la mappatura all'area di disegno grafico. Tutti i parametri sono obbligatori.

Se una parte dell'oggetto disegnato si trova all'esterno della finestra, tale parte verrà tagliata (non mostrata) e non verrà visualizzato un messaggio di errore.

SetWindow 0,160,0,120

imposterà la finestra di output su 0,0 nell'angolo in basso a sinistra con larghezza di 160 e altezza di 120

DrawLine 0,0,100,100

SetWindow 0,160,0,120

SetPen 3,3

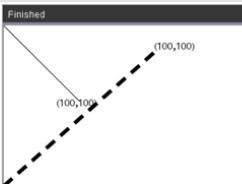
DrawLine 0,0,100,100

Se x_{Min} è maggiore o uguale a x_{Max} oppure y_{Min} è maggiore o uguale a y_{Max} , viene visualizzato un messaggio di errore.

Qualsiasi oggetto disegnato prima di un comando SetWindow non verrà tracciato nuovamente nella nuova configurazione.

Per ripristinare i valori predefiniti dei parametri della finestra, utilizzare:

SetWindow 0,0,0,0



UseBuffer**UseBuffer**

Disegna in un buffer grafico fuori schermo invece che sullo schermo (per migliorare le prestazioni).

Questo comando viene utilizzato insieme a PaintBuffer per aumentare la velocità di visualizzazione su schermo quando il programma genera più oggetti grafici.

Con UseBuffer, tutti gli elementi grafici vengono visualizzati solo dopo aver eseguito il comando PaintBuffer successivo.

Il comando UseBuffer deve essere utilizzato una sola volta nel programma, vale a dire che ogni volta che si utilizza PaintBuffer non occorre utilizzare anche UseBuffer.

UseBuffer

```
For n,1,10
x:=randInt(0,300)
y:=randInt(0,200)
```

```
raggio:=randInt(10,50)
```

```
Wait 0,5
```

```
DrawCircle x,y,raggio
```

```
EndFor
```

```
PaintBuffer
```

Questo programma visualizzerà tutti i 10 cerchi contemporaneamente.

Se il comando “UseBuffer” viene rimosso, ciascun cerchio sarà visualizzato come è disegnato.

Vedere anche: [PaintBuffer](#)

Elementi vuoti (nulli)

Quando si analizzano i dati del mondo reale, può accadere di non disporre sempre di una serie di dati completa. TI-Nspire™ consente l'uso di elementi vuoti o nulli. Ciò permette di proseguire con i dati a disposizione, anziché dover ricominciare da capo o scartare i casi incompleti.

Per un esempio di dati con elementi vuoti, vedere *"Rappresentazione grafica dei dati di un foglio di calcolo"* nel capitolo Foglio elettronico.

La funzione **delVoid()** consente di rimuovere elementi vuoti da una lista. Consente inoltre di testare un elemento vuoto. Per ulteriori informazioni, vedere **delVoid()**, pagina 40 e **isVoid()**, pagina 79.

Nota: per inserire manualmente un elemento vuoto in un'espressione matematica, digitare “_” o la parola chiave **void**. La parola chiave **void** viene convertita automaticamente in un carattere “_” quando l'espressione viene calcolata. Per digitare il carattere “_” sul palmare, premere **ctrl**  .

Calcoli con elementi vuoti

La maggior parte dei calcoli con un inserimento vuoto producono un risultato vuoto. Vedere i casi speciali sotto.

<code>_ </code>	<code>_</code>
<code>gcd(100,_)</code>	<code>_</code>
<code>3+_</code>	<code>_</code>
<code>{5,_,10}-{3,6,9}</code>	<code>{2,_,1}</code>

Argomenti di lista contenti elementi vuoti

Le funzioni e i comandi seguenti ignorano (saltano) gli elementi vuoti che trovano negli argomenti di lista.

count, countIf, cumulativeSum, freqTable>list, frequency, max, mean, median, product, stDevPop, stDevSamp, sum, sumIf, varPop e varSamp, nonché i calcoli di regressione, le statistiche **OneVar, TwoVar e FiveNumSummary**, gli intervalli di confidenza e i test statisticci.

<code>sum({2,_,3,5,6,6})</code>	16.6
<code>median({1,2,_,_,_,3})</code>	2
<code>cumulativeSum({1,2,_,4,5})</code>	{1,3,_,7,12}
<code>cumulativeSum({1,2,3,4,5})</code>	$\begin{bmatrix} 1 & 2 \\ 3 & - \\ 5 & 6 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ 4 & - \\ 9 & 8 \end{bmatrix}$

SortA e SortD spostano in fondo tutti gli elementi vuoti del primo argomento.

<code>{5,4,3,_,1} → list1</code>	<code>{5,4,3,_,1}</code>
<code>{5,4,3,2,1} → list2</code>	<code>{5,4,3,2,1}</code>
<code>SortA list1,list2</code>	<code>Done</code>
<code>list1</code>	<code>{1,3,4,5,_,}</code>
<code>list2</code>	<code>{1,3,4,5,2}</code>

Nelle regressioni, un elemento vuoto in una lista X o Y introduce un elemento vuoto corrispondente nel residuo.

$\{1,2,3,__,5\} \rightarrow list1$	$\{1,2,3,__,5\}$
$\{1,2,3,4,5\} \rightarrow list2$	$\{1,2,3,4,5\}$
SortD $list1, list2$	Done
$list1$	$\{5,3,2,1,__\}$
$list2$	$\{5,3,2,1,4\}$
$list1 := \{1,2,3,4,5\}; list2 := \{2,__,3,5,6,6\}$	
	$\{2,__,3,5,6,6\}$
LinRegMx $list1, list2$	Done
<i>stat.Resid</i>	$\{0.434286,__, -0.862857, -0.011429, 0.44\}$
<i>stat.XReg</i>	$\{1,__, 3, 4, 5\}$
<i>stat.YReg</i>	$\{2,__, 3, 5, 6, 6\}$
<i>stat.FreqReg</i>	$\{1,__, 1, 1, 1\}$

Una categoria omessa nelle regressioni introduce un elemento vuoto corrispondente nel residuo.

$list1 := \{1,3,4,5\}; list2 := \{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
$cat := \{"M", "M", "F", "F"\}; incl := \{"F"\}$	$\{"F"\}$
	$\{"F"\}$
LinRegMx $list1, list2, cat, incl$	Done
<i>stat.Resid</i>	$\{__, __, 0, 0, __\}$
<i>stat.XReg</i>	$\{__, __, 4, 5\}$
<i>stat.YReg</i>	$\{__, __, 5, 6, 6\}$
<i>stat.FreqReg</i>	$\{__, __, 1, 1, 1\}$

Una frequenza 0 nelle regressioni introduce un elemento vuoto corrispondente nel residuo.

$list1 := \{1,3,4,5\}; list2 := \{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx $list1, list2, \{1,0,1,1\}$	Done
<i>stat.Resid</i>	$\{0.069231,__, -0.276923, 0.207692\}$
<i>stat.XReg</i>	$\{1,__, 4, 5\}$
<i>stat.YReg</i>	$\{2,__, 5, 6, 6\}$
<i>stat.FreqReg</i>	$\{1,__, 1, 1, 1\}$

Scelte rapide per l'inserimento di espressioni matematiche

Le scelte rapide permettono di inserire elementi di espressioni matematiche per digitazione invece di usare il Catalogo o la tavolozza Simboli. Per esempio, per inserire l'espressione $\sqrt{6}$, è possibile digitare `sqrt(6)` nella riga di introduzione. Quando si preme `enter`, l'espressione `sqrt(6)` viene modificata in $\sqrt{6}$. Alcune scelte rapide sono utili sia per il palmare sia per la tastiera del computer. Altre sono utili principalmente dalla tastiera del computer.

