



TI-*Nspire*TM

TI-Nspire™
Guida di riferimento

La presente Guida è relativa alla versione 4.3 del software TI-Nspire™. Per ottenere la versione più aggiornata della documentazione, visitare il sito education.ti.com/guides.

Informazioni importanti

Salvo per quanto dichiarato espressamente nella licenza che accompagna un programma, Texas Instruments non rilascia alcuna garanzia, esplicita o implicita, incluse fra le altre le garanzie implicite di commerciabilità e di idoneità ad uno scopo particolare, per quanto riguarda programmi o materiali cartacei e rende disponibili tali materiali esclusivamente su base "tal quale." In nessun caso Texas Instruments è responsabile verso chicchessia di danni speciali, collaterali, incidentali o consequenziali in relazione con o derivanti dall'acquisto o dall'uso di questi materiali, e la sola ed esclusiva responsabilità di Texas Instruments, indipendentemente dalla forma dell'azione, non supera l'importo indicato nella licenza per il programma. Inoltre Texas Instruments non è responsabile di rivendicazioni di alcun genere contro l'uso di questi materiali da parte di chiunque.

Licenza

Leggere per intero la licenza installata in **C:\Program Files\TI Education\<TI-Nspire™ Product Name>\license**.

© 2006 - 2016 Texas Instruments Incorporated

Sommario

Informazioni importanti	2
Sommario	3
Modelli di espressione	5
Elenco alfabetico	12
A	12
B	21
C	26
D	53
E	65
F	76
G	86
I	93
L	101
M	118
N	127
O	136
P	139
Q	149
R	152
S	166
T	195
U	212
V	212
W	214
X	216
Z	217
Simboli	226
Elementi vuoti (nulli)	254
Scelte rapide per l'inserimento di espressioni matematiche	256
EOS™ (Equation Operating System) gerarchia	258
Codici di errore e messaggi	260
Codici di avvertenza e messaggi	269
Informazioni generali	271
Informazioni sul servizio di manutenzione e riparazione del prodotto TI e sulla	271

garanzia	
Indice	273

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** **÷**



Esempio:

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

Nota: vedere anche / (divisione), pagina 228.

Modello di esponente

Tasto **^**



Esempio:

$$2^3 \quad 8$$

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 229.

Modello di radice quadrata

Tasti **ctrl** **x²**



Esempio:

$$\sqrt{\{9, a, 4\}} \quad \{3, \sqrt{a}, 2\} \quad 2$$

Nota: vedere anche $\sqrt()$ (radice quadrata), pagina 240.

Modello di radice ennesima

Tasti **ctrl** **^**



Esempio:



Nota: vedere anche **root()**, pagina 163.

Modello di radice ennesima

Tasti

$$\begin{array}{r} \sqrt[3]{8} \\ \hline \sqrt[3]{\{8, 27, b\}} \\ \hline \left\{ \begin{array}{l} \frac{1}{2, 3, b^3} \end{array} \right\} \end{array}$$

Modello di funzione esponenziale e

Tasti

e

Esponenziale naturale e elevato a potenza

Nota: vedere anche **e¹⁽⁾**, pagina 65.

Esempio:

$$e^1$$

$$e^{1.}$$

2.71828182846

Modello di log

Tasto

log

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

Nota: vedere anche **log()**, pagina 114.

Esempio:

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

0.5

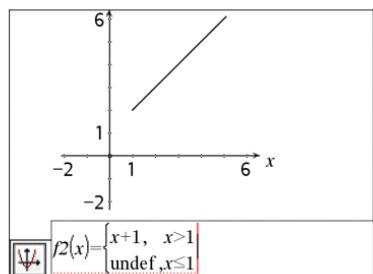
Modello di funzione piecewise a 2 tratti

Catalogo >

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 **piecewise()**, pagina 141.

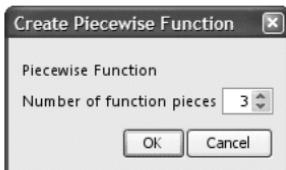
Esempio:



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.

Nota: vedere anche **piecewise()**, pagina 141.

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 194.

Esempio:

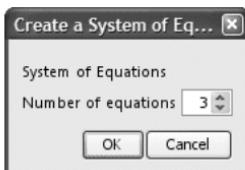
$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y\right) \quad x=\frac{5}{2} \text{ and } y=-\frac{5}{2}$$

$$\text{solve}\left(\begin{cases} y=x^2-2 \\ x+2 \cdot y=1 \end{cases}, x, y\right) \quad x=\frac{-3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1$$

Modello di sistema di N equazioni

Catalogo > 

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



Esempio:

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

Nota: vedere anche **system()**, pagina 194.

Modello di valore assoluto

Catalogo > 



Nota: vedere anche **abs()**, pagina 12.

Esempio:

Modello di valore assoluto**Catalogo >**

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

Modello di gg°pp'ss.ss"**Catalogo >**

[0°00'00]"

Esempio:

$$\begin{array}{r} 30^\circ 15' 10'' \\ \hline 10891 \cdot \pi \\ 64800 \end{array}$$

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:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot a \quad \begin{bmatrix} a & 2 \cdot a \\ 3 \cdot a & 4 \cdot a \end{bmatrix}$$

Crea una matrice 2 x 2.

Modello di matrice (1 x 2)**Catalogo >**

$$[\square \ \square]$$

Esempio:

$$\text{crossP}([1 \ 2], [3 \ 4]) \quad [0 \ 0 \ -2]$$

Modello di matrice (2 x 1)**Catalogo >**

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

Esempio:

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

Modello di matrice (m x n)**Catalogo >**

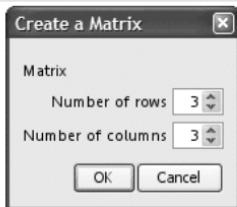
Il modello appare dopo la richiesta di specificare il numero di righe e colonne.

Esempio:

$$\text{diag} \begin{pmatrix} 4 & 2 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{pmatrix} \quad [4 \ 2 \ 9]$$

Modello di matrice (m x 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 > 

$$\sum_{\square = \square}^{\square} (\square)$$

Esempio:

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

25

Nota: vedere anche $\Sigma()$ (sumSeq), pagina 241.

Modello di prodotto (Π)

Catalogo > 

$$\prod_{\square = \square}^{\square} (\square)$$

Esempio:

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

$\frac{1}{120}$

Nota: vedere anche $\Pi()$ (prodSeq), pagina 240.

Modello di derivata prima

Catalogo > 

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

Esempio:

$$\frac{d}{dx}(x^3)$$

$3 \cdot x^2$

Il modello di derivata prima può essere utilizzato anche per calcolare la derivata prima in un punto.

$$\frac{d}{dx}(x^3)|_{x=3}$$

27

Modello di derivata prima

Catalogo > 

Nota: vedere anche **d()** (**derivata**), pagina 237.

Modello di derivata seconda

Catalogo > 

$$\frac{d^2}{dx^2}(\square)$$

Il modello di derivata seconda può essere utilizzato anche per calcolare la derivata seconda in un punto.

Nota: vedere anche **d()** (**derivata**), pagina 237.

Esempio:

$$\frac{d^2}{dx^2}(x^3) \quad 6 \cdot x$$

$$\frac{d^2}{dx^2}(x^3)|_{x=3} \quad 18$$

Modello di derivata ennesima

Catalogo > 

$$\frac{d^{\square}}{dx^{\square}}(\square)$$

Il modello di derivata *n*-esima può essere utilizzato per calcolare la derivata *n*-esima.

Nota: vedere anche **d()** (**derivata**), pagina 237.

Esempio:

$$\frac{d^3}{dx^3}(x^3)|_{x=3} \quad 6$$

Modello di integrale definito

Catalogo > 

$$\int_a^b \square dx$$

Nota: vedere anche **j()** (**integral()**), pagina 226.

Esempio:

$$\int_a^b x^2 dx \quad \frac{b^3}{3} - \frac{a^3}{3}$$

modello di integrale indefinito

Catalogo > 

$$\int \square dx$$

Nota: vedere anche **j()** (**integral()**), pagina 226.

Esempio:

$$\int x^2 dx \quad \frac{x^3}{3}$$

$$\lim_{\substack{[x] \rightarrow [y]}} ([x])$$

Esempio:

$$\lim_{x \rightarrow 5} (2 \cdot x + 3)$$

13

Utilizzare - o (-) per il limite da sinistra.
Utilizzare + per il limite da destra.

Nota: vedere anche **limit()**, pagina 103.

Elenco alfabetico

Gli elementi i cui nomi sono composti da caratteri non alfabetici (come ad esempio +, !, >) sono elencati alla fine della presente sezione, pagina 226. 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 >

abs(*EsprI*)⇒*espressione*

$$\left| \left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\} \right| = \left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\}$$

abs(*ListaI*)⇒*lista*

$$|2-3 \cdot i| = \sqrt{13}$$

abs(*MatriceI*)⇒*matrice*

$$|z| = |z|$$

Restituisce il valore assoluto dell'argomento.

$$|x+y \cdot i| = \sqrt{x^2+y^2}$$

Nota: vedere anche **Modello di valore assoluto**, pagina 7.

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 >

amortTbl(*NPmt,N,I,PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], valoreArrotondato*)⇒*matrice*

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

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 209.

- 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 $\Sigma\text{Int}()$ e $\Sigma\text{Prn}()$, pagina 241, e $\text{bal}()$, pagina 21.

and

Espressione booleana1 **and** Espressione booleana2 \Rightarrow Espressione booleana

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

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 and 0h3D5F	0h2C16
--------------------	--------

Importante: è zero, non la lettera O.

In modalità base Bin:

0b100101 and 0b100	0b100
--------------------	-------

In modalità base Dec:

and

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 > **angle(*Espr1*)**⇒*espressione*

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

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

In modalità angolo in gradi:

angle(0+2·i)

90

In modalità angolo in gradi (gradi centesimali):

angle(0+3·i)

100

In modalità angolo in radianti:

angle(1+i)

 $\frac{\pi}{4}$

angle(z)

 $-\pi \cdot (\text{sign}(z) - 1)$

angle(x+i·y)

 $\frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right)$ **angle(*Listal*)**⇒*lista***angle(*Matrice1*)**⇒*matrice*

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

angle({1+2·i, 3+0·i, 0-4·i})

 $\left\{ \frac{\pi}{2} - \tan^{-1}\left(\frac{1}{2}\right), 0, -\frac{\pi}{2} \right\}$

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 189).

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 189).