Dalla tastiera del palmare o del computer

Per inserire questo:	digitare questa scelta rapida:
π	<code>pi</code>
θ	<code>theta</code>
∞	<code>infinity</code>
\leq	<code><=</code>
\geq	<code>>=</code>
\neq	<code>/=</code>
\Rightarrow (implicazione logica)	<code>=></code>
\Leftrightarrow (doppia implicazione logica, XNOR)	<code><=></code>
\rightarrow (memorizza operatore)	<code>=:</code>
$ $ (valore assoluto)	<code>abs(...)</code> (Valore assoluto)
$\sqrt()$	<code>sqrt(...)</code> (Radice quadrata)
$\Sigma()$ (Modello di somma)	<code>sumSeq(...)</code>
$\Pi()$ (Modello di prodotto)	<code>prodSeq(...)</code>
$\sin^{-1}()$, $\cos^{-1}()$, ...	<code>arcsin(...)</code> , <code>arccos(...)</code> , ...
Δ lista()	<code>deltaList(...)</code>

Dalla tastiera del computer

Per inserire questo:	digitare questa scelta rapida:
i (unità immaginaria)	<code>@i</code>
e (logaritmo naturale in base e)	<code>@e</code>
E (notazione scientifica)	<code>@E</code>
\top (trasposizione)	<code>@t</code>

Per inserire questo:	digitare questa scelta rapida:
Γ (radiani)	@r
$^\circ$ (Gradi)	@d
G gradianti	@g
\angle (angolo)	@<
► (conversione)	@>
►Decimal, ►approxFraction (), ecc.	@>Decimal, @>approxFraction(), ecc. (), ecc.

EOS™ (Equation Operating System) gerarchia

Questa sezione descrive il sistema EOS™ (Equation Operating System) utilizzato da TI-Nspire™ Sistema integrato per la matematica e le scienze sperimentali. Numeri, variabili e funzioni vengono introdotte in successione. Il software EOS™ calcola le espressioni e le equazioni utilizzando raggruppamenti racchiusi tra parentesi e in base alle priorità descritte sotto.

Ordine di Valutazione

Livello	Operatore
1	Parentesi tonde (), parentesi quadre [], parentesi graffe { }
2	Conversione indiretta (#)
3	Chiamate di funzione
4	Post operatori: gradi-primi-secondi($^{\circ},'$), fattoriale (!), percentuale (%), radiante (Γ), pedice ([]), trasposizione (T)
5	Elevamento a potenza, operatore di potenza (^)
6	Segno negativo (-)
7	Concatenazione di stringhe (&)
8	Moltiplicazione (\cdot), divisione (/)
9	Addizione (+), sottrazione (-)
10	Rapporti di uguaglianza: uguale (=), non uguale (\neq o \neq), minore di (<), minore di o uguale a (\leq o \leq), maggiore di (>), maggiore di o uguale a (\geq o \geq)
11	not logico
12	and logico
13	Logico or
14	xor, nor, nand
15	Implicazione logica (\Rightarrow)
16	Doppia implicazione logica, XNOR (\Leftrightarrow)
17	Operatore di vincolo (" ")
18	Memorizzazione (\rightarrow)

Parentesi tonde, quadre e graffe

Tutti i calcoli racchiusi tra parentesi tonde, quadre o graffe vengono eseguiti per primi. Ad esempio, nell'espressione $4(1+2)$, EOS™ calcola prima la parte di espressione racchiusa tra parentesi tonde, $1+2$, quindi moltiplica il risultato, 3, per 4.

In un'espressione o in un'equazione, tutte le parentesi tonde, quadre e graffe aperte devono essere chiuse. Diversamente, viene visualizzato un messaggio di errore a indicare l'elemento mancante. Ad esempio, $(1+2)/(3+4$ produrrà un messaggio di errore "Manca)."

Nota: poiché il software TI-Nspire™ consente di definire funzioni personalizzate, un nome di variabile seguito da un'espressione racchiusa tra parentesi viene considerato una "chiamata di funzione" e non una moltiplicazione implicita. Ad esempio $a(b+c)$ è la funzione a calcolata per $b+c$. Per moltiplicare l'espressione $b+c$ per la variabile a, utilizzare la moltiplicazione esplicita: $a*(b+c)$.

Conversione indiretta

L'operatore di conversione indiretta (#) converte una stringa in un nome di variabile o di funzione. Ad esempio, $\#("x"&"y"&"z")$ crea il nome di variabile xyz. La conversione indiretta consente inoltre di creare e modificare variabili dall'interno di un programma. Ad esempio, se $10\rightarrow r$ e $"r"\rightarrow s1$, allora $\#s1=10$.

Post operatori

I post operatori sono operatori che vengono inseriti immediatamente dopo un argomento, come ad esempio $5!$, 25% o $60^{\circ}15'45''$. Gli argomenti seguiti da un post operatore vengono calcolati al quarto livello di priorità. Ad esempio, nell'espressione $4^3!$, $3!$ viene calcolato per primo. Il risultato, 6, diventa quindi l'esponente di 4 che darà come risultato 4096.

Elevazione a potenza

L'elevamento a potenza (^) e l'elevazione a potenza elemento per elemento (.[^]) vengono calcolati da destra a sinistra. Ad esempio, l'espressione 2^3^2 viene calcolata allo stesso modo di $2^(3^2)$ dando come risultato 512. Ciò è diverso da $(2^3)^2$, che dà come risultato 64.

Segno negativo

Per introdurre un numero negativo, premere [(-) quindi il numero. Le post operazioni e l'elevamento a potenza vengono eseguiti prima dell'operazione di cambiamento di segno. Ad esempio, il risultato di $-x^2$ è un numero negativo e $-9^2 = -81$. Utilizzare le parentesi per elevare al quadrato un numero negativo, come ad esempio $(-9)^2$ che dà come risultato 81.

Vincolo (“|”)

L'argomento che segue l'operatore di vincolo (“|”) fornisce una serie di restrizioni che influiscono sul calcolo dell'argomento che precede l'operatore.

TI-Nspire CX II - Funzioni di programmazione di TI-Basic

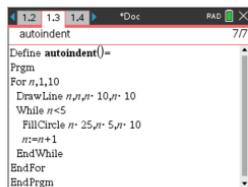
Rientro automatico in Program Editor

Program Editor di TI-Nspire™ esegue ora il rientro automatico delle istruzioni all'interno di un comando di blocco.

I comandi di blocco sono If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry.

L'editor anteporrà automaticamente gli spazi per programmare i comandi all'interno di un comando di blocco. Il comando di chiusura del blocco verrà allineato con il comando di apertura.

Di seguito viene mostrato un esempio di rientro automatico in comandi di blocco nidificati.



```
1.2 | 1.3 | 1.4 *Dot RAD X
autoident
Define autoident()=
Prgm
For n,1,10
DrawLine n,n,n- 10,n- 10
While n<5
  FillCircle n- 25,n- 5,n- 10
  n:=n+1
EndWhile
EndFor
EndPrgm
```

I frammenti di codice copiati e incollati manterranno il rientro originale.

Se si apre un programma creato in una versione precedente del software, verrà mantenuto il rientro originale.

Miglioramento dei messaggi di errore per TI-Basic

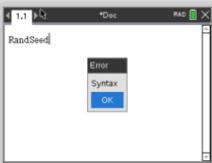
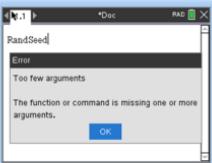
Errori

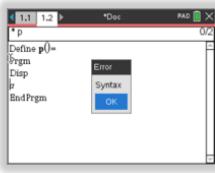
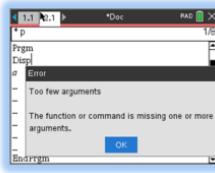
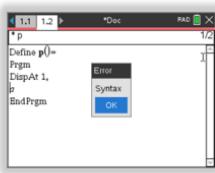
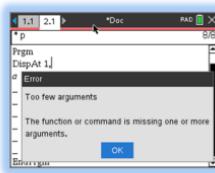
Condizione di errore	Nuovo messaggio
Errore nell'istruzione condizionale (If/While)	Un'istruzione condizionale non ha restituito VERO o FALSO NOTA: con la modifica che stabilisce di posizionare il cursore sulla linea con l'errore, non è più necessario specificare se l'errore si trova in un'istruzione "If" o in un'istruzione "While".
EndIf mancante	Trovata istruzione end diversa dall'istruzione EndIf attesa
EndFor mancante	Trovata istruzione end diversa dall'istruzione EndFor attesa
EndWhile mancante	Trovata istruzione end diversa dall'istruzione EndWhile attesa

Condizione di errore	Nuovo messaggio
EndLoop mancante	Trovata istruzione end diversa dall'istruzione EndLoop attesa
EndTry mancante	Trovata istruzione end diversa dall'istruzione EndTry attesa
“ Then ” omessa dopo If <condition>	If..Then mancante
“ Then ” omessa dopo ElseIf <condition>	Manca Then nel blocco: ElseIf .
Quando sono stati incontrate “ Then ”, “ Else ” ed “ ElseIf ” al di fuori dei blocchi di controllo	Else non valida fuori dai blocchi: If..Then..Endif o Try..EndTry
“ ElseIf ” appare al di fuori del blocco “ If..Then..Endif ”	ElseIf non valida fuori dal blocco: If..Then..Endif
“ Then ” appare al di fuori del blocco “ If....Endif ”	Then non valida fuori dal blocco: If..Endif

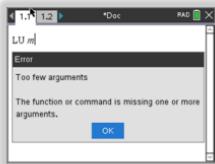
Errori di sintassi

Nel caso in cui dei comandi che devono contenere uno o più argomenti vengono chiamati con un elenco di argomenti incompleto, verrà generato il messaggio di errore “**Argomenti mancanti**” invece di un errore “**Sintassi**”.

Comportamento corrente	Nuovo comportamento CX II
	
	

Comportamento corrente	Nuovo comportamento CX II
	
	

Nota: quando un elenco di argomenti incompleto non viene seguito da una virgola, viene visualizzato il messaggio di errore: “Argomenti mancanti”. Ciò si verifica anche nelle versioni precedenti.



Costanti e valori

Nella seguente tabella sono elencate le costanti con i rispettivi valori disponibili durante l'esecuzione delle conversioni di unità. Possono essere digitate manualmente o selezionate dall'elenco **Costanti** in **Utilità > Conversioni di unità** (palmare: premere  3).

Costante	Nome	Valore
_c	Velocità della luce	299792458 _m/_s
_Cc	Costante di coulomb	8987551792.261 _m/_F
_Fc	Costante di Faraday	96485.33212 _coul/_mol
_g	Accelerazione di gravità	9.80665 _m/_s ²
_Gc	Costante gravitazionale	6.6743E-11 _m ³ /_kg/_s ²
_h	Costante di Planck	6.62607015E-34 _J_s
_k	Costante di Boltzmann	1.380649E-23 _J/_°K
_μ0	Permeabilità del vuoto	1.25663706212E-6 _N/_A ²
_μb	Magnetone di Bohr	9.27400994E-24 _J_m ² /_Wb
_Me	Massa a riposo dell'elettrone	9.1093837015E-31 _kg
_Mμ	Massa del muone	1.883531627E-28 _kg
_Mn	Massa a riposo del neutrone	1.67492749804E-27 _kg
_Mp	Massa a riposo del protone	1.67262192369E-27 _kg
_Na	Numero di Avogadro	6.02214076E23 /_mol
_q	Carica dell'elettrone	1.602176634E-19 _coul
_Rb	Raggio di bohr	5.29177210903E-11 _m
_Rc	Costante molare del gas	8.314462618 _J/_mol/_°K
_Rdb	costante di Rydberg	10973731.568160/_m
_Re	Raggio dell'elettrone	2.8179403262E-15 _m
_u	Massa atomica	1.6605390666E-27 _kg
_Vm	Volume molare	2.241396954E-2 _m ³ /_mol
_ε0	Permittività del vuoto	8.8541878128E-12 _F/_m
_σ	Costante di Stefan-Boltzmann	5.670367E-8 _W/_m ² /_°K ⁴
_Φ0	Quanto di flusso magnetico	2.067833831E-15 _Wb

Codici di errore e messaggi

Quando si produce un errore, il relativo codice viene assegnato alla variabile *errCode*. Programmi e funzioni definite dall'utente possono esaminare *errCode* per determinare la causa dell'errore. Per un esempio dell'uso di *errCode*, vedere l'Esempio 2 del comando **Try** (pagina 166).

Nota: alcune condizioni di errore si riferiscono solo ai prodotti TI-Nspire™ CAS, mentre altre solo ai prodotti TI-Nspire™.

Codice errore	Descrizione
10	Una funzione non ha restituito un valore
20	Una prova non ha saputo stabilire se VERO o FALSO. In generale, non è possibile confrontare variabili non definite. Ad esempio, la prova If $a < b$ causerà questo errore se o a o b sono indefiniti al momento dell'esecuzione dell'istruzione If.
30	Argomento non può essere un nome di cartella.
40	Argomento errato
50	Argomento di tipo errato Due o più argomenti devono essere dello stesso tipo.
60	Argomento deve essere un'espressione booleana o un numero intero
70	Argomento deve essere un numero decimale
90	Argomento deve essere una lista
100	Argomento deve essere una matrice
130	Argomento deve essere una stringa
140	Argomento deve essere un nome di variabile. Accertarsi che il nome: <ul style="list-style-type: none">• non inizi con un numero• non contenga spazi o caratteri speciali• non utilizzi trattini bassi o virgole in modo non valido• non superi i limiti di lunghezza Per ulteriori informazioni, vedere il capitolo Calcolatrice nella documentazione.
160	Argomento deve essere un'espressione
165	Carica delle batterie insufficiente per inviare/ricevere Inserire nuove batteria prima di inviare o ricevere.
170	Estremo

Codice errore	Descrizione
	L'estremo inferiore deve essere minore dell'estremo superiore per definire l'intervallo di ricerca.
180	<p>Interruzione</p> <p>È stato premuto il tasto [esc] o [fn]+[on] durante un calcolo lungo o durante l'esecuzione del programma.</p>
190	<p>Definizione circolare</p> <p>Questo messaggio viene visualizzato per evitare l'esaurimento della memoria durante la sostituzione infinita di valori di variabile nel corso di una semplificazione. Ad esempio, $a+1>a$, dove a è una variabile indefinita, causerà questo errore.</p>
200	<p>Condizione non valida</p> <p>Ad esempio, $\text{solve}(3x^2-4=0,x) \mid x<0 \text{ or } x>5$ produrrebbero questo messaggio di errore perché il vincolo è separato da "or" invece che da "and."</p>
210	<p>Tipo di dati non valido</p> <p>Un argomento è un tipo di dati sbagliato.</p>
220	Limite dipendente
230	<p>Dimensione</p> <p>Un indice di lista o di matrice non è valido. Ad esempio, se la lista $\{1,2,3,4\}$ è memorizzata in $L1$, allora $L1[5]$ è un errore di dimensione perché $L1$ contiene solo quattro elementi.</p>
235	Errore di dimensione. Elementi insufficienti nelle liste.
240	<p>Dimensioni non corrispondenti</p> <p>Due o più argomenti devono essere della stessa dimensione. Ad esempio, $[1,2]+[1,2,3]$ è un errore di dimensioni non corrispondenti perché le matrici contengono un numero diverso di elementi.</p>
250	Divisione per zero
260	<p>Argomento errato</p> <p>Deve essere presente un argomento in un dominio specificato. Ad esempio $\text{rand}(0)$ non è valido.</p>
270	Nome di variabile duplicato
280	Else ed Elself non validi fuori dal blocco If..EndIf
290	EndTry non trova la corrispondente istruzione Else

Codice errore	Descrizione
295	Numero eccessivo di iterazioni
300	Attesa lista o matrice a 2 o 3 elementi
310	Il primo argomento di nSolvedeve essere un'equazione a una sola variabile. Non può contenere una variabile non calcolata che non sia la variabile cercata.
320	Il primo argomento di solve o cSolve deve essere un'equazione o una disequazione. Ad esempio, solve($3x^2-4,x$) non è valido perché il primo argomento non è un'equazione.
345	Unità di misura non coerenti
350	Indice non valido
360	La stringa da convertire non è un nome di variabile valido
380	Risultato indefinito Il precedente calcolo non ha prodotto un risultato oppure non è stato inserito alcun calcolo precedente.
390	Assegnazione non valida
400	Valore di assegnazione non valido
410	Comando non valido
430	Non valido per le impostazioni di modo correnti
435	Ipotesi non valida
440	Moltiplicazione sottintesa non valida Ad esempio, $x(x+1)$ non è valido, mentre $x*(x+1)$ è la sintassi corretta. Ciò per evitare confusione tra chiamate di funzione e moltiplicazioni implicite.
450	Non valido in una funzione o nella attuale espressione In una funzione definita dall'utente sono validi solo certi comandi.
490	Non valido nel blocco Try..EndTry
510	Lista o matrice non valida
550	Non valido fuori da una funzione o un programma Alcuni comandi non sono validi fuori da una funzione o un programma. Ad esempio, Local non può essere utilizzato in una funzione o in un programma.
560	Non valido fuori dai blocchi Loop..EndLoop, For..EndFor o While..EndWhile Ad esempio, il comando Exit è valido solo in questi blocchi loop.