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.dflInteract	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

Catalogo >

approx(*Espr1*)⇒*espressione*

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(*Listal*)⇒*lista*

approx(*Matrice1*)⇒*matrice*

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

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

approx({sin(π), cos(π)})	{0., -1.}
approx([sqrt(2) sqrt(3)])	[1.41421 1.73205]

►approxFraction()

Catalogo >

Espr ►**approxFraction**

([*Tol*])⇒*espressione*

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()

Catalogo >

approxRational(*Espr*, *tol*)⇒*espressione*

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 38.****arccosh()****Vedere $\cosh^{-1}()$, pagina 39.****arccot()****Vedere $\cot^{-1}()$, pagina 40.****arccoth()****Vedere $\coth^{-1}()$, pagina 41.****arccsc()****Vedere $\csc^{-1}()$, pagina 44.****arccsch()****Vedere $\csch^{-1}()$, pagina 45.****arcLen() (Lunghezza arco)****Catalogo >** **arcLen(*Espr1,Var,Inizio,Fine*)**
⇒*espressione*Restituisce la lunghezza dell'arco di *Espr1* da *Inizio* a *Fine* in funzione della variabile *Var*.

$$\frac{\text{arcLen}(\cos(x),x,0,\pi)}{\text{arcLen}(f(x),x,a,b)} \quad 3.8202$$
$$\int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} dx$$

La lunghezza dell'arco viene calcolata come un integrale, basandosi sulla definizione di una funzione.

arcLen(*Lista1,Var,Inizio,Fine*)⇒*lista*Restituisce una lista delle lunghezze dell'arco di ciascun elemento di *Lista1* da *Inizio* a *Fine* in funzione di *Var*.

$$\text{arcLen}(\{\sin(x),\cos(x)\},x,0,\pi) \quad \{3.8202, 3.8202\}$$

arcsec()**Vedere $\sec^{-1}()$, pagina 167.**

arcsech()**Vedere $\operatorname{sech}^{-1}()$, pagina 168.****arcsin()****Vedere $\sin^{-1}()$, pagina 179.****arcsinh()****Vedere $\sinh^{-1}()$, pagina 180.****arctan()****Vedere $\tan^{-1}()$, pagina 196.****arctanh()****Vedere $\tanh^{-1}()$, pagina 197.****augment() (Affianca/concatena)****Catalogo > ****augment(Lista1, Lista2)⇒lista**

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

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

augment(Matrice1, Matrice2)⇒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(m1,m2)	$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$

avgRC() (Tasso di variazione media)**Catalogo >**

avgRC(*Espr1*, *Var* [=Valore] [, *Incr*])⇒*espressione*

$$\text{avgRC}(f(x), x, h) \quad \frac{f(x+h) - f(x)}{h}$$

avgRC(*Espr1*, *Var* [=Valore] [, *Listal1*])⇒*lista*

$$\text{avgRC}(\sin(x), x, h)|_{x=2} \quad \frac{\sin(h+2) - \sin(2)}{h}$$

avgRC(*Listal1*, *Var* [=Valore] [, *Incr*])⇒*lista*

$$\text{avgRC}(x^2 - x + 2, x) \quad 2 \cdot (x - 0.4995)$$

avgRC(*Matrice1*, *Var* [=Valore] [, *Incr*])⇒*matrice*

$$\text{avgRC}(x^2 - x + 2, x, 0.1) \quad 2 \cdot (x - 0.45)$$

$$\text{avgRC}(x^2 - x + 2, x, 3) \quad 2 \cdot (x + 1)$$

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()****Catalogo >**

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

$$\text{bal}(5, 6, 5.75, 5000, , 12, 12) \quad 833.11$$

bal(*NPmt*, *tabellaAmmortamento*)⇒*valore*

$$\text{tbl} := \text{amortTbl}(6, 6, 5.75, 5000, , 12, 12)$$

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

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

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* e *PmtAt* sono descritti nella tabella degli argomenti TVM, pagina 209.

$$\text{bal}(4, \text{tbl}) \quad 1674.27$$

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 209.

- 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(*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 12.

Nota: vedere anche **ΣInt()** e **ΣPrn()**, pagina 241.

►Base2

Interol ►Base2⇒*intero*

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

Converte *Interol* 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*

256►Base2	0b100000000
0h1F►Base2	0b11111

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 binaria, indipendentemente dalla modalità Base impostata.

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

-1 è visualizzato come

0hFFFFFFFFFFFFFFF in modalità base Esadecimale 0b111...111 (la cifra 1 ripetuta 64 volte) in modalità base Binaria

- 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

►Base10

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 esadecimali devono sempre avere, 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 8 cifre.

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

0b10011►Base10	19
0h1F►Base10	31

►Base16

Interol ►Base16⇒*intero*

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

256►Base16	0h100
0b111100001111►Base16	0hF0F

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, *InteroI* 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 22.

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

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 $P(0 \leq X \leq valoreSuperiore)$ ⇒*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 le probabilità di successo *p* per ciascuna prova.

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

binomPdf()

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)**

ceiling(*EsprI*)⇒*intero*

ceiling(.456)

1.

Restituisce il più vicino numero intero \leq all'argomento.

L'argomento può essere un numero reale o complesso.

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

ceiling(*Listal*)⇒*lista*

ceiling({{-3.1, 1, 2.5}}) { -3., 1, 3. }

ceiling(*MatriceI*)⇒*matrice*

ceiling([[0, -3.2·i], [1.3, 4]]) [[0, -3·i], [2., 4]]

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

centralDiff()

centralDiff(*EsprI, Var [=Valore]*, *[,Incr]*)⇒*espressione*

centralDiff(cos(x), x, h)

$$\frac{-(\cos(x-h)-\cos(x+h))}{2 \cdot h}$$

centralDiff(*EsprI, Var [=Valore]*, *[,Incr]*) | *Var=Valore*⇒*espressione*

$$\lim_{h \rightarrow 0} \frac{\text{centralDiff}(\cos(x), x, h)}{-\sin(x)}$$

centralDiff(*EsprI, Var [=Valore]*, *[,Lista]*)⇒*lista*

centralDiff(x^3, x, 0.01)

$$3 \cdot (x^2 + 0.000033)$$

centralDiff(*Listal, Var [=Valore]*, *[,Incr]*)⇒*lista*

centralDiff(cos(x), x)|x=π/2

$$-1.$$

centralDiff(*MatriceI, Var [=Valore]*, *[,Incr]*)⇒*matrice*

centralDiff(x^2, x, {0.01, 0.1})

$$\{2 \cdot x, 2 \cdot x\}$$

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 *Listal* o *MatriceI*, l'operazione viene mappata sui valori della lista o sugli elementi della matrice.

Nota: vedere anche `e d()`.

cFactor() (Fattore complesso)

cFactor(*EsprI*[,*Var*])⇒*espressione*

cFactor(*Listal*[,*Var*])⇒*lista*

cFactor(*MatriceI*[,*Var*])⇒*matrice*

cFactor(*EsprI*) restituisce la scomposizione in fattori di *EsprI* rispetto a tutte le variabili con un denominatore comune.

EsprI viene scomposto, per quanto possibile, in fattori razionali lineari, anche se ciò introduce nuovi numeri non reali. Questa procedura è utile qualora si desideri ottenere una scomposizione in fattori relativamente a più di una variabile.

cFactor(*EsprI*,*Var*) restituisce *EsprI* scomposto in fattori relativamente alla variabile *Var*.

EsprI viene scomposto, per quanto possibile, in fattori lineari in *Var*, pur con costanti non reali, anche se vengono introdotte costanti irrazionali o sottoespressioni che sono irrazionali in altre variabili.

cFactor($a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x$)	$a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i)$
cFactor($x^2 + \frac{4}{9}$)	$\frac{(3 \cdot x - 2 \cdot i) \cdot (3 \cdot x + 2 \cdot i)}{9}$
cFactor($x^2 + 3$)	$x^2 + 3$
cFactor($x^2 + a$)	$x^2 + a$

cFactor($a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x$)	$a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i)$
cFactor($x^2 + 3, x$)	$(x + \sqrt{3} \cdot i) \cdot (x - \sqrt{3} \cdot i)$
cFactor($x^2 + a, x$)	$(x + \sqrt{a} \cdot -i) \cdot (x + \sqrt{a} \cdot i)$

cFactor() (Fattore complesso)

Catalogo >

I fattori ed i rispettivi termini vengono ordinati con *Var* come variabile principale. Le potenze simili di *Var* sono ridotte in ciascun fattore. Includere *Var* se si desidera che la scomposizione in fattori tenga conto solo di tale variabile e che le espressioni irrazionali siano incluse in qualsiasi altra variabile per aumentare la scomposizione in fattori relativamente a *Var*. Si può verificare una scomposizione in fattori incidentale relativamente ad altre variabili.

Nell'impostazione Auto della modalità **Auto/Approssimato**, l'inclusione di *Var* permette inoltre l'approssimazione con coefficienti a virgola mobile nel caso in cui i coefficienti irrazionali non possano essere esplicitamente espressi in termini concisi con le funzioni incorporate. Anche qualora vi sia una sola variabile, se si include *Var* la scomposizione in fattori può risultare più completa.

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

char() (Stringa di caratteri)

Catalogo >

char(*Intero***)**⇒carattere

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.

cFactor($x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3$)
$x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3$
cFactor($x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3, x$)
$(x-0.964673) \cdot (x+0.611649) \cdot (x+2.12543) \cdot (x+4.18816)$

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

charPoly() (Polinomio caratteristico)

Catalogo >

charPoly
(*matriceQuadrata*,*Var*)⇒espressione
polinomiale

$m := \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$
--	---

charPoly(<i>m</i> , <i>x</i>)	$-x^3+5 \cdot x^2+7 \cdot x-35$
charPoly(<i>m</i> , <i>x</i> ² +1)	$-x^6+2 \cdot x^4+14 \cdot x^2-24$
charPoly(<i>m</i> , <i>m</i>)	0

charPoly
(*matriceQuadrata*,*Espr*)⇒espressione
polinomiale

charPoly
(*matriceQuadrata1*,*Matrice2*)⇒espressione polinomiale

charPoly() (Polinomio caratteristico)**Catalogo > **

Restituisce il polinomio caratteristico di *matriceQuadrata*. Il polinomio caratteristico di $n \times n$ matrice A , indicato da $p_A(\lambda)$, è il polinomio definito da

$$p_A(\lambda) = \det(\lambda \cdot I - A)$$

dove I indica la matrice identica $n \times n$.

matriceQuadrata1 e *matriceQuadrata2* devono avere le stesse dimensioni.

 $\chi^2\text{way}$ **Catalogo > **

$\chi^2\text{way}$ *MatriceOss*

chi22way *MatriceOss*

Esegue una verifica c^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 189).

Per informazioni sull'effetto di elementi vuoti in una matrice, vedere "Elementi vuoti (nulli)", pagina 254.

Variabile di output	Descrizione
stat. χ^2	Statistica Chi quadrato: somma $(\text{osservati} - \text{attesi})^2 / \text{attesi}$
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

 $\chi^2\text{Cdf()}$ **Catalogo > **

$\chi^2\text{Cdf}$

(

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 254.

 $\chi^2\text{GOF}$ 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 189).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

Variabile di output	Descrizione
<i>stat.χ^2</i>	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

Variabile di output	Descrizione
stat.df	Gradi di libertà per le statistiche chi quadrato
stat.CompList	Contributi statistici chi quadrato elementari

$\chi^2\text{Pdf}()$

Catalogo >

$\chi^2\text{Pdf}(ValX,gl) \Rightarrow$ 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 254.

ClearAZ (Cancella AZ)

Catalogo >

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 212.

5→b	5
b	5
ClearAZ	Done
b	b

ClrErr

Catalogo >

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 205.

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 140 e **Try**, pagina 205.

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().

comDenom() (Denominatore comune)

comDenom(*Espr1[,Var]*)⇒espressione

comDenom(*Listal[,Var]*)⇒lista

comDenom(*Matrice1[,Var]*)⇒matrice

$$\text{comDenom} \left(\frac{y^2+y}{(x+1)^2} + y^2+y \right) \\ \frac{x^2 \cdot y^2 + x^2 \cdot y + 2 \cdot x \cdot y^2 + 2 \cdot x \cdot y + 2 \cdot y^2 + 2 \cdot y}{x^2 + 2 \cdot x + 1}$$

comDenom(*Espr1*) restituisce una frazione ridotta con numeratore e denominatore completamente espansi.

comDenom(*Espr1,Var*) restituisce una frazione ridotta con numeratore e denominatore espansi rispetto a *Var*. I termini ed i rispettivi fattori sono ordinati considerando *Var* la variabile principale. Le potenze simili di *Var* sono ridotte. Si può verificare una scomposizione in fattori incidentale dei coefficienti ridotti. Questo procedimento, rispetto all'omissione di *Var*, permette di risparmiare tempo, memoria e spazio sullo schermo, rendendo inoltre l'espressione più comprensibile. Le successive operazioni eseguite sul risultato sono più veloci e non rischiano di esaurire la memoria.