Codice errore	Descrizione
565	Non valido fuori da un programma
570	Nome di percorso non valido Ad esempio, \var non è valido.
575	Numero complesso in forma polare non valido
580	Chiamata di programma non valida Non è possibile chiamare un programma all'interno di funzioni o espressioni quali $1+p(x)$, dove p è un programma.
600	Tabella non valida
605	Uso di unità non valido
610	Nome di variabile non valido in una istruzione Local
620	Nomi di variabile o di funzione non valido
630	Chiamata di variabile non valida
640	Sintassi del vettore non valida
650	Errore di collegamento Trasmissione non completata tra due unità. Verificare che il cavo di collegamento sia inserito correttamente ad entrambe le estremità.
665	Matrice non diagonizzabile
670	Memoria quasi esaurita 1. Eliminare alcuni dati in questo documento 2. Salvare e chiudere questo documento Se le istruzioni 1 e 2 non producono l'esito sperato, togliere e rimettere le batterie
672	Risorsa esaurita
673	Risorsa esaurita
680	Manca (
690	Manca)
700	Manca "
710	Manca]
720	Manca }

Codice errore	Descrizione
730	Manca inizio o fine del blocco
740	Manca Then nel blocco If..EndIf
750	Il nome non è una funzione o un programma
765	Nessuna funzione selezionata
780	Nessuna soluzione trovata
800	<p>Risultato non reale</p> <p>Ad esempio, se il software è impostato su Real, $\sqrt{(-1)}$ non è valido.</p> <p>Per consentire risultati complessi, modificare l'impostazione di modalità "Real or Complex" in RETTANGOLARE (RECTANGULAR) o POLARE (POLAR).</p>
830	Superamento della memoria
850	<p>Programma non trovato</p> <p>Impossibile trovare una chiamata di programma all'interno di un altro programma nel percorso specificato durante l'esecuzione.</p>
855	Funzione di tipo Rand non consentita nel grafico
860	Ricorsione troppo profonda
870	Nome riservato o variabile di sistema
900	<p>Argomento errato</p> <p>Impossibile applicare al set di dati il modello mediana-mediana.</p>
910	Errore di sintassi
920	Testo non trovato
930	<p>Argomenti mancanti</p> <p>Nella funzione o nel comando mancano uno o più argomenti.</p>
940	<p>Troppi argomenti</p> <p>L'espressione o equazione contiene un numero eccessivo di argomenti e non può essere calcolata.</p>
950	Troppi indici
955	Troppe variabili non definite
960	<p>La variabile non è definita</p> <p>Nessun valore assegnato alla variabile. Utilizzare uno dei seguenti comandi:</p>

Codice errore	Descrizione
	<ul style="list-style-type: none"> • sto \rightarrow • := • Define <p>per assegnare valori alle variabili.</p>
965	SO privo di licenza
970	Variabile in uso, di conseguenza non sono consentite chiamate e modifiche
980	La variabile è protetta
990	<p>Nome variabile non valido</p> <p>Accertarsi che il nome non superi i limiti di lunghezza</p>
1000	Valori dei parametri di Window
1010	Zoom
1020	Errore interno
1030	Violazione della memoria protetta
1040	Funzione non supportata. Questa funzione richiede il sistema CAS (Computer Algebra System). Provare TI-Nspire™ CAS.
1045	Operatore non supportato. Questo operatore richiede il sistema CAS (Computer Algebra System). Provare TI-Nspire™ CAS.
1050	Funzione non supportata. Questo operatore richiede il sistema CAS (Computer Algebra System). Provare TI-Nspire™ CAS.
1060	L'argomento dell'inserimento deve essere numerico. Sono consentiti solo inserimenti contenenti valori numerici.
1070	L'argomento della funzione trigonometrica è troppo grande per una riduzione precisa.
1080	Uso non supportato di Ans. Questa applicazione non supporta Ans.
1090	<p>La funzione non è definita. Utilizzare uno dei seguenti comandi:</p> <ul style="list-style-type: none"> • Define • := • sto \rightarrow <p>per definire una funzione.</p>
1100	<p>Calcolo non reale</p> <p>Ad esempio, se il software è impostato su "Real" $\sqrt{(-1)}$ non è valido.</p>

Codice errore	Descrizione
	Per consentire risultati complessi, modificare l'impostazione di modalità "Real or Complex" in RETTANGOLARE (RECTANGULAR) o POLARE (POLAR).
1110	Estremi non validi
1120	Nessun cambio di segno
1130	Argomento non può essere una lista o una matrice
1140	<p>Argomento errato</p> <p>Il primo argomento deve essere un'espressione polinomiale nel secondo argomento. Se il secondo argomento viene omesso, il software tenta di selezionare un'impostazione predefinita.</p>
1150	<p>Argomento errato</p> <p>I primi due argomenti devono essere espressioni polinomiali nel terzo argomento. Se il terzo argomento viene omesso, il software tenta di selezionare un'impostazione predefinita.</p>
1160	<p>Nome di percorso libreria non valido</p> <p>Un nome di percorso deve avere la forma <code>xxx\yyy</code>, dove:</p> <ul style="list-style-type: none"> • La parte <code>xxx</code> può avere da 1 a 16 caratteri. • La parte <code>yyy</code> può avere da 1 a 15 caratteri. <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1170	<p>Uso di nome percorso libreria non valido</p> <ul style="list-style-type: none"> • Impossibile assegnare un valore a un nome percorso utilizzando Define, <code>:=</code> o <code>sto →</code>. • Impossibile dichiarare un nome percorso come variabile Local oppure utilizzarlo come parametro in una definizione di funzione o programma.
1180	<p>Nome variabile libreria non valido.</p> <p>Accertarsi che il nome:</p> <ul style="list-style-type: none"> • non contenga un punto • non inizi con un trattino basso • non contenga più di 15 caratteri <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1190	<p>Documento Libreria non trovato:</p> <ul style="list-style-type: none"> • Verificare che la libreria sia nella cartella <code>MieLibrerie</code>. • Aggiornare le librerie. <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>

Codice errore	Descrizione
1200	<p>Variabile libreria non trovata:</p> <ul style="list-style-type: none"> • Verificare che la variabile libreria esista nella prima attività nella libreria. • Accertarsi che la variabile libreria sia stata definita come LibPub o LibPriv. • Aggiornare le librerie. <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1210	<p>Nome collegamento libreria non valido.</p> <p>Accertarsi che il nome:</p> <ul style="list-style-type: none"> • non contenga un punto • non inizi con un trattino basso • non contenga più di 16 caratteri • non sia un nome riservato <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
1220	<p>Errore di dominio.</p> <p>Le funzioni Retta tangente e Retta normale supportano solo funzioni con valori reali.</p>
1230	<p>Errore di dominio.</p> <p>Gli operatori di conversione trigonometrica non sono supportati nelle modalità in gradi o gradianti.</p>
1250	<p>Argomento errato</p> <p>Utilizza un sistema di equazioni lineari.</p> <p>Esempio di sistema di due equazioni lineari con variabili x e y:</p> $3x+7y=5$ $2y-5x=-1$
1260	<p>Argomento errato:</p> <p>Il primo argomento di nfMin o nfMax deve essere un'espressione a una sola variabile. Non può contenere una variabile non calcolata che non sia la variabile cercata.</p>
1270	<p>Argomento errato</p> <p>L'ordine della derivata deve essere uguale a 1 o 2.</p>
1280	<p>Argomento errato</p> <p>Utilizzare un polinomio in forma normale in una sola variabile.</p>

Codice errore	Descrizione
1290	Argomento errato Utilizzare un polinomio in una sola variabile.
1300	Argomento errato I coefficienti del polinomio devono dare come risultato valori numerici.
1310	Argomento errato: Non è stato possibile calcolare una funzione per uno o più dei suoi argomenti.
1380	Argomento errato: Non sono autorizzate le chiamate nidificate alla funzione domain().

Codici di avvertenza e messaggi

È possibile utilizzare la `warnCodes()` funzione per memorizzare i codici delle avvertenze generate dal calcolo di un'espressione. Questa tabella riporta ciascun codice numerico e il relativo messaggio associato. Per un esempio di memorizzazione dei codici di avvertenza, vedere `warnCodes()`, pagina 175.

Codice di avvertenza	Messaggio
10000	L'operazione può introdurre soluzioni false Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10001	Derivare un'equazione può produrre un'equazione falsa
10002	Soluzione dubbia Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10003	Accuratezza dubbia Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10004	L'operazione può far perdere soluzioni Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10005	cSolve può specificare più zeri
10006	Solve può specificare più zeri Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10007	È possibile che ci siano altre soluzioni. Provare a definire i valori inferiori e superiori adeguati e/o una ipotesi. Esempi che usano solve(): <ul style="list-style-type: none">• <code>solve(Equation, Var=Approssimativo) lowBound<Var<upBound</code>• <code>solve(Equation, Var) lowBound<Var<upBound</code>• <code>solve(Equation,Var=Approssimativo)</code> Se applicabile, provare a utilizzare metodi grafici per verificare i risultati.
10008	Il dominio del risultato può essere minore del dominio dell'introduzione.
10009	Il dominio del risultato può essere maggiore del dominio dell'introduzione.
10012	Calcolo non reale
10013	∞^0 o indef^0 sostituito da 1
10014	indef^0 sostituito da 1
10015	1^∞ o 1^indef sostituito da 1

Codice di avvertenza	Messaggio
10016	1 ^{indef} sostituito da 1
10017	Overflow sostituito da ∞ o $-\infty$
10018	L'operazione richiede e restituisce un valore a 64 bit
10019	Risorsa esaurita, semplificazione forse incompleta
10020	L'argomento della funzione trigonometrica è troppo grande per una riduzione precisa
10021	L'introduzione contiene un parametro non definito. Il risultato potrebbe non essere valido per tutti i valori di parametro possibili.
10022	La specifica di estremi inferiore e superiore appropriati può produrre una soluzione.
10023	Lo scalare è stato moltiplicato per la matrice identità.
10024	Risultato ottenuto utilizzando aritmetica approssimata.
10025	L'equivalenza non può essere verificata in modalità EXACT (Esatta).
10026	È possibile ignorare il vincolo. Specificare un vincolo nella forma "\'Costante variabile MathTestSymbol' oppure un insieme di queste forme, ad esempio 'x<3 and x>-12'

Informazioni Generali

Guida online

education.ti.com/eguide

Selezionare il proprio Paese per maggiori informazioni sul prodotto.

Contattare l'assistenza TI

education.ti.com/ti-cares

Selezionare il proprio Paese per assistenza tecnica e altre risorse.

Informazioni su servizi e garanzia

education.ti.com/warranty

Selezionare il proprio Paese per informazioni sulla durata e sui termini della garanzia o sull'assistenza ai prodotti.

Garanzia limitata. La presente garanzia non pregiudica i diritti spettanti per legge.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Indice

-		→ designazione di unità	201
-		-	
-		-, operatore di vincolo	202
!		-	
!, fattoriale	194	+, addizione	185
"		-	
", notazione in secondi	200	/, divisione[*]	186
#		-	
#, conversione indiretta	198	≠, diverso[*]	190
#, operatore conversione indiretta	227	=, uguale	189
%		-	
%, percentuale	189	>, maggiore di	192
&		-	
&, aggiunge	194	Π, prodotto[*]	194
*		-	
-, moltiplicazione	186	Σ(), somma[]	195
,		ΣInt()	196
, notazione in primi	200	ΣPrn()	197
•		-	
•-, punto sottrazione	188	√, radice quadrata[*]	194
•*, punto moltiplicazione	188	≤	
./, punto divisione	188	≤, minore di o uguale a	191
.^, punto elevato a potenza	189	≥	
.+, punto addizione	188	≥, maggiore di o uguale a	192
:		▶	
:=, assegna	204	▶, converti in angolo in gradianti	
^		▶ [Grad]	71
^-1, reciproco	202	▶approxFraction()	13
^, elevamento a potenza	186		

	A
►Base10, visualizza come numero decimale[Base10]	18
►Base16, visualizza come esadecimale[Base16]	19
►Base2, visualizza come binario [Base2]	17
►Cylind, visualizza come vettore in forma cilindrica[Cylind]	35
►DD, visualizza angolo decimale[DD]	
►Decimal, visualizza il risultato nella forma decimale[Decimal] ..	37
►DMS, visualizza come gradi/primi/secondi[DMS] ..	44
►Polar, visualizza come vettore polare[Polare]	117
►Rad, converti in angolo in radianti ..	
►Rect, visualizza come vettore rettangolare	129
►Sphere, visualizza come vettore sferico[Sphere]	153
⇒	
⇒, implicazione logica[*]	193, 224
→	
→, memorizza	203
↔	
↔, doppia implicazione logica[*] ...	193
©	
©, commento	204
°	
°, gradi/primi/secondi[*]	200
°, notazione in gradi[*]	200
0	
0b, indicatore binario	204
0h, indicatore esadecimale	204
1	
10^(), potenza di dieci	202
abs(), valore assoluto	7
addizione, +	185
affianca/concatena, augment() ..	15
aggiunge, &	194
all'interno della stringa, inString() ..	74
altrimenti, Else	72
amortTbl(), tabella di ammortamento	7, 16
and, operatore boolean	8
angle(), angolo	9
angolo, angle()	9
ANOVA, analisi della varianza a una variabile	9
ANOVA2way, analisi della varianza a due dimensioni	10
Ans, ultimo risultato	12
approssima, approx()	13
approx(), approssima	13
approxRational()	13
arccos()	14
arccosh()	14
arccot()	14
arccoth()	14
arccsc()	14
arccsch()	14
arcocoseno, $\cos^{-1}()$	28
arcoseno, $\sin^{-1}()$	149
arcotangente, $\tan^{-1}()$	161
arcsec()	14
arcsech()	14
arcsin()	14
arcsinh()	14
arctan()	15
arctanh()	15
argomenti nelle funzioni TVM	170
argomenti TVM	170
arrotondamento, round()	138
arrotondato per difetto, floor()	55
arrotondato per eccesso, ceiling() ..	20-21, 32
augment(), affianca/concatena	15
autovalore, eigVl()	46
autovettore, eigVc()	46
avgRC(), tasso di variazione media ..	15

B

binario	
indicatore, 0b	204
visualizza, ►Base2	17
binomCdf() (Funzione della	
probabilità cumulativa per	
la distribuzione binomiale)	20
binomPdf()	20
blocco di variabile e gruppi di	
variabili	90

C

calcola polinomio, polyEval()	118
calcolo, ordine di	226
campione casuale	128
cancella	
errore, ClrErr	24
Cancellare	209
caratteri	
codice di carattere, ord()	115
stringa, char()	21
Cdf()	53
ceiling(), arrotondato per eccesso ..	20
centralDiff()	21
char(), stringa di caratteri	21
ClearAZ	24
ClrErr, cancella errore	24
codici di avvertenza e messaggi	241
Codici di errore e messaggi	241
colAugment	25
colDim(), dimensione colonna di	
matrice	25
colNorm(), norma colonna di	
matrice	25
comando Stop	158
Comando Wait	175
combinazioni, nCr()	105
commento, ©	204
complesso	
coniugato, conj()	25
con, 	202
conj(), complesso coniugato	25
constructMat(), Costruisci matrice	26
conteggio condizionato elementi in	
una lista, countif()	31
conteggio elementi in una lista,	
count()	31

conteggio giorni tra le date, dbd()	36
conversione indiretta, #	198
converti	
►Rad	125
4Grad	71
coordinata polare, R►Pr()	125
coordinata polare, R►Pθ()	125
coordinata x rettangolare, P►Rx()	115
coordinata y rettangolare, P►Ry()	115
copia variabile o funzione, CopyVar	26
corrMat(), matrice di correlazione	27
cos ⁻¹ , arcocoseno	28
cos(), coseno	27
coseno, cos()	27
cosh ⁻¹ (), arcocoseno iperbolico	29
cosh(), coseno iperbolico	29
Costruisci matrice, constructMat()	26
cot ⁻¹ (), arcocotangente	30
cot(), cotangente	30
cotangente, cot()	30
coth ⁻¹ (), arcocotangente iperbolica	31
coth(), cotangente iperbolica	30
count(), conteggio elementi in una	
lista	31
countif(), conteggio condizionato	
elementi in una lista	31
cPolyRoots()	32
crossP(), prodotto vettoriale	32
csc ⁻¹ (), cosecante inversa	33
csc(), cosecante	33
csch ⁻¹ (), cosecante iperbolica	
inversa	33
csch(), cosecante iperbolica	33
CubicReg, regressione cubica	34
cumulativeSum(), somma	
cumulativa	35
Cycle, ripeti	35
D	
d(), prima derivata	194
da lista a matrice, list►mat()	88
da matrice a lista, mat►list()	95
dbd(), giorni tra le date	36
decimale	
visualizza angolo, ►DD	36
visualizza intero, ►Base10	18
Define	37
Define LibPriv	38