Se *Var* non compare in *Espr1*, **comDenom(*Espr1,Var*)** restituisce una frazione ridotta con numeratore e denominatore non espansi. Tali risultati permettono di solito di risparmiare ulteriore tempo, memoria e spazio sullo schermo. Le successive operazioni eseguite sul risultato sono più veloci e non rischiano di esaurire la memoria, grazie a tali risultati, parzialmente scomposti in fattori.

$$\text{comDenom} \left(\frac{y^2+y}{(x+1)^2} + y^2+y, x \right) \\ \frac{x^2 \cdot y \cdot (y+1) + 2 \cdot x \cdot y \cdot (y+1) + 2 \cdot y \cdot (y+1)}{x^2 + 2 \cdot x + 1}$$

$$\text{comDenom} \left(\frac{y^2+y}{(x+1)^2} + y^2+y, y \right) \\ \frac{y^2 \cdot (x^2 + 2 \cdot x + 2) + y \cdot (x^2 + 2 \cdot x + 2)}{x^2 + 2 \cdot x + 1}$$

Define *comden(exprn)*=*comDenom(exprn,abc)*
Done

$$\text{comden} \left(\frac{y^2+y}{(x+1)^2} + y^2+y \right) \quad \frac{(x^2 + 2 \cdot x + 2) \cdot y \cdot (y+1)}{(x+1)^2}$$

comDenom() (Denominatore comune)

Catalogo >

Anche qualora non vi sia nessun denominatore, la funzione **comden** è spesso un modo veloce per ottenere una scomposizione in fattori parziale, se **factor()** è troppo lento oppure se si rischia di esaurire la memoria.

Suggerimento: inserire questa definizione della funzione **comden()** e utilizzarla regolarmente come alternativa a **comDenom()** e **factor()**.

$$\text{comden}\left(1234 \cdot x^2 \cdot (y^3 - y) + 2468 \cdot x \cdot (y^2 - 1)\right) \\ 1234 \cdot x \cdot (x \cdot y + 2) \cdot (y^2 - 1)$$

completeSquare ()

Catalogo >

completeSquare(EsprOEq, Var)⇒espressione o equazione

completeSquare(EsprOEq, Var^Potenza)⇒espressione o equazione

completeSquare(EsprOEq, Var1, Var2 [...])⇒espressione o equazione

completeSquare(EsprOEq, {Var1, Var2 [...]})⇒espressione o equazione

Converte un'espressione polinomiale quadratica dalla forma $a \cdot x^2 + b \cdot x + c$ nella forma $a \cdot (x-h)^2 + k$

Oppure

Converte un'equazione quadratica dalla forma $a \cdot x^2 + b \cdot x + c = d$ nella forma $a \cdot (x-h)^2 = k$

Il primo argomento deve essere un'espressione o un'equazione quadratica nella forma standard rispetto al secondo argomento.

Il secondo argomento deve essere un singolo termine a una variabile o un singolo termine a una variabile elevato a una potenza razionale, per esempio x, y^2 oppure $z^{(1/3)}$.

La terza e la quarta sintassi tentano di completare il quadrato rispetto alle variabili $Var1, Var2 [...]$.

$$\text{completeSquare}(x^2 + 2 \cdot x + 3, x) \quad (x+1)^2 + 2$$

$$\text{completeSquare}(x^2 + 2 \cdot x - 3, x) \quad (x+1)^2 - 4$$

$$\text{completeSquare}(x^6 + 2 \cdot x^3 + 3 \cdot x^3) \quad (x^3 + 1)^2 + 2$$

$$\text{completeSquare}(x^2 + 4 \cdot x + y^2 + 6, y + 3 = 0, x, y) \quad (x+2)^2 + (y+3)^2 = 10$$

$$\text{completeSquare}(3 \cdot x^2 + 2 \cdot y + 7, y^2 + 4 \cdot x = 3, \{x, y\}) \\ 3 \cdot \left(x + \frac{2}{3}\right)^2 + 7 \cdot \left(y + \frac{1}{7}\right)^2 = \frac{94}{21}$$

$$\text{completeSquare}(x^2 + 2 \cdot x \cdot y, x, y) \quad (x+y)^2 - y^2$$

conj() (Coniugato)

Catalogo >

conj(*Espr1*)⇒*espressione*

conj(*Listal*)⇒*lista*

conj(*MatriceI*)⇒*matrice*

Restituisce il complesso coniugato dell'argomento.

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

conj(1+2·i)	1-2·i
conj([2 1-3·i;-i -7])	[2 1+3·i;i -7]
conj(z)	z
conj(x+i;y)	x-y·i

constructMat() (Costruisci matrice)

Catalog >

constructMat

(

Espr

,

Var1

,*Var2,numRighe,numColonne*)⇒*matrice*

Restituisce una matrice sulla base degli argomenti.

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

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)

Catalog >

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.

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.

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

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.,bb.</i>	Done
getVarInfo()	
	$\begin{cases} aa.a \text{ "NUM" } "0" \\ aa.b \text{ "NUM" } "0" \\ bb.a \text{ "NUM" } "0" \\ bb.b \text{ "NUM" } "0" \end{cases}$

corrMat() (Matrice di correlazione)

Catalogo >

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

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

►cos (Coseno)

Catalogo >

Espr ►cos

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

Rappresenta *Espr* rispetto al coseno. È un operatore di conversione della visualizzazione. Può essere utilizzato solo alla fine della riga di introduzione.

►cos riduce tutte le potenze di sin(...) modulo 1-cos(...)^2 in modo che qualsiasi potenza rimanente di cos(...) abbia esponenti compresi nell'intervallo (0, 2). Pertanto, il risultato non conterrà sin(...) se e solo se sin(...) si presenta nell'espressione data solamente con esponenti pari.

$$\frac{[\sin(x)]^2 \rightarrow \cos}{1 - [\cos(x)]^2}$$

Nota: questo operatore di conversione non è supportato nelle modalità di misurazione degli angoli in Gradi o Gradianti (gradi centesimali). Prima di utilizzarlo, accertarsi che la modalità angolare sia impostata su Radiani e che *Espr* non contenga riferimenti esplicativi ad angoli in gradi o gradanti.

cos() (Coseno)Tasto **cos(*Espr1*)**⇒*espressione*

In modalità angolo in gradi:

$$\cos\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

$$\cos(45) \quad \frac{\sqrt{2}}{2}$$

$$\cos(\{0,60,90\}) \quad \left\{1, \frac{1}{2}, 0\right\}$$

cos(*Listal*)⇒*lista***cos(*Espr1*)** restituisce sotto forma di espressione il coseno dell'argomento.**cos(*Listal*)** restituisce una lista dei coseni di tutti gli elementi di *Listal*.

Nota: conformemente alla modalità di misurazione degli angoli impostata, l'argomento viene interpretato come angolo in gradi, gradanti o radianti. È possibile utilizzare °, G o ' per escludere provvisoriamente la modalità d'angolo selezionata.

In modalità angolo in gradanti (gradi centesimali):

$$\cos(\{0,50,100\}) \quad \left\{1, \frac{\sqrt{2}}{2}, 0\right\}$$

In modalità angolo in radianti:

$$\cos\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

$$\cos(45^\circ) \quad \frac{\sqrt{2}}{2}$$

cos**(matriceQuadrata1)**⇒*matriceQuadrata*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:

In modalità angolo in radianti:

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

cos() (Coseno)

Tasto 

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(Espr1) \Rightarrow espressione

In modalità angolo in gradi:

cos⁻¹(List1) \Rightarrow lista

$$\cos^{-1}(1)$$

0

cos⁻¹(Espr1) restituisce sotto forma di espressione l'angolo il cui coseno è *Espr1*.

In modalità angolo in gradi (gradi centesimali):

cos⁻¹(List1) restituisce la lista dell'inversa dei coseni di ciascun elemento di *List1*.

$$\cos^{-1}(0)$$

100

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

In modalità angolo in radianti:

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

$$\cos^{-1}(\{0,0.2,0.5\}) \quad \left\{ \frac{\pi}{2}, 1.36944, 1.0472 \right\}$$

cos⁻¹
(matriceQuadrata1) \Rightarrow *matriceQuadrata*

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

cos⁻¹() (Arcocoseno)

Tasto

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.

$$\begin{aligned} \cos^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \\ \begin{bmatrix} 1.73485+0.064606 \cdot i & -1.49086+2.10514 \\ -0.725533+1.51594 \cdot i & 0.623491+0.77836 \cdot i \\ -2.08316+2.63205 \cdot i & 1.79018-1.27182 \cdot i \end{bmatrix} \end{aligned}$$

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

cosh() (Coseno iperbolico)

Catalogo >

cosh(Espr1)⇒espressione

cosh(Lista1)⇒lista

cosh(Espr1) restituisce sotto forma di espressione il coseno iperbolico dell'argomento.

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 gradi:

$$\cosh\left(\left(\frac{\pi}{4}\right)r\right) \quad \cosh(45)$$

In modalità angolo in radianti:

$$\begin{aligned} \cosh\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) \\ \begin{bmatrix} 421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628 \end{bmatrix} \end{aligned}$$

cosh⁻¹() (Arcocoseno iperbolico)

Catalogo >

cosh⁻¹(Espr1)⇒espressione

cosh⁻¹(Lista1)⇒lista

cosh⁻¹(Espr1) restituisce sotto forma di espressione l'inversa del coseno iperbolico dell'argomento.

$$\begin{aligned} \cosh^{-1}(1) &= 0 \\ \cosh^{-1}(\{1,2,1,3\}) &= \{0,1.37286,\cosh^{-1}(3)\} \end{aligned}$$

cosh⁻¹() (Arcocoseno iperbolico)

Catalogo > 

cosh⁻¹(List₁) restituisce una lista dell'inversa dei coseni iperbolici di ciascun elemento di *List₁*.

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

cosh⁻¹

(matriceQuadrata₁) \Rightarrow 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:

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

$$\begin{array}{ll} 2.52503+1.73485 \cdot i & -0.009241-1.4908i \\ 0.486969-0.725533 \cdot i & 1.66262+0.623491i \\ -0.322354-2.08316 \cdot i & 1.26707+1.79018i \end{array}$$

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

cot() (Cotangente)

Tasto 

cot(Espr₁) \Rightarrow espressione

cot(List₁) \Rightarrow lista

Restituisce la cotangente dell'*espressione₁* oppure restituisce una lista delle cotangenti di tutti gli elementi di *list₁*.

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

In modalità angolo in gradi:

$$\cot(45)$$

1

In modalità angolo in gradi (gradi centesimali):

$$\cot(50)$$

1

In modalità angolo in radianti:

$$\cot(\{1, 2, 1, 3\}) \quad \left\{ \frac{1}{\tan(1)}, -0.584848, \frac{1}{\tan(3)} \right\}$$

cot⁻¹() (Arcocotangente)

Tasto 

cot⁻¹(Espr₁) \Rightarrow espressione

cot⁻¹(List₁) \Rightarrow lista

In modalità angolo in gradi:

$$\cot^{-1}(1)$$

45

cot⁻¹(*l*) (Arcocotangente)

Tasto 

Restituisce l'angolo la cui cotangente è *Esprl* oppure restituisce una lista contenente l'inversa delle cotangenti di ciascun elemento di *Listal*.

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 centesimali:

cot⁻¹(1)

50

In modalità angolo in radianti:

cot⁻¹(1)

$\frac{\pi}{4}$

coth() (Cotangente iperbolica)

Catalogo > 

coth(*Esprl*)⇒espressione

coth(*Listal*)⇒lista

Restituisce la cotangente iperbolica di *Esprl* o restituisce una lista delle cotangenti iperboliche di tutti gli elementi di *Listal*.

coth(1.2)	1.19954
coth({1,3,2})	$\left\{ \frac{1}{\tanh(1)}, 1.00333 \right\}$

coth⁻¹(*l*) Arcocotangente iperbolica

Catalogo > 

coth⁻¹(*Esprl*)⇒espressione

coth⁻¹(*Listal*)⇒lista

Restituisce la cotangente iperbolica inversa di *Esprl* oppure restituisce una lista contenente l'inversa delle cotangenti iperboliche di ciascun elemento di *Listal*.

coth ⁻¹ (3.5)	0.293893
coth ⁻¹ ({-2,2,1,6})	$\left\{ \frac{-\ln(3)}{2}, 0.518046, \frac{\ln\left(\frac{7}{5}\right)}{2} \right\}$

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 254.

count(2,4,6)	3
count({2,4,6})	3
count(2,{4,6},{8 10 12 14})	7
count(1/2,3+4*i,undef,"hello",x+5.,sign(0))	2

Nell'ultimo esempio, sono contati solo 1/2 e 3+4*i. I restanti argomenti, presupponendo che *x* sia indefinito, non danno come risultato valori numerici.