Define LibPub	39	effettivo	
Define, definisci	37	eigVc(), autovettore	46
definisci, Define	37	eigVl(), autovalore	46
definizione		elementi vuoti	222
funzione o programma privato	38	elementi vuoti (nulli)	222
programma o funzione pubblica	39	elementi vuoti, rimuovi	40
deltaList()	39	elevamento a potenza, \wedge	186
DelVar, elimina variabile	39	elimina	
delVoid(), rimuovi elementi vuoti	40	elementi vuoti dalla lista	40
densità di probabilità t di Student, tPdf()	165	eliminazione	
densità di probabilità, normPdf()	109	variabile, DelVar	39
derivata prima		else if, ElseIf	47
modello di	5	ElseIf, else if	47
derivata seconda		end	
modello di	5	loop, EndLoop	94
derivate		per, EndFor	56
derivata numerica, nDeriv()	106	EndTry, fine tentativo	166
derivata numerica, nDerivative()	105	EndWhile, fine mentre	177
prima derivata, d()	194	EOS (Equation Operating System)	226
destra, right()	75, 135	Equation Operating System (EOS)	226
det(), determinante matrice	40	equazioni simultanee, simult()	148
deviazione standard, stdDev()	156-157, 173	errori e soluzione dei problemi	
diag(), diagonale matrice	41	cancella errore, ClrErr	24
dim(), dimensione	41	passa errore, PassErr	116
dimensione, dim()	41	esadecimale	
dimensioni di righe di matrice, rowDim()	139	indicatore, Oh	204
disegna	210-212	visualizza, ►Base16	19
Disp, visualizza dati	42	esci, Exit	50
DispAt	42	esclusione con l'operatore "!"	202
disposizioni semplici, nPr()	110	esponente, E	198
distribuzione normale cumulativa		esponenti	
inversa (invNorm()	77	modello di	1
diverso, \neq	190	espressioni	
divisione intera, intDiv()	75	da stringa a espressione, expr()	51
divisione, P	186	etichetta, Lbl	80
doppia implicazione logica, \Leftrightarrow	193	euler(), Euler function	48
dot		Exit, esci	50
prodotto, dotP()	44	exp(), e alla potenza	50
dotP(), prodotto scalare	44	expr(), da stringa a espressione	51
E			
e alla potenza, e $^$ ()	50	ExpReg, regressione esponenziale	51
E, esponente	198	F	
e $^$ (), funzione esponenziale	45	F Test su due campioni	59
eff, converti da tasso nominale a	45	factor(), fattorizza	53
		fattoriale, !	194
		fattorizza, factor()	53
		Fdrbinom()	77
		Fill, riempì matrice	54

fine			
funzione, EndFunc	60	normCdf()	109
mentre, EndWhile	177	normPdf()	109
tentativo, EndTry	166	poissCdf()	117
fine funzione, EndFunc	60	poissPdf()	117
fine loop, EndLoop	94	tCdf()	163
fine mentre, EndWhile	177	tPdf()	165
fine se, EndIf	72	χ^2 way()	22
FiveNumSummary	54	χ^2 Cdf()	22
floor(), arrotondato per difetto	55	χ^2 GOF()	23
For	56	χ^2 Pdf()	24
For, per	56	funzioni e programmi definiti	
forma a scalini per righe, ref()	54	dallutente	38-39
forma a scalini ridotta per righe	55	funzioni e variabili	
matrici, rref()	56	copia	26
forma a scalini ridotta per righe, rref	57	140 funzioni finanziarie, tvmFV()	169
()	57	140 funzioni finanziarie, tvmI()	169
format(), stringa formato	56	140 funzioni finanziarie, tvmN()	170
fpart(), funzione parte frazionaria	57	140 funzioni finanziarie, tvmPmt()	170
frazione propria, propFrac	121	121 funzioni finanziarie, tvmPV()	170
frazioni			
frzProp	121	G	
modello di	1	G, gradienti	199
freqTable()	58	gcd(), massimo comun divisore	61
frequency()	58	geomCdf()	61
Func, funzione	60	geomPdf()	62
Func, funzione programma	60	Get	62, 216
funzione esponenziale e	60	getDenom(), ottieni/restituisci	
modello di	2	denominatore	63
funzione esponenziale, e^()	45	getKey()	63
funzione piecewise a 2 tratti	58	getLangInfo(), ottieni/restituisci	
modello di	2	informazioni sulla lingua	67
funzione piecewise a N tratti	60	getLockInfo(), testa lo stato	
modello di	2	bloccato di una variabile o	
funzioni		di un gruppo di variabili	67
definite dallutente	37	getMode(), impostazioni di modo	67
funzione programma, Func	60	getNum(), ottieni/restituisci	
parte frazionaria, fpart()	57	numeratore	69
funzioni definite dallutente	37	GetStr	69
funzioni di distribuzione		getType(), get type of variable	69
binomCdf()	77	getVarInfo(), ottieni/restituisci	
binomCdf() (Funzione della		informazioni variabile	70
probabilità cumulativa		giorni tra le date, dbd()	36
per la distribuzione		Goto, vai a	71
binomiale)	20	gruppi, blocco e sblocco	90, 173
binomPdf()	20	gruppi, test stato di blocco	67
invNorm()	77		
invt()	78		
Invχ ² ()	76		

identity(), matrice identità	71	libreria	81
If, se	72	crea collegamenti a oggetti	81
ifFn()	73	libShortcut(), crea collegamenti a	
imag(), parte non reale	74	oggetti libreria	81
implicazione logica, \Rightarrow	193, 224	linea di regressione mediana-	
imposta		mediana, MedMed	97
modo, setMode()	144	lingua	
impostazioni di modo, getMode() ..	67	ottiene informazioni sulla lingua	67
impostazioni, visualizza correnti ..	67	LinRegBx, regressione lineare	82
in mezzo alla stringa, mid()	98	LinRegMx, regressione lineare	83
indice interno di rendimento		LinRegIntervals, regressione lineare	84
modificato, mirr()	99	LinRegtTest (t Test regressione	
inString(), all'interno della stringa ..	74	lineare)	85
int(), intero	75	linSolve()	87
intDiv(), divisione intera	75	listMat(), da lista a matrice	88
integrale definito		lista, conteggio condizionato	31
modello di	6	lista, conteggio elementi	31
intero, int()	75	liste	
interpolate(), interpolare	75	affianca/concatena, augment()	15
inversa, \wedge^{-1}	202	da lista a matrice, listMat() ...	88
inverti righe matrice, rowSwap() ...	140	da matrice a lista, matList() ...	95
invF()	76	differenza, @list()	87
invNorm(), distribuzione normale		differenze in una lista, @list() ..	87
cumulativa inversa	77	elementi vuoti in	222
invt()	78	in mezzo alla stringa, mid()	98
InvX ² ()	76	massimo, max()	95
iPart(), parte intera	78	minimo, min()	99
iperbolico		nuova, newList()	106
arcocoseno, cosh ⁻¹ ()	29	ordinamento ascendente, SortA	152
arcoseno, sinh ⁻¹ ()	150	ordinamento discendente, SortD	153
arcotangente, tanh ⁻¹ ()	162	prodotto scalare, dotP()	44
coseno, cosh()	29	prodotto vettoriale, crossP() ...	32
seno, sinh()	150	prodotto, product()	120
tangente, tanh()	162	somma cumulativa,	
indice di rendimento interno	78	cumulativeSum()	35
indice di rendimento interno, irr		somma, sum()	158-159
()	78	In(), logaritmo naturale	88
isPrime(), verifica numero primo ...	79	LnReg, regressione logaritmica	89
isVoid(), test per nullità	79	Local, variabile locale	90
		locale, Local	90
		Lock, blocca variabile o gruppo di	
		variabili	90
		Log	
		modello di	2
Lbl, etichetta	80	logaritmi	88
lcm, minimo comune multiplo	80	logaritmo naturale, In()	88
left(), sinistra	80	Logistic, regressione logistica	91
LibPriv	38	LogisticD, regressione logistica	92
LibPub	39	loop, Loop	94

Lbl, etichetta	80	modello di	2
lcm, minimo comune multiplo	80	logaritmi	88
left(), sinistra	80	logaritmo naturale, In()	88
LibPriv	38	Logistic, regressione logistica	91
LibPub	39	LogisticD, regressione logistica	92
		loop, Loop	94

Loop, loop	94	scomposizione inferiore-superiore, LU	94
LU, scomposizione matrice inferiore-superiore	94	scomposizione QR, QR	122
lunghezza della stringa	41	somma cumulativa, cumulativeSum()	35
M			
maggiore di o uguale a, 	192	somma, sum()	158-159
maggiore di, >	192	sottomatrice, subMat()	158, 160
massimo comune divisore, gcd()	61	trasposizione, T	160
massimo, max()	95	matrici casuali, randMat()	127
matr•list(), da matrice a lista	95	max(), massimo	95
matrice (1 × 2)	4	mean(), media	96
modello di	4	media, mean()	96
matrice (2 × 1)	4	median(), mediana	96
modello di	4	mediana, median()	96
matrice (2 × 2)	4	MedMed, linea di regressione	97
modello di	4	mediana-mediana	97
matrice (m × n)	4	memorizzazione	203-204
modello di	4	simbolo, &	203-204
matrice casuale, randMat()	127	mentre, While	177
matrice di correlazione, corrMat()	27	mid(), in mezzo alla stringa	98
matrice identità, identity()	71	min(), minimo	99
matrici	15	minimo comune multiplo, lcm	80
affianca/concatena, augment()	46	minimo, min()	99
autovalore, eigVl()	46	minore di o uguale a, {	191
autovettore, eigVc()	88	mirr(), indice interno di rendimento	99
da lista a matrice, list•mat()	95	modificato	99
da matrice a lista, mat•list()	40	mod(), modulo	100
determinante, det()	41	modelli	5
diagonale, diag()	25	derivata prima	5
dimensione colonna, colDim()	41	derivata seconda	5
dimensione, dim()	41	esponente	1
forma a scalini per righe matrici, ref()	130	frazione	1
massimo, max()	101	funzione esponenziale e	2
minimo, min()	25	funzione piecewise a 2 tratti	2
moltiplicazione e somma di righe, mRowAdd()	106	funzione piecewise a N tratti	2
norma colonna, colNorm()	101	integrale definito	6
nuova, newMat()	120	Log	2
operazione con righe, mRow()	188	matrice (1 × 2)	4
prodotto, product()	188	matrice (2 × 1)	4
punto addizione, .+	189	matrice (2 × 2)	4
punto divisione, .P	188	matrice (m × n)	4
punto elevato a potenza, ^	189	prodotto (P)	5
punto moltiplicazione, .*	188	radice ennesima	1
punto sottrazione, .N	188	radice quadrata	1
riempimento, Fill	54	sistema di equazioni (2 equazioni)	3
		sistema di equazioni (N equazioni)	3
		sommatoria (G)	5

valore assoluto	3-4	numerica	
modi		derivata, nDeriv()	106
impostazione, setMode()	144	derivata, nDerivative()	105
modulo, mod()	100	integrale, nInt()	107
moltiplicazione, *	186	soluzione, nSolve()	111
mRow(), operazione con righe di		nuova	
matrice	101	lista, newList()	106
mRowAdd(), moltiplicazione e		matrice, newMat()	106
somma di righe di matrice ..	101		
MultReg (Regressione lineare		O	
multipla)	101		
MultRegIntervals () (Intervalli di		oggetti	
confidenza della previsione		crea collegamenti a libreria	81
di regressione multipla)	101	OneVar, statistiche a una variabile ..	112
MultRegTests () (Verifica sulla		operatore conversione indiretta (#)	227
regressione lineare		operatore di vincolo " "	202
multipla)	102	operatore di vincolo. ordine di	
		valutazione	226
N		operatori	
nand, operatore booleano	104	ordine di calcolo	226
nCr(), combinazioni	105	operatori boolean	
nDerivative(), derivata numerica ...	105	and	8
newList(), nuova lista	106	nor	108
newMat(), nuova matrice	106	operatori booleani	
nfMax(), massimo di una funzione		\Rightarrow	193
calcolato numericamente ..		not	109
nfMin(), minimo di una funzione	106	or	113
calcolato numericamente ..		xor	177
nInt(), integrale numerico	106	Operatori booleani	
nom(), converti tasso effettivo in		\Rightarrow	224
nominaile	107	\Leftrightarrow	193
nor, operatore boolean	108	nand	104
norm(), norma di Frobenius	108	or (Boolean), oppure	113
norma di Frobenius, norm()	109	or, operatore booleano	113
norma riga matrice, rowNorm()	109	ord(), codice numerico di carattere	115
normale casuale, randNorm()	139	ordinamento	
normCdf()	127	ascendente, SortA	152
normPdf()	109	discendente, SortD	153
not, operatore booleano	109	ottieni/restituisci	
notazione in gradi, -	109	denominatore, getDenom() ...	63
notazione in gradi/primi/secondi	200	informazioni variabile,	
notazione in gradianti, G	200	getVarInfo()	67, 70
notazione in primi,	200	numeratore, getNum()	69
notazione in secondi, "	200		
nPr(), disposizioni semplici	110	P	
npv(), valore presente netto	111	P \Rightarrow Rx(), coordinata x rettangolare ..	115
nSolve(), soluzione numerica	111	P \Rightarrow Ry(), coordinata y rettangolare ..	115
nullità, test per	79	parte intera, iPart()	78
		parte non reale, imag()	74

passa errore, PassErr	116	sottrazione, .N	188
PassErr, passa errore	116	su potenza, .^	189
Pdf()	57		
per, For	56		
percentuale, %	189		
piecewise()	116	Q	
poissCdf()	117	QR, scomposizione QR	122
poissPdf()	117	QuadReg, regressione quadratica	123
polare		quando, when()	176
visualizza vettore, ►Polar		QuartReg, regressione quartica	124
polinomi	117		
calcola, polyEval()	118	R	
polinomi casuali, randPoly()	128	R, radianti	199
polinomio casuale, randPoly()	128	R►Pr(), coordinata polare	125
polyEval(), calcola polinomio	118	R►θ(), coordinata polare	125
PolyRoots()	118	radianti, R	199
potenza di dieci, 10^()	202	radice ennesima	
PowerReg, regressione su potenza	119	modello di	1
Prgm, definisci programma	120	radice quadrata	
probabilità di distribuzione normale, normCdf()	109	modello di	1
probabilità di distribuzione t di Student, tCdf()	163	radice quadrata, √()	154, 194
prodotto (P)	5	rand(), numero casuale	126
modello di	32	randBin, numero casuale	126
prodotto vettoriale, crossP()	120	randInt(), intero casuale	126
prodotto, product()	194	randMat(), matrice casuale	127
prodotto, Π()	120	randNorm(), normale casuale	127
prodSeq()	120	randPoly(), polinomio casuale	128
product(), prodotto	120	randSamp()	128
programmazione	120	RandSeed, seme numero casuale	128
definisci programma, Prgm	120	real(), reale	129
display data, Disp	42	reale, real()	129
passa errore, PassErr	116	reciproco, ^-1	202
visualizza dati, Vis	141	ref(), forma a scalini per righe	130
programmi	38	RefreshProbeVars	131
definizione di libreria privata	39	regressione cubica, CubicReg	34
definizione di libreria pubblica	39	regressione esponenziale, ExpReg	51
programmi e programmazione	38	regressione lineare, LinRegAx	83
cancella errore, ClrErr	39	regressione lineare, LinRegBx	82, 84
fine tentativo, EndTry	24	regressione logaritmica, LnReg	89
tentativo, Try	166	regressione logistica, Logistic	91
visualizza schermata I/O, Vis	166	regressione logistica, LogisticD	92
visualizza schermo I/O, Disp	141	regressione quadratica, QuadReg	123
propFrac, frazione propria	42	Regressione quartica (QuartReg)	124
punto	121	regressione sinusoidale, SinReg	151
addizione, .+	188	regressione su	
divisione, .P	188	potenza,	
moltiplicazione, .*	188	PowerReg ... 118-119, 132, 134, 163	
regressioni		cubica, CubicReg	34
esponenziale, ExpReg		esponenziale, ExpReg	51