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 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 254.

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({x^-2,x^-1,1,x,x^2},x)	1
--------------------------------	---

Conta il numero di elementi uguali a *x*; questo esempio presuppone che la variabile *x* sia indefinita.

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

Conta 1 e 3.

countif()**Catalogo >**

Nota: vedere anche **sumIf()**, pagina 193 e **frequency()**, pagina 84.

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()**Catalogo >**

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*.

Poli deve essere un polinomio in una variabile.

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

Nota: vedere anche **polyRoots()**, pagina 145.

polyRoots(y^3+1,y) { -1 }

cPolyRoots(y^3+1,y) {

$$\left\{-1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i\right\}$$
}

polyRoots(x^2+2x+1,x) { -1,-1 }

cPolyRoots({1,2,1}) { -1,-1 }

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

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*.

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

crossP({a1,b1},{a2,b2})

{ 0,0,a1·b2-a2·b1 }

crossP({0.1,2.2,-5},{1,-0.5,0})

{ -2.5,-5.,-2.25 }

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

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

csc() (Cosecante)

Tasto 

csc(*Espr1*) \Rightarrow espressione

csc(*Listal*) \Rightarrow lista

Restituisce la cosecante di *Espr1* oppure restituisce una lista contenente le cosecanti di tutti gli elementi in *Listal*.

In modalità angolo in gradi:

$$\csc(45) \quad \frac{\sqrt{2}}{2}$$

In modalità angolo in gradi centesimali:

$$\csc(50) \quad \frac{\sqrt{2}}{2}$$

In modalità angolo in radianti:

$$\csc\left(\left\{1, \frac{\pi}{2}, \frac{\pi}{3}\right\}\right) \quad \left\{\frac{1}{\sin(1)}, 1, \frac{2\sqrt{3}}{3}\right\}$$

csc⁻¹() (Cosecante inversa)

Tasto 

csc⁻¹(*Espr1*) \Rightarrow espressione

csc⁻¹(*Listal*) \Rightarrow lista

Restituisce l'angolo la cui cosecante è *Espr1* oppure restituisce una lista contenente l'inversa delle cosecanti di ciascun elemento di *Listal*.

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

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

In modalità angolo in gradi:

$$\csc^{-1}(1) \quad 90$$

In modalità angolo in gradi centesimali:

$$\csc^{-1}(1) \quad 100$$

In modalità angolo in radianti:

$$\csc^{-1}(\{1, 4, 6\}) \quad \left\{\frac{\pi}{2}, \sin^{-1}\left(\frac{1}{4}\right), \sin^{-1}\left(\frac{1}{6}\right)\right\}$$

csech (Cosecante iperbolica)

Catalogo > 

csech(*Espr1*) \Rightarrow espressione

csech(*Listal*) \Rightarrow lista

Restituisce la cosecante iperbolica di *Espr1* oppure restituisce una lista di cosecanti iperboliche di tutti gli elementi di *Listal*.

$$\operatorname{csch}(3) \quad \frac{1}{\sinh(3)}$$

$$\operatorname{csch}(\{1, 2, 1, 4\}) \quad \left\{\frac{1}{\sinh(1)}, 0.248641, \frac{1}{\sinh(4)}\right\}$$

csch⁻¹() (Cosecante iperbolica inversa)

Catalogo >

csch⁻¹(Espr1)⇒espressione**csch⁻¹(Listal)⇒lista**

Restituisce la cosecante iperbolica inversa di *Espr1* oppure restituisce una lista contenente le cosecanze iperboliche inverse di ciascun elemento di *Listal*.

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

csch ⁻¹ (1)	sinh ⁻¹ (1)
csch ⁻¹ ({1,2,1,3})	$\left\{ \sinh^{-1}(1), 0.459815, \sinh^{-1}\left(\frac{1}{3}\right) \right\}$

cSolve() (Risolvi in campo complesso)

Catalogo >

cSolve(Equazione, Var)⇒espressione booleana**cSolve(Equazione, Var=Campione)⇒espressione booleana****cSolve(Disequazione, Var)⇒espressione booleana**

cSolve(x ³ =-1,x)	
$x = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i$ or $x = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i$ or $x = -1$	
solve(x ³ =-1,x)	$x = -1$

Restituisce possibili soluzioni complesse di un'equazione o di una disequazione rispetto a *Var*. Il fine è quello di produrre tutte le possibili soluzioni reali e non reali. Anche se *Equazione* è reale, **cSolve()** ammette risultati non reali nel modo reale.

Sebbene tutte le variabili non definite che non terminano con un trattino basso (*_*) siano elaborate come se fossero reali, **cSolve()** può risolvere equazioni polinomiali a soluzioni complesse.

cSolve() (Risovi in campo complesso)

Catalogo >

cSolve() imposta temporaneamente il dominio complesso durante la soluzione, sebbene il dominio corrente sia reale. Nel dominio complesso, le potenze frazionarie con denominatori dispari utilizzano l'ambito principale invece di quello reale. Pertanto le soluzioni ottenute con **solve()** per le equazioni contenenti tali potenze frazionarie non sono necessariamente un sottoinsieme di quelle ottenute con **cSolve()**.

cSolve() inizia con metodi simbolici esatti. Con l'eccezione della modalità **Esatto**, **cSolve()** utilizza eventualmente anche la scomposizione in fattori complessa approssimata iterativa di polinomi.

Nota: vedere anche **cZeros()**, **solve()** e **zeros()**.

Nota: se *Equazione* è di tipo non polinomiale con funzioni quali **abs()**, **angle()**, **conj()**, **real()** o **imag()**, è necessario inserire un trattino basso (premere

ctrl) dopo *Var*. Per impostazione predefinita, una variabile viene considerata come valore reale.

Se si utilizza *var_*, la variabile verrà considerata come un numero complesso.

È necessario utilizzare *var_* anche per qualsiasi altra variabile in *Equazione* che potrebbe avere valori non reali. In caso contrario, si potrebbero ottenere dei risultati inaspettati.

**cSolve(Eqn1andEqn2
[and...],VarOCampione1,VarOCampione2
[, ...])** ⇒ espressione booleana

**cSolve(SistemaDiEquazioni,
VarOCampione1,VarOCampione2 [, ...])**
⇒ espressione booleana

$cSolve\left(x^3 = -1, x\right)$	false
$solve\left(x^3 = -1, x\right)$	$x = -1$

In modalità Mostra cifre impostata su Fissa 2:

$exact(cSolve(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3 = 0, x))$
$x \cdot (x^4 + 4 \cdot x^3 + 5 \cdot x^2 - 6) = 3$
$cSolve(Ans, x)$
$x = -1.11 + 1.07 \cdot i \text{ or } x = -1.11 - 1.07 \cdot i \text{ or } x = -2.3$

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

$cSolve(\operatorname{conj}(z_) = 1 + i, z_)$	$z_ = 1 - i$
---	--------------

Restituisce possibili soluzioni complesse ai sistemi di equazioni algebriche, dove ogni *varOCampione* specifica una variabile in base alla quale risolvere l'equazione.

In alternativa è possibile specificare un valore campione iniziale per la variabile. Ogni *varOCampione* deve avere la forma:

variabile

– oppure –

variabile = numero reale o non reale

Ad esempio, x è valido come pure $x=3+i$.

Se tutte le equazioni sono polinomiali e NON si indica alcun valore campione iniziale, **cSolve()** utilizza il metodo di eliminazione lessicale di Gröbner/Buchberger per tentare di determinare **tutte** le soluzioni complesse.

Le soluzioni complesse comprendono soluzioni reali e non reali, come nell'esempio a destra.

I sistemi di equazioni polinomiali possono avere variabili aggiuntive senza valori, ma rappresentano valori numerici dati che potrebbero essere sostituiti successivamente.

È inoltre possibile includere variabili risolutorie che non compaiono nelle equazioni. Queste soluzioni mostrano come le famiglie di soluzioni possano contenere costanti arbitrarie della forma ck , dove k è un suffisso intero compreso tra 1 e 255.

Nota: gli esempi che seguono usano il trattino basso (premere **ctrl** **—**) in modo che le variabili vengano considerate come complesse.

$$\text{cSolve}\left(u_{_} \cdot v_{_} - u_{_} = v_{_} \text{ and } v_{_}^2 = -u_{_}, \{u_{_}, v_{_}\}\right)$$

$$u_{_} = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v_{_} = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } u_{_} = \frac{1}{2} \downarrow$$

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

$$\text{cSolve}\left(u_{_} \cdot v_{_} - u_{_} = c_{_} \cdot v_{_} \text{ and } v_{_}^2 = -u_{_}, \{u_{_}, v_{_}\}\right)$$

$$u_{_} = \frac{-(\sqrt{1-4 \cdot c_{_}}+1)^2}{4} \text{ and } v_{_} = \frac{\sqrt{1-4 \cdot c_{_}}+1}{2} \text{ or } u_{_} \rightarrow$$

$$\text{cSolve}\left(u_{_} \cdot v_{_} - u_{_} = v_{_} \text{ and } v_{_}^2 = -u_{_}, \{u_{_}, v_{_}, w_{_}\}\right)$$

$$u_{_} = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v_{_} = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } w_{_} = c8 \text{ or } u_{_} \rightarrow$$

Per i sistemi polinomiali, i tempi di calcolo o l'insufficienza di memoria possono dipendere in gran parte dall'ordine in cui sono elencate le variabili risolutorie. Se la scelta iniziale esaurisce la memoria (o la pazienza), provare a ridisporre le variabili all'interno delle equazioni e/o della lista *varOCampione*.

Se non viene indicato alcun valore campione e se anche una sola equazione è non polinomiale per una qualsiasi variabile, ma tutte le equazioni sono lineari per tutte le variabili risolutorie **cSolve()** utilizza l'eliminazione gaussiana per tentare di determinare tutte le soluzioni.

Se un sistema non è né polinomiale per tutte le sue variabili né lineare per le variabili risolutorie, **cSolve()** determina al più una soluzione utilizzando un metodo di approssimazione iterativa. A tale scopo, il numero di variabili risolutorie deve essere uguale al numero di equazioni e tutte le altre variabili delle equazioni devono poter essere semplificate in numeri.

Spesso si rende necessario utilizzare un valore campione non reale per determinare una soluzione non reale. Per ottenere una convergenza, può essere necessario che il valore campione debba essere abbastanza prossimo alla soluzione.

$$\begin{aligned} \text{cSolve}\left(u_{_}+v_{_}=\mathbf{e}^{w_{_}} \text{ and } u_{_}-v_{_}=i, \{u_{_}, v_{_}\}\right) \\ u_{_}=\frac{\mathbf{e}^{w_{_}}+i}{2} \text{ and } v_{_}=\frac{\mathbf{e}^{w_{_}}-i}{2} \end{aligned}$$

$$\begin{aligned} \text{cSolve}\left(\mathbf{e}^{z_{_}}=w_{_} \text{ and } w_{_}=z_{_}^2, \{w_{_}, z_{_}\}\right) \\ w_{_}=0.494866 \text{ and } z_{_}=-0.703467 \end{aligned}$$

$$\begin{aligned} \text{cSolve}\left(\mathbf{e}^{z_{_}}=w_{_} \text{ and } w_{_}=z_{_}^2, \{w_{_}, z_{_}=1+i\}\right) \\ w_{_}=0.149606+4.8919 \cdot i \text{ and } z_{_}=1.58805+1 \cdot i \end{aligned}$$

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

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

Calcola la regressione polinomiale cubicay = a · x³+b · x²+c · x+d sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189).

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 254.

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(ListaI)⇒lista

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

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

cumulativeSum()

Catalogo >

cumulativeSum(*Matrice1*)⇒*matrice*

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

Un elemento vuoto (nullo) in *Listal* o *Matrice1* produce un elemento vuoto nella lista o matrice risultante. Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 254.

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

[2 2 3]►Cylind $\left[2\cdot\sqrt{2} \angle \frac{\pi}{4} 3\right]$

cZeros(*Espr*, *Var*)⇒*lista*

Restituisce una lista di possibili valori reali e non reali per *Var* per i quali *Espr*=0.

cZeros() calcola **expList(cSolve(*Espr*=0,*Var*),*Var*)**. Per il resto, **cZeros()** è simile a **zeros()**.

Nota: vedere anche **cSolve()**, **solve()** e **zeros()**.

Nota: se *Espr* è di tipo non polinomiale con funzioni quali **abs()**, **angle()**, **conj()**, **real()**, or **imag()**, è necessario inserire un trattino basso (premere **ctrl** **—**) dopo *Var*. Per impostazione predefinita, una variabile viene considerata come valore reale. Se si utilizza *var_*, la variabile verrà considerata come un numero complesso.

È necessario utilizzare *var_* anche per qualsiasi altra variabile in *Espr* che potrebbe avere valori non reali. In caso contrario, si potrebbero ottenere dei risultati inaspettati.

cZeros({*Espr*1, *Espr*2 [, ...] }, {*VarOCampione*1, *VarOCampione*2 [, ...] })⇒*matrice*

Restituisce le possibili posizioni in cui le espressioni sono simultaneamente zero. Ogni *VarOCampione* specifica un'incognita il cui valore si desidera calcolare.

In alternativa è possibile specificare un valore campione iniziale per la variabile. Ogni *varOCampione* deve avere la forma:

variabile

– oppure –

variabile = *numero reale o non reale*

Ad esempio, *x* è valido come pure *x*=3+*i*.

In modalità Mostra cifre impostata su Fissa 3:

cZeros($x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3, x$)
 $\{ -1.1138 + 1.07314 \cdot i, -1.1138 - 1.07314 \cdot i, -2, \dots \}$

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

cZeros($\text{conj}(z) - 1 - i, z$) {1-i}

Se tutte le equazioni sono polinomiali e NON si indica alcun valore campione iniziale, **cZeros()** utilizza il metodo di eliminazione lessicale di Gröbner/Buchberger per tentare di determinare **tutti** gli zeri complessi.

Gli zeri complessi comprendono zeri reali e non reali, come nell'esempio a destra.

Ciascuna riga della matrice risultante rappresenta uno zero alternativo, con i componenti ordinati come nella lista *VarOCampione*. Per estrarre una riga, indicizzare la matrice per [riga].

Le espressioni polinomiali simultanee possono avere variabili aggiuntive senza valori, ma rappresentano valori numerici dati che potrebbero essere sostituiti successivamente.

È inoltre possibile includere variabili incognite che non compaiono nelle espressioni. Questi zeri mostrano come le famiglie di zeri possano contenere costanti arbitrarie della forma ck , dove k è un suffisso intero compreso tra 1 e 255.

Per i sistemi polinomiali, i tempi di calcolo o l'insufficienza di memoria possono dipendere in gran parte dall'ordine in cui sono elencate le incognite. Se la scelta iniziale esaurisce la memoria (o la pazienza), provare a ridisporre le variabili all'interno delle espressioni e/o della lista *VarOCampione*.

Nota: gli esempi che seguono usano il trattino basso (premere **ctr** **–**) in modo che le variabili vengano considerate come complesse.

$$\text{cZeros}\left(\left\{u_{_}v_{_}-u_{_}v_{_}^2+u_{_}\right\}, \{u_{_}, v_{_}\}\right)$$

$$\begin{bmatrix} 0 & 0 \\ \frac{1-\sqrt{3}}{2} \cdot i & \frac{1+\sqrt{3}}{2} \cdot i \\ \frac{2}{2} & \frac{2}{2} \\ \frac{1+\sqrt{3}}{2} \cdot i & \frac{1-\sqrt{3}}{2} \cdot i \\ \frac{2}{2} & \frac{2}{2} \end{bmatrix}$$

Estrarre la riga 2:

$$\text{Ans}[2] \quad \begin{bmatrix} \frac{1-\sqrt{3}}{2} \cdot i & \frac{1+\sqrt{3}}{2} \cdot i \end{bmatrix}$$

$$\text{cZeros}\left(\left\{u_{_}v_{_}-u_{_}c_{_}v_{_}^2+u_{_}\right\}, \{u_{_}, v_{_}\}\right)$$

$$\begin{bmatrix} 0 & 0 \\ \frac{-(\sqrt{1-4 \cdot c_{_}}-1)^2}{4} & \frac{-(\sqrt{1-4 \cdot c_{_}}-1)}{2} \\ \frac{4}{4} & \frac{2}{2} \\ \frac{-(\sqrt{1-4 \cdot c_{_}}+1)^2}{4} & \frac{\sqrt{1-4 \cdot c_{_}}+1}{2} \end{bmatrix}$$

$$\text{cZeros}\left(\left\{u_{_}v_{_}-u_{_}v_{_}^2+u_{_}\right\}, \{u_{_}, v_{_}, w_{_}\}\right)$$

$$\begin{bmatrix} 0 & 0 & \text{c4} \\ \frac{1-\sqrt{3}}{2} \cdot i & \frac{1+\sqrt{3}}{2} \cdot i & \text{c4} \\ \frac{2}{2} & \frac{2}{2} & \text{c4} \\ \frac{1+\sqrt{3}}{2} \cdot i & \frac{1-\sqrt{3}}{2} \cdot i & \text{c4} \\ \frac{2}{2} & \frac{2}{2} & \text{c4} \end{bmatrix}$$

cZeros() (Zeri complessi)

Catalogo >

Se non viene indicato alcun valore campione e se anche una sola espressione è non polinomiale per una qualsiasi variabile ma tutte le espressioni sono lineari per tutte le incognite, **cZeros()** utilizza l'eliminazione gaussiana per tentare di determinare tutti gli zeri.

Se un sistema non è né polinomiale per tutte le sue variabili né lineare per le incognite, **cZeros()** determina al più uno zero utilizzando un metodo di approssimazione iterativa. A tale scopo, il numero di incognite deve essere uguale al numero di espressioni e tutte le altre variabili delle espressioni devono poter essere semplificate in numeri.

Spesso si rende necessario utilizzare un valore campione non reale per determinare uno zero non reale. Per ottenere una convergenza, può essere necessario che il valore campione debba essere abbastanza prossimo allo zero.

$$\begin{aligned} \text{cZeros}\left(\left\{u_{-}+v_{-}-e^{w_{-}}, u_{-}-v_{-}-i\right\}, \left\{u_{-}, v_{-}\right\}\right) \\ \left[\frac{e^{w_{-}+i}}{2} \quad \frac{e^{w_{-}-i}}{2}\right] \end{aligned}$$

$$\begin{aligned} \text{cZeros}\left(\left\{e^{z_{-}-w_{-}}, w_{-}-z_{-}^2\right\}, \left\{w_{-}, z_{-}\right\}\right) \\ [0.494866 \quad -0.703467] \end{aligned}$$

$$\begin{aligned} \text{cZeros}\left(\left\{e^{z_{-}-w_{-}}, w_{-}-z_{-}^2\right\}, \left\{w_{-}, z_{-}=1+i\right\}\right) \\ [0.149606+4.8919 \cdot i \quad 1.58805+1.54022 \cdot i] \end{aligned}$$

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)

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

GGMM.AA (formato usato generalmente in Europa)

►DD (Visualizza angolo decimale)

Valore1 ►DD⇒*valore*

Listal ►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 centesimali:

1►DD	$\frac{9}{10}$ °
------	------------------

In modalità angolo in radianti:

(1.5)►DD	85.9437°
----------	----------

►Decimal (Decimale)

Espr1 ►Decimal⇒*espressione*

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

Listal ►Decimal⇒*espressione*

Matrice1 ►Decimal⇒*espressione*

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.

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 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.

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 $g(x,y)=\text{Func}$	Done
If $x>y$ Then	
Return x	
Else	
Return y	
EndIf	
EndFunc	
$g(3,-7)$	3

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

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

Nota: vedere anche **Define LibPriv**, pagina 56 e **Define LibPub**, pagina 56.

Define LibPriv (Definisci libreria privata)

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.

Nota: vedere anche **Define**, pagina 54 e **Define LibPub**, pagina 56.

Define LibPub (Definisci libreria pubblica)

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

Define LibPub (Definisci libreria pubblica)

Catalogo >

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 54 e **Define LibPriv**, pagina 56.

deltaList()

Vedere Δ List(), pagina 110.

deltaTmpCnv()

Vedere Δ tmpCnv(), pagina 203.

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 212.

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.*

$2 \rightarrow a$	2
$(a+2)^2$	16
DelVar <i>a</i>	Done
$(a+2)^2$	$(a+2)^2$

<i>aa.a:=45</i>	45
<i>aa.b:=5.67</i>	5.67
<i>aa.c:=78.9</i>	78.9
getVarInfo()	$\begin{bmatrix} aa.a & "NUM" & "0" \\ aa.b & "NUM" & "0" \\ aa.c & "NUM" & "0" \end{bmatrix}$
DelVar <i>aa.</i>	Done
getVarInfo()	"NONE"

delVoid()

Catalogo >

delVoid(*Lista1*) \Rightarrow *lista*

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

Restituisce una lista che ha il contenuto di *List1* con tutti gli elementi vuoti (nulli) rimossi.

Per ulteriori informazioni sugli elementi vuoti, vedere, pagina 254.

derivative()Vedere *d()*, pagina 237.**deSolve() Soluzione di equazioni differenziali**

**deSolve(ODE1°O2°Ordine, Var,
varDipendente)⇒soluzione generale**

Restituisce un'equazione che specifica in modo esplicito o implicito una soluzione generale per l'equazione differenziale ordinaria (ODE) di 1° o 2° ordine. Nell'ODE:

- Usare il simbolo di primo (premere) per indicare la derivata prima della variabile dipendente rispetto alla variabile indipendente.
- Usare due simboli di primo per indicare la derivata seconda corrispondente.

Il simbolo di primo viene utilizzato solo per le derivate all'interno di deSolve(). Negli altri casi utilizzare **d()**.

La soluzione generale di un'equazione di primo ordine contiene una costante arbitraria della forma *ck*, dove *k* è un suffisso intero compreso tra 1 e 255. La soluzione di un'equazione di secondo ordine contiene due costanti di questo tipo.

Applicare **solve()** ad una soluzione implicita per tentare di convertirla in una o più soluzioni esplicite equivalenti.

$$\begin{aligned} \text{deSolve}\left(y''+2 \cdot y'+y=x^2, x, y\right) \\ y=(c3 \cdot x+c4) \cdot e^{-x}+x^2-4 \cdot x+6 \\ \text{right}(Ans) \rightarrow \text{temp} \quad (c3 \cdot x+c4) \cdot e^{-x}+x^2-4 \cdot x+6 \\ \frac{d^2}{dx^2}\{\text{temp}\}+2 \cdot \frac{d}{dx}\{\text{temp}\}+\text{temp}-x^2 \quad 0 \end{aligned}$$

DelVar *temp*

Done

$$\begin{aligned} \text{deSolve}\left(y'=\{\cos(y)\}^2 \cdot x, x, y\right) \quad \tan(y)=\frac{x^2}{2}+c4 \end{aligned}$$

deSolve() Soluzione di equazioni differenziali

Catalogo >

Quando si confrontano i risultati ottenuti con le soluzioni date dal libro di testo o dal manuale, tenere presente che metodi diversi introducono costanti arbitrarie in punti diversi dei calcoli, il che può dare soluzioni generali diverse.

deSolve

(ODE1°Ordine and condizioneIniziale, Var, varDipendente) ⇒ soluzione particolare

Restituisce una soluzione particolare che soddisfa *ODE1°Ordine e condizioneIniziale*. Di solito ciò risulta più semplice che non determinare una soluzione generale, sostituire i valori iniziali, calcolare la soluzione in base ad una costante arbitraria e quindi sostituire il valore nella soluzione generale.

condizioneIniziale è un'equazione della forma:

varDipendente (valoreInizialeDipendente) = valoreInizialeDipendente

valoreInizialeIndipendente e valoreInizialeDipendente possono essere variabili come *x0* e *y0* che non contengono alcun valore. Una differenziazione implicita può servire a verificare le soluzioni implicite.

deSolve

(ODE2°Ordine and condIniziale1 and condIniziale2, Var, varDipendente) ⇒ soluzione particolare

Restituisce una soluzione particolare che soddisfa *ODE2°ordine* e che ha un valore specifico della variabile dipendente e la derivata prima nello stesso punto.

Per *condIniziale1*, utilizzare la forma:

varDipendente (valoreInizialeDipendente) = valoreInizialeDipendente

solve(Ans,y)	$y=\tan^{-1}\left(\frac{x^2+2\cdot c4}{2}\right)+n3\cdot \pi$
--------------	---

<i>Ans c4=c-1 and n3=0</i>	$y=\tan^{-1}\left(\frac{x^2+2\cdot(c-1)}{2}\right)$
----------------------------	---

$\sin(y)=\left(y \cdot e^x + \cos(y)\right) \cdot y' \rightarrow ode$	
$\sin(y)=\left(e^x \cdot y + \cos(y)\right) \cdot y'$	
deSolve(<i>ode</i> and $y(0)=0,x,y) \rightarrow soln$	
$\frac{(2 \cdot \sin(y)+y^2)}{2}=(e^x-1) \cdot e^{-x} \cdot \sin(y)$	
<i>soln x=0 and y=0</i>	true
<i>ode y'=impDiff(soln,x,y)</i>	true
DelVar <i>ode,soln</i>	Done

deSolve($y''=y^{\frac{1}{2}}$ and $y(0)=0$ and $y'(0)=0,t,y$)	$\frac{2 \cdot y^{\frac{3}{4}}}{3}=t$
---	---------------------------------------

solve(Ans,y)	$y=\frac{2^{\frac{3}{2}} \cdot (3 \cdot t)^{\frac{3}{2}}}{4}$ and $t \geq 0$
--------------	--

deSolve() Soluzione di equazioni differenziali

Catalogo >

Per *condIniziale2*, utilizzare la forma:

depVar (*valoreInizialeIndipendente*) =
valore l°DerivataIniziale

deSolve

(

ODE2°Ordine

and*condLimite1***and***condLimite2*, *Var*,
varDipendente) \Rightarrow soluzione particolare

Restituisce una soluzione particolare che soddisfa *ODE2°Ordine* e che ha valori specificati in due punti diversi.

$$\text{deSolve}\left(w'' - \frac{2 \cdot w'}{x} + \left(9 + \frac{2}{x^2}\right) \cdot w = x \cdot e^x \text{ and } w\left(\frac{\pi}{6}\right) = 0 \text{ and } w\left(\frac{\pi}{3}\right) = 0, x, w\right)$$

$$w = \frac{x \cdot e^x}{(\ln(e))^2 + 9} + \frac{e^{\frac{\pi}{3}} \cdot x \cdot \cos(3 \cdot x)}{(\ln(e))^2 + 9} - \frac{e^{\frac{\pi}{6}} \cdot x \cdot \sin(3 \cdot x)}{(\ln(e))^2 + 9}$$

$$\text{deSolve}(y''=x \text{ and } y(0)=1 \text{ and } y'(2)=3, x, y)$$

$$y = \frac{x^3}{6} + x + 1$$

$$\text{deSolve}(y''=2 \cdot y' \text{ and } y(3)=1 \text{ and } y'(4)=2, x, y)$$

$$y = e^{2 \cdot x - 8} - e^{-2 + 1}$$

det() (Determinante)

Catalogo >

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.

$$\det\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$a \cdot d - b \cdot c$$

$$\det\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$-2$$

$$\det\left(\text{identity}(3) - x \cdot \begin{bmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ -6 & -2 & 7 \end{bmatrix}\right)$$

$$-(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$$

$$\begin{bmatrix} 1.e20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow mat1$$

$$\begin{bmatrix} 1.e20 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\det(mat1)$$

$$0$$

$$\det(mat1, 1)$$

$$1.e20$$

- 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:
 $5E^{-14} \cdot \max(\dim(matriceQuadrata)) \cdot \text{rowNorm}$

det() (Determinante)

Catalogo >

(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*.

$$\begin{array}{|c c c|} \hline 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \\ \hline \end{array} \quad \begin{array}{|c c c|} \hline 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \\ \hline \end{array}$$

diag(*Ans*)

matriceQuadrata deve essere quadrata.

dim() (Dimensione)

Catalogo >

dim(Lista)⇒intero

dim({0,1,2})

3

Restituisce le dimensioni di *Listा*.

dim(Matrice)⇒lista

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

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

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

►DMS (Gradi/primi/secondi)

Espr ►DMS

Lista ►DMS

Matrice ►DMS

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 246.

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.

In modalità angolo in gradi:

{45.371}►DMS	45°22'15.6"
{ { 45.371,60 } }►DMS	{ 45°22'15.6",60° }

domain()

Catalogo >

domain(*Espr1, Var*)⇒*espressione*

Restituisce il dominio di *Espr1* rispetto alla *Var*.

domain() può essere utilizzato per esaminare i domini delle funzioni. È limitato al dominio reale e finito.

Questa funzionalità ha dei limiti dovuti alle lacune della semplificazione della computer algebra e degli algoritmi del risolutore.

Ci sono funzioni che non possono essere utilizzate come argomenti per **domain()**, indipendentemente dal fatto se appaiono esplicitamente oppure nell'ambito di variabili e funzioni definite dall'utente. Nell'esempio che segue, l'espressione non può essere semplificata dal momento che $\int()$ è una funzione non consentita.

$$\text{domain}\left(\begin{cases} \frac{x}{t} \\ \frac{1}{t} \end{cases}, dt, x\right) \rightarrow \text{domain}\left(\begin{cases} \frac{x}{t} \\ \frac{1}{t} \end{cases}, dt, x\right)$$

dominantTerm() (Termine dominante)

Catalogo >

dominantTerm(*Espr1, Var [, Punto]*)⇒*espressione*

dominantTerm(*Espr1, Var [, Punto]*) | *Var>Punto*⇒*espressione*

dominantTerm(*Espr1, Var [, Punto]*) | *Var<Punto*⇒*espressione*

domain(x^2, x)	$-\infty < x < \infty$
domain($\frac{x+1}{x^2+2 \cdot x}, x$)	$x \neq -2 \text{ and } x \neq 0$
domain($(\sqrt{x})^2, x$)	$0 \leq x < \infty$
domain($\frac{1}{x+y}, y$)	$y \neq -x$

dominantTerm($\tan(\sin(x)) - \sin(\tan(x)), x$)

$$\frac{x^7}{30}$$

dominantTerm($\frac{1 - \cos(x-1)}{(x-1)^3}, x, 1$)

$$\frac{1}{2 \cdot (x-1)}$$

dominantTerm($x^{-2} \cdot \tan(x^3), x$)

$$\frac{1}{x^3}$$

dominantTerm($\ln(x^x - 1) \cdot x^{-2}, x$)

$$\frac{\ln(x \cdot \ln(x))}{x^2}$$

Restituisce il termine dominante di una rappresentazione di serie di potenze di *Espr1* sviluppate intorno a *Punto*. Il termine dominante è quello la cui grandezza cresce più rapidamente in prossimità di *Var* = *Punto*. La potenza risultante di (*Var* - *Punto*) può avere esponente negativo e/o frazionario. Il coefficiente di questa potenza può includere logaritmi di (*Var* - *Punto*) e altre funzioni di *Var* che sono dominate da tutte le potenze di (*Var* - *Punto*) aventi lo stesso segno esponenziale.

Il valore predefinito di Punto è 0. Punto può essere ∞ o $-\infty$, in questi casi il termine dominante sarà quello avente l'esponente più grande di *Var* anziché l'esponente più piccolo di *Var*.

dominantTerm(...) restituisce
“**dominantTerm(...)**” se non è in grado di determinare una tale rappresentazione, come ad esempio per singolarità essenziali quali $\sin(1/z)$ in $z=0$, $e^{-1/z}$ in corrispondenza di $z=0$ o e^z in $z = \infty$ o $-\infty$.

Se la serie o una delle sue derivate presenta una discontinuità in corrispondenza del *Punto*, il risultato potrebbe contenere sottoespressioni della forma $\text{sign}(\dots)$ o $\text{abs}(\dots)$ per una variabile di sviluppo reale o della forma $(-1)^{\text{floor}(\dots\text{angle}(\dots))}$ per una variabile di sviluppo complessa, che termina con “_”. Se si pensa di utilizzare il termine dominante solo per i valori da una sola parte di *Punto*, aggiungere a **dominantTerm(...)** quello appropriato tra “| *Var* > *Punto*”, “| *Var* < *Punto*”, “| *Var* ≥ *Punto*” o “*Var* ≤ *Punto*” per ottenere un risultato più semplice.

dominantTerm() si distribuisce su liste e matrici indicate come primo argomento.

$\text{dominantTerm}\left(e^{\frac{-1}{z}}, z\right)$	$\left(\frac{-1}{z}\right)$
$\text{dominantTerm}\left(e^{\frac{-1}{z}}, z, 0\right)$	$\left(\frac{-1}{z}\right)$
$\text{dominantTerm}\left(\left(1+\frac{1}{n}\right)^n, n, \infty\right)$	e
$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 0\right)$	$\frac{\pi \cdot \text{sign}(x)}{2}$
$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x\right) _{x>0}$	$\frac{\pi}{2}$

dominantTerm() è utile per conoscere l'espressione più semplice possibile che sia asintotica rispetto a un'altra espressione come *Var* → *Punto*. **dominantTerm()** è utile inoltre quando non è ovvio quale sarà il grado del primo termine diverso da zero di una serie e non si desidera procedere per ipotesi iterative, sia interattivamente che per mezzo di un loop di programma.

Nota: vedere anche **series()**, pagina 171.

dotP() (Prodotto scalare)

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.

dotP($\{a,b,c\}, \{d,e,f\}$)	$a \cdot d + b \cdot e + c \cdot f$
dotP($\{1,2\}, \{5,6\}$)	17
dotP($\begin{bmatrix} a & b & c \end{bmatrix}, \begin{bmatrix} d & e & f \end{bmatrix}$)	$a \cdot d + b \cdot e + c \cdot f$
dotP($\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 4 & 5 & 6 \end{bmatrix}$)	32

E

e^() (Funzione esponenziale)

Tasto

e^(Espr1)⇒espressione

Restituisce e elevato alla potenza di *Espr1*.

Nota: vedere anche **e** **modello di funzione esponenziale**, pagina 6.

e^1	e
$e^{1.}$	2.71828
e^{3^2}	e^9

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

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

e^() (Funzione esponenziale)

Tasto

e^A(List1)⇒lista

Restituisce e elevato alla potenza di ciascun elemento di *List1*.

e^A(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.

e^{1,1..0,5}

{e,2.71828,1.64872}

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

$$\begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$$

eff()

Catalogo >

eff(tassoNominal, CpY)⇒valore

eff(5.75,12)

5.90398

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

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

Nota: vedere anche **nom()**, pagina 131.

eigVc() (Autowettore)

Catalogo >

eigVc(matriceQuadrata)⇒matrice

In modalità formato rettangolare complesso:

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:

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

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

eigVc(m1)

$$\begin{bmatrix} -0.800906 & 0.767947 & 0 \\ 0.484029 & 0.573804+0.052258\cdot i & 0.5738 \\ 0.352512 & 0.262687+0.096286\cdot i & 0.2626 \end{bmatrix}$$

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

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

eigVc() (Autovettore)

Catalogo >

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:

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

$$\begin{aligned} \text{eigVl}(m1) \\ \{-4.40941, 2.20471 + 0.763006 \cdot i, 2.20471 - 0 \cdot i\} \end{aligned}$$

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

Else

Vedere If, pagina 94.

Elsef

Catalogo >

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 $g(x)=\text{Func}$
If $x \leq -5$ Then
Return 5
Elseif $x > -5$ and $x < 0$ Then
Return $\neg x$
Elseif $x \geq 0$ and $x \neq 10$ Then
Return x
Elseif $x = 10$ Then
Return 3
EndIf
EndFunc

Done

EndFor**Vedere For, pagina 81.****EndFunc****Vedere Func, pagina 85.****EndIf****Vedere If, pagina 94.****EndLoop****Vedere Loop, pagina 117.****EndPrgm****Vedere Prgm, pagina 147.****EndTry****Vedere Try, pagina 205.****EndWhile****Vedere While, pagina 216.****euler ()****Catalogo >**

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

Equazione differenziale:

 $y' = 0.001 * y * (100 - y)$ e $y(0) = 10$

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

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

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

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

Utilizza il metodo di Eulero per risolvere il sistema

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

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.

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

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

ListaDiExpr 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*.

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*).

Confrontare il precedente risultato con la soluzione esatta del CAS ottenuta utilizzando *deSolve()* e *seqGen()*:

$$deSolve(y=0.001 \cdot y \cdot (100-y) \text{ and } y(0)=10, t, y)$$

$$y=\frac{100 \cdot (1.10517)^t}{(1.10517)^t+9.}$$

$$seqGen\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t+9.}, t, y, \{0, 100\}\right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189\}$$

Sistema di equazioni

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

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

$$euler\left(\begin{cases} y1'= -y1 + 0.1 \cdot y1 \cdot y2 \\ y2'= 3 \cdot y2 - y1 \cdot y2 \end{cases}, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right)$$

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. & 5. \\ 2. & 1. & 1. & 3. & 27. & 243. \\ 5. & 10. & 30. & 90. & 90. & -2070. \end{bmatrix}$$

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()

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.

Menu Hub

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

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

Reimpostare l'elemento blu su OFF.

Send "SET COLOR.BLUE OFF"	<i>Done</i>
---------------------------	-------------

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.

<i>fadein()</i>	<i>Done</i>
-----------------	-------------

eval ()

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 87), **GetStr** (pagina 91) e **Send** (pagina 168).

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

exact() (Esatto)

exact(*Espresione* [, *Tolleranza*])⇒*espressione*

exact(*Lista* [, *Tolleranza*])⇒*lista*

exact(*Matrice* [, *Tolleranza*])⇒*matrice*

Utilizza l'aritmetica della modalità Esatto per restituire, se possibile, l'equivalente razionale dell'argomento.

Tolleranza specifica la tolleranza per la conversione; l'impostazione predefinita è 0 (zero).

Catalogo >

<i>exact(0.25)</i>	$\frac{1}{4}$
<i>exact(0.333333)</i>	$\frac{333333}{1000000}$
<i>exact(0.333333,0.001)</i>	$\frac{1}{3}$
<i>exact(3.5·x+y)</i>	$\frac{7·x}{2}+y$
<i>exact({0.2,0.33,4.125})</i>	$\left\{\frac{1}{5}, \frac{33}{100}, \frac{33}{8}\right\}$

Exit (Esci)**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.

Catalogo >

Descrizione della funzione:

Define <i>g()</i> =Func	<i>Done</i>
Local <i>temp,i</i>	
0→ <i>temp</i>	
For <i>i</i> ,1,100,1	
<i>temp</i> + <i>i</i> → <i>temp</i>	
If <i>temp</i> >20 Then	
Exit	
EndIf	
EndFor	
EndFunc	
<i>g()</i>	21

Espr ►exp

Rappresenta *Espr* rispetto all'esponente naturale e . È un operatore di conversione della visualizzazione. Può essere utilizzato solo alla fine della riga di introduzione.

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

$\frac{d}{dx} \left(e^x + e^{-x} \right)$	$2 \cdot \sinh(x)$
$2 \cdot \sinh(x) \blacktriangleright \text{exp}$	$e^x - e^{-x}$

exp() (e alla potenza)

Tasto

exp(*Espr1*)⇒espressione

Restituisce e elevato alla potenza di *Espr1*.

Restituisce e elevato alla potenza di *Valore1*.

e^1	e
$e^{1.}$	2.71828
e^3^2	e^9

Nota: vedere anche **e** modello di funzione esponenziale, pagina 6.

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(*Listal*)⇒lista

$e^{\{1,1.,0.5\}}$	$\{e, 2.71828, 1.64872\}$
--------------------	---------------------------

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

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()**.

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

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

exp►list()**Catalogo >** **exp►list(*Espr*,*Var*)⇒*lista***

Ricerca in *Espr* le equazioni separate dalla parola "or" e restituisce una lista contenente la parte destra delle equazioni nella forma *Var*=*Espr*. In questo modo viene semplificato il compito di estrarre alcuni valori delle soluzioni inclusi nei risultati delle funzioni **solve()**, **cSolve()**, **fMin()** e **fMax()**.

Nota: **exp►list()** non è necessario con le funzioni **zeros** e **cZeros()** in quanto queste restituiscono direttamente una lista contenente i valori delle soluzioni.

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

$\text{solve}(x^2-x-2=0,x)$	$x=-1 \text{ or } x=2$
exp►list(solve(x^2-x-2=0,x),x)	{ -1,2 }

expand() (Espandi)**Catalogo >** **expand(*Espr1* [, *Var*])⇒*espressione***

$$\text{expand}((x+y+1)^2) \\ x^2+2 \cdot x \cdot y+2 \cdot x+y^2+2 \cdot y+1$$

expand(*Lista1* [, *Var*])⇒*lista*

$$\text{expand}\left(\frac{x^2-x+y^2-y}{x^2 \cdot y^2-x^2 \cdot y-x \cdot y^2+x \cdot y}\right)$$

expand(*Matrice1* [, *Var*])⇒*matrice*

$$\frac{1}{x-1}-\frac{1}{x}+\frac{1}{y-1}-\frac{1}{y}$$

expand(*Espr1*) restituisce l'espansione di *Espr1* relativamente a tutte le sue variabili. Si tratta di un'espansione polinomiale per i polinomi e di una parziale espansione di frazioni per le espressioni razionali.

La funzione **expand()** permette di trasformare *Espr1* in una somma e/o differenza di termini semplici. Al contrario, **factor()** permette di trasformare *Espr1* in un prodotto e/o in un quoziente di fattori semplici.

expand(*Espr1,Var*) restituisce l'espansione di *Espr1* relativamente a *Var*. Le potenze simili di *Var* sono ridotte. I termini ed i rispettivi fattori sono ordinati considerando *Var* la variabile principale. Si può verificare accidentalmente la scomposizione in fattori o l'espansione dei coefficienti ridotti.

Questo procedimento, rispetto all'omissione di *Var*, permette di risparmiare tempo, memoria e spazio sullo schermo, rendendo inoltre l'espressione più comprensibile.

Anche in presenza di una sola variabile, *Var* permette di rendere più completa la scomposizione in fattori del denominatore utilizzata per la parziale espansione di frazioni.

Suggerimento: per le espressioni razionali, **propFrac()** costituisce un'alternativa più veloce ma meno completa di **expand()**.

Nota: vedere anche **comDenom()** per l'espansione di un numeratore di una frazione con denominatore espanso.

expand(*Espr1,[Var]*) distribuisce inoltre logaritmi e potenze frazionarie indipendentemente da *Var*. Per una maggiore distribuzione dei logaritmi e delle potenze frazionarie, possono essere richieste condizioni per garantire che alcuni fattori siano non negativi.

expand(*Espr1,[Var]*) distribuisce inoltre valori assoluti, **sign()** e gli esponenziali, indipendentemente da *Var*.

Nota: vedere anche **tExpand()** per l'espansione di funzioni trigonometriche con angoli somma e multipli di angoli.

$\text{expand}\left(\frac{(x+y+1)^2}{x^2 \cdot y}, y\right)$	$y^2 + 2 \cdot y \cdot (x+1) + (x+1)^2$
$\text{expand}\left(\frac{(x+y+1)^2}{x^2 \cdot x}, x\right)$	$x^2 + 2 \cdot x \cdot (y+1) + (y+1)^2$
$\text{expand}\left(\frac{x^2 - x + y^2 - y}{x^2 \cdot y^2 - x^2 \cdot y - x \cdot y^2 + x \cdot y}, y\right)$	$\frac{1}{y-1} - \frac{1}{y} + \frac{1}{x \cdot (x-1)}$
$\text{expand}(Ans, x)$	$\frac{1}{x-1} - \frac{1}{x} + \frac{1}{y \cdot (y-1)}$

$\text{expand}\left(\frac{x^3 + x^2 - 2}{x^2 - 2}, x\right)$	$\frac{2 \cdot x}{x^2 - 2} + x + 1$
$\text{expand}(Ans, x)$	$\frac{1}{x - \sqrt{2}} + \frac{1}{x + \sqrt{2}} + x + 1$

$\ln(2 \cdot x \cdot y) + \sqrt{2 \cdot x \cdot y}$	$\ln(2 \cdot x \cdot y) + \sqrt{2 \cdot x \cdot y}$
$\text{expand}(Ans)$	$\ln(x \cdot y) + \sqrt{2 \cdot \sqrt{x} \cdot \sqrt{y}} + \ln(2)$
$\text{expand}(Ans), y \geq 0$	$\ln(x) + \sqrt{2 \cdot \sqrt{x} \cdot \sqrt{y}} + \ln(y) + \ln(2)$
$\text{sign}(x \cdot y) + x \cdot y + e^{2 \cdot x + y}$	$\text{sign}(x \cdot y) + x \cdot y + e^{2 \cdot x + y}$
$\text{expand}(Ans)$	$e^{2 \cdot x + y} + \text{sign}(x \cdot y) + x \cdot y $
	$\text{sign}(x) \cdot \text{sign}(y) + x \cdot y + (e^x)^2 \cdot e^y$

expr() (Da stringa a espressione)

Catalogo >

expr(Stringa)⇒espressione

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

expr("1+2+x^2+x")	$x^2 + x + 3$
expr("expand((1+x)^2)")	$x^2 + 2 \cdot x + 1$
"Define cube(x)=x^3" → <i>funcstr</i>	"Define cube(x)=x^3"
expr(<i>funcstr</i>)	<i>Done</i>
<i>cube</i> (2)	8

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 189).

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 254.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot (b)^x$
stat.a, stat.b	Coefficienti di regressione

Variabile di output	Descrizione
stat.r ²	Coefficiente di determinazione lineare di dati trasformati
stat.r	Coefficiente di correlazione per dati trasformati ($x, \ln(y)$)
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 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>

F

factor() (Fattorizza)

Catalogo >

factor(*Espr1[, Var]*)⇒espressione

factor(*ListaI[, Var]*)⇒lista

factor(*MatriceI[, Var]*)⇒matrice

factor(*Espr1*) restituisce la scomposizione in fattori di *Espr1* per tutte le sue variabili con denominatore comune.

Espr1 viene scomposto in fattori razionali lineari (per quanto possibile) senza introdurre nuove sottoespressioni non reali. Questa procedura è utile qualora si desideri ottenere una scomposizione in fattori relativamente a più di una variabile.

factor(*Espr, Var*) restituisce la scomposizione in fattori di *Espr1* relativamente alla variabile *Var*.

Espr1 viene scomposto (per quanto possibile) in fattori reali che sono lineari in *Var*, anche se introduce costanti irrazionali o sottoespressioni irrazionali in altre variabili.

$$\begin{aligned} \text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a) \\ a \cdot (a-1) \cdot (a+1) \cdot (x-1) \cdot (x+1) \\ \hline \text{factor}(x^2 + 1) \\ x^2 + 1 \\ \hline \text{factor}(x^2 - 4) \\ (x-2) \cdot (x+2) \\ \hline \text{factor}(x^2 - 3) \\ x^2 - 3 \\ \hline \text{factor}(x^2 - a) \\ x^2 - a \end{aligned}$$

$$\begin{aligned} \text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a, x) \\ a \cdot (a^2 - 1) \cdot (x-1) \cdot (x+1) \\ \hline \text{factor}(x^2 - 3, x) \\ (x + \sqrt{3}) \cdot (x - \sqrt{3}) \\ \hline \text{factor}(x^2 - a, x) \\ (x + \sqrt{a}) \cdot (x - \sqrt{a}) \end{aligned}$$

I fattori ed i rispettivi termini vengono ordinati con *Var* come variabile principale. Le potenze simili di *Var* sono ridotte in ciascun fattore. Includere *Var* se si desidera che la scomposizione in fattori tenga conto solo di tale variabile e che le espressioni irrazionali siano incluse in qualsiasi altra variabile per aumentare la scomposizione in fattori relativamente a *Var*. Si può verificare una scomposizione in fattori incidentale relativamente ad altre variabili.

Nell'impostazione Auto della modalità **Auto/Approssimato**, l'inclusione di *Var* permette inoltre l'approssimazione con coefficienti a virgola mobile nel caso in cui i coefficienti irrazionali non possano essere esplicitamente espressi in termini concisi con le funzioni incorporate. Anche qualora vi sia una sola variabile, se si include *Var* la scomposizione in fattori può risultare più completa.

Nota: vedere anche **comDenom()** per ottenere una veloce scomposizione in fattori parziale quando **factor()** non è sufficientemente veloce o la memoria è esaurita.

Nota: vedere anche **cFactor()** per una scomposizione completa dei coefficienti complessi nel tentativo di ottenere fattori lineari.

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.

Per arrestare manualmente un calcolo:

- **Palmare:** Tenere premuto il tasto 

factor($x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3$)	$x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3$
factor($x^5+4 \cdot x^4+5 \cdot x^3-6 \cdot x-3, x$)	$(x-0.964673) \cdot (x+0.611649) \cdot (x+2.12543) \cdot (x^2+3.12543 \cdot x+3.12543)$

factor(152417172689)	123457 · 1234577
isPrime(152417172689)	false

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
(
estremoInf
,estremoSup,glNumer,glDenom)⇒numero
se estremoInf e estremoSup sono numeri,
lista se estremoInf e estremoSup sono liste
```

```
Fcdf
(
estremoInf
,estremoSup,glNumer,glDenom)⇒numero
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 *Espr*, *varMatrice*⇒*matrice*

Sostituisce ciascun elemento della variabile *varMatrice* con *Espr*.

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

Fill (Riempì)

Catalogo > 

Fill *Espr, varLista*⇒*lista*

Sostituisce ciascun elemento della variabile *varLista* con *Espr*.

varLista deve esistere già.

$\{1,2,3,4,5\} \rightarrow alist$

$\{1,2,3,4,5\}$

Fill 1.01,alist

Done

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 189).

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 254

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 >

floor(*Espr1*)⇒*intero*

floor(-2.14)

-3.

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

L'argomento può essere un numero reale o complesso.

floor(*List1*)⇒*lista*

floor($\left\{ \frac{3}{2}, 0, -5.3 \right\}$) {1, 0, -6.}

floor(*Matrice1*)⇒*matrice*

floor($\begin{bmatrix} 1.2 & 3.4 \\ 2.5 & 4.8 \end{bmatrix}$) [1. 3. | 2. 4.]

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

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

fMax() (Massimo funzione)

Catalogo >

fMax(*Espr, Var*)⇒*Espressione booleana*

fMax($1-(x-a)^2-(x-b)^2, x$) $x = \frac{a+b}{2}$

fMax(*Espr, Var, estremoInferiore*)

fMax($0.5 \cdot x^3 - x - 2, x$) $x = \infty$

fMax(*Espr, Var, estremoInferiore, estremoSuperiore*)

fMax(*Espr, Var*) | *estremoInferiore*≤*Var*≤*estremoSuperiore*

Restituisce un'espressione booleana che specifica i possibili valori di *Var* che massimizzano *Espr* o che individuano il suo estremo superiore.

È possibile utilizzare l'operatore ("|") di vincolo che limita l'intervallo di soluzione e/o specifica altri vincoli.

fMax($0.5 \cdot x^3 - x - 2, x$)| $x \leq 1$ $x = -0.816497$

Quando la modalità **Auto o Approssimato** è impostata su Approssimato, **fMax()** ricerca ripetutamente un massimo locale approssimato. Questo procedimento è generalmente più veloce, soprattutto se si utilizza l'operatore "|" per limitare la ricerca ad un intervallo relativamente piccolo che contiene esattamente un massimo locale.

Nota: vedere anche **fMin()** e **max()**.

fMin() (Minimo funzione)

Catalogo >

fMin(Espr, Var)⇒Espressione booleana

fMin(Espr, Var,estremoInferiore)

fMin(Espr, Var,estremoInferiore,estremoSuperiore)

fMin(Espr, Var) | estremoInferiore≤Var ≤estremoSuperiore

Restituisce un'espressione booleana che specifica i possibili valori di *Var* che minimizzano *Espr* o che individuano il suo estremo inferiore.

È possibile utilizzare l'operatore ("|") di vincolo che limita l'intervallo di soluzione e/o specifica altri vincoli.

Quando la modalità **Auto o Approssimato** è impostata su Approssimato, **fMin()** ricerca ripetutamente un minimo locale approssimato. Questo procedimento è generalmente più veloce, soprattutto se si utilizza l'operatore "|" per limitare la ricerca ad un intervallo relativamente piccolo che contiene esattamente un minimo locale.

Nota: vedere anche **fMax()** e **min()**.

$fMin(1-(x-a)^2-(x-b)^2, x)$	$x = -\infty \text{ or } x = \infty$
$fMin(0.5 \cdot x^3 - x - 2, x) x \geq 1$	$x = 1$

For

Catalogo >

For Var, Bass, Alto [, Incr]

Done

Bass

EndFor

Esegue iterativamente le istruzioni di *Bass* per ciascun valore di *Var*, da *Bass* 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.

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

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

Local *tempsum, step, i*

$0 \rightarrow \text{tempsum}$

$1 \rightarrow \text{step}$

For *i, 1, 100, step*

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

EndFor

EndFunc

g()

5050

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.

format() (Formato)

format(*Espr*[, *stringaFormato*])⇒*stringa*

Restituisce *Espr* come stringa di caratteri basata sul modello di formato.

Espr deve poter essere semplificata in un numero.

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.

G[n][c]: analogo al formato fisso, separa inoltre le cifre a sinistra del separatore decimale 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.

format(1.234567,"f3")	"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"

fPart() Funzione parte frazionaria**Catalogo > ****fPart(*EsprI*)** \Rightarrow espressione

fPart(-1.234) -0.234

fPart(*Listal*) \Rightarrow lista

fPart({1, 2, 3, 7.003}) {0, -0.3, 0.003}

fPart(*MatriceI*) \Rightarrow 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.

F Pdf() (Probabilità di distribuzione F)**Catalogo > ****F Pdf(*valX,glNumer,glDenom*)** \Rightarrow numero se *ValX* è un numero, lista se *ValX* è una lista**F Pdf(*valX,glNumer,glDenom*)** \Rightarrow numero se *ValX* è un numero, lista se *ValX* è una listaCalcola la probabilità di distribuzione F in *valX* per il *glNumer* (gradi di libertà) e *glDenom* specificati.**freqTable►list()****Catalogo > ****freqTable►list(*Listal,Lista
ListaInteriFreq*)** \Rightarrow listafreqTable►list({1,2,3,4},{1,4,3,1})
{1,2,2,2,2,3,3,3,4}Restituisce una lista contenente gli elementi di *Listal* espansi secondo le frequenze in *ListaInteriFreq*. Questa funzione può essere utilizzata per costruire una tabella di frequenze per l'applicazione Dati e statistiche.freqTable►list({1,2,3,4},{1,4,0,1})
{1,2,2,2,2,4}*Listal* può essere qualsiasi lista valida.*ListaInteriFreq* deve avere la stessa dimensione di *Listal* e deve contenere solo elementi interi non negativi. Ciascun elemento specifica il numero di volte che l'elemento corrispondente di *Listal* verrà ripetuto nella lista dei risultati. Un valore zero esclude l'elemento corrispondente di *Listal*.

freqTable@list()

Catalogo > 

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 254.

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

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, $? \leq b(1)$), countIf(list, $b(1) < ? \leq b(2)$), ..., countIf(list, $b(n-1) < ? \leq b(n)$), 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 254.

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

Nota: vedere anche **countIf()**, pagina 42.

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.

FTest_2Samp (Verifica F su due campioni)

Catalogo > 

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

FTest_2Samp (Verifica F su due campioni)

Catalogo >

FTest_2Samp *List1,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 189).

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 254.

Variabile di output	Descrizione
stat.F	Statistica U 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 = n1-1
stat.dfDenom	Gradi di libertà del denominatore = n2-1
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

Catalogo >

Func
Blocco

Definizione di una funzione Piecewise (definita a tratti):

EndFunc

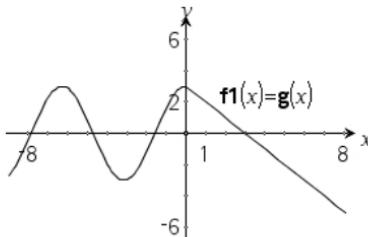
Modello per la creazione di una funzione definita dall'utente.

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.

Define $g(x)=\text{Func}$ *Done*

```
If  $x < 0$  Then
  Return  $3 \cdot \cos(x)$ 
Else
  Return  $3 - x$ 
EndIf
EndFunc
```

Risultato della rappresentazione grafica $g(x)$ 

G

gcd() (Massimo comun divisore)

Catalogo >

 $\text{gcd}(\text{Numero1}, \text{Numero2}) \Rightarrow \text{espressione}$ $\text{gcd}(18,33)$

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.

 $\text{gcd}(\text{Lista1}, \text{Lista2}) \Rightarrow \text{lista}$ $\text{gcd}(\{12,14,16\}, \{9,7,5\})$

{3,7,1}

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

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

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

geomCdf() (Probabilità geometrica cumulata)

Catalogo >

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() (Probabilità per la distribuzione geometrica discreta)

Catalogo >

geomPdf(*p*,*ValX*) \Rightarrow 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

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 ..."**.

Menu Hub

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

— 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 implicita. 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.

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 91 e **Send**, pagina 168.

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

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

getDenom() (Ottieni/restituisce denominatore)

Catalogo >

getDenom(*Espr1*) \Rightarrow *espressione*

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

getDenom $\left(\frac{x+2}{y-3}\right)$	$y-3$
getDenom $\left(\frac{2}{7}\right)$	7
getDenom $\left(\frac{1+y^2+y}{x+y^2}\right)$	$x \cdot y$

getLangInfo() (Ottieni informazioni sulla lingua)

Catalog > 

getLangInfo()⇒*stringa*

getLangInfo()

"en"

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"

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 113 e **unLock**, pagina 212.

<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

getMode(interoNomeModo)⇒valore

getMode(0)⇒lista

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

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 172.

getMode(0)	
	{1,7,2,1,3,1,4,1,5,1,6,1,7,1,8,1}
getMode(1)	7
getMode(8)	1

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=Radiani, 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
Auto o Approssimato (Auto or Approx.)	5	1=Auto, 2=Approssimato, 3=Esatto

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/restituisce numeratore)

Catalogo >

getNum