linea mediana-mediana,		variabili	224
MedMed	97	scelte rapide della tastiera	224
logaritmica, LnReg	89	scelte rapide, tastiera	224
Logistic (Regressione logistica) ..	91	scomposizione QR, QR	122
logistica, Logistic	92	se, If	72
MultReg (Regressione lineare		sec ⁻¹ (), secante inversa	141
multipla)	101	sec(), secante	140
quadratica, QuadReg	123	sech ⁻¹ (), secante iperbolica inversa	141
quartica, QuartReg	124	sech(), secante iperbolica	141
regressione lineare, LinRegAx ..	83	segno negativo, introduzione di	
regressione lineare, LinRegBx ..	82, 84	numeri negativi	227
regressione su potenza,		segno, sign()	147
PowerReg	119	seme numero casuale, RandSeed	128
sinusoidale, SinReg	151	seno, sin()	149
regressioni su potenza, PowerReg ..	132, 134	seq(), sequenza	142
regressions, regressions		seqGen()	142
regressione su potenza,		seqn()	143
PowerReg	118, 163	sequence, seq()	142-143
remain(), resto	132	sequenza, seq()	142
Request	132	setMode(), imposta modo	144
RequestStr	134	shift(), sposta	146
restituisci, Return	135	sign(), segno	147
resto, remain()	132	simult(), equazioni simultanee	148
Return, restituisci	135	sin ⁻¹ (), arcoseno	149
riempi	214-215	sin(), seno	149
right(), destra	135	sinh ⁻¹ (), arcoseno iperbolico	150
right, right()	48, 175	sinh(), seno iperbolico	150
rimuovi		sinistra, left()	80
elementi vuoti dalla lista	40	SinReg, regressione sinusoidale	151
ripeti, Cycle	35	sistema di equazioni (2 equazioni)	
risultati a due variabili, TwoVar	171	modello di	3
risultati, statistica	155	sistema di equazioni (N equazioni)	
risultato (ultimo), Ans	12	modello di	3
rk23(), funzione Runge-Kutta	135	somma cumulativa, cumulativeSum(
rotate(), ruota	137)	35
round(), arrotondamento	138	somma dell'interesse pagato	196
rowAdd(), somma di righe di		somma di capitale versato	197
matrice	139	somma di righe di matrice, rowAdd(
rowDim(), dimensione righe di)	139
matrice	139	somma, sum()	158
rowNorm(), norma riga matrice	139	somma, $\Sigma()$	195
rowSwap(), inverti righe matrice	140	sommatoria (G)	
rref(), forma a scalini ridotta per		modello di	5
righe	140	SortA, ordinamento ascendente	152
ruota, rotate()	137	SortD, ordinamento discendente	153
S		sostituzione con l'operatore "!"	202
sblocco di variabili e gruppi di	173	sottomatrice, subMat()	158, 160
		sottrazione, -	185
		sposta, shift()	146

sqrt(), radice quadrata	154	subMat(), sottomatrice	158, 160
stat.results	155	sum(), somma	158
stat.values	156	sumIf()	159
statistica		sumSeq()	160
combinazioni, nCr()	105		
deviazione standard, stdDev			T
()	156-157, 173	T, trasposizione	160
disposizioni semplici, nPr()	110	tabella di ammortamento, amortTbl(
fattoriale, !	194	()	7, 16
media, mean()	96	tan ⁻¹ (), arcotangente	161
mediana, median()	96	tan(), tangente	160
normale casuale, randNorm()	127	tangente, tan()	160
risultati a due variabili, TwoVar	171	tanh ⁻¹ (), arcotangente iperbolica	162
statistiche a una variabile,		tanh(), tangente iperbolica	162
OneVar	112	tasso di variazione media, avgRC()	15
varianza, variance()	174	tasso effettivo, eff()	45
statistica seme numero casuale,		tasso nominale, nom()	108
RandSeed	128	tCdf(), probabilità di distribuzione t	
statistiche a una variabile, OneVar	112	di Student	163
stdDevPop(), deviazione standard		test per nullità, isVoid()	79
della popolazione	156	Test_2S, verifica F su due campioni	59
stdDevSamp(), deviazione standard		Text, comando	163
del campione	157	tInterval, intervallo di confidenza t	164
string(), da espressione a stringa	158	tInterval_2Samp, intervallo di	
stringa		confidenza t su due	
dimensione, dim()	41	campioni	165
lunghezza	41	tPdf(), densità di probabilità t di	
stringa di caratteri, char()	21	Student	165
stringa formato, format()	56	trace() (Traccia)	166
stringhe		trasposizione, T	160
aggiunge, &	194	trattino basso, _	201
codice di carattere, ord()	115	Try, comando gestione errore	166
conversione indiretta, #	198	tTest, verifica t	167
da espressione a stringa, string(tTest_2Samp, verifica t su due	
)	158	campioni	168
da stringa a espressione, expr()	51	tvmFV()	169
destra, right()	75, 135	tvmI()	169
dimensione, dim()	41	tvmN()	170
formato, format()	56	tvmPmt()	170
formattazione	56	tvmPV()	170
in mezzo alla stringa, mid()	98	TwoVar, risultati a due variabili	171
ruota, rotate()	137		
sinistra, left()	80		U
sposta, shift()	146		
stringa di caratteri, char()	21	uguale, =	189
utilizzo per creare nomi di		unitV(), vettore unità	173
variabile	227	unLock, sblocca variabile o gruppo di	
strings		variabili	173
right, right()	48, 175		

V			
vai a, Goto	71	numero decimale, »Base10	18
valore assoluto		vettore in forma cilindrica, 4Cylind	35
modello di	3-4	vettore polare, »Polar	117
valore del denaro rapportato al tempo, importo della rata ..	170	vettore sferico, »Sphere	153
valore del denaro rapportato al tempo, interesse	169	visualizza come vettore	
valore del denaro rapportato al tempo, numero di rate	170	rettangolare, »Rect	129
valore del denaro rapportato al tempo, valore futuro	169	visualizza dati, Disp	42
valore del denaro rapportato al tempo, valore presente	170	visualizza dati, Vis	141
valore presente netto, npv()	111	visualizza gradi/primi/secondi, »DMS	44
valori dei risultati, statistica	156	visualizza vettore in forma cilindrica, »Cylind	35
variabile		visualizza vettore sferico, »Sphere ..	153
creazione di un nome da una stringa di caratteri	227		
variabile locale, Local	90	W	
variabili			
cancella tutte con il nome di un solo carattere	24	x², quadrato	187
elimina, DelVar	39	XNOR	193
locale, Local	90	xor, or esclusivo booleano	177
variabili e funzioni		X	
copia	26		
variabili, blocco e sblocco	67, 90, 173	zInterval , intervallo di confidenza z ..	178
varianza, variance()	174	zInterval_1Prop , intervallo di confidenza z per una proporzione	179
varPop()	173	zInterval_2Prop , intervallo di confidenza z per due proporzioni	179
varSamp() , varianza campione	174	zInterval_2Samp , intervallo di confidenza z su due campioni	180
verifica numero primo, isPrime()	79	zTest	181
Verifica t sulla regressione lineare multipla	102	zTest_1Prop , verifica z per una proporzione	182
verifica t, tTest	167	zTest_2Prop , verifica z per due proporzioni	183
vettore unità, unitV()	173	zTest_2Samp , verifica z su due campioni	183
vettori			
prodotto scalare, dotP()	44	Δ	
prodotto vettoriale, crossP()	32		
unità, unitV()	173	Δlist() , differenza in una lista	87
visualizza vettore in forma cilindrica, »Cylind	35		
Vis, visualizza dati	141		
visualizza come			
angolo decimale, »DD	36		
binario, »Base2	17		
esadecimale, »Base16	19		
gradi/primi/secondi, »DMS	44		

X

χ^2 2way	22
χ^2 Cdf()	22
χ^2 GOF	23
χ^2 Pdf()	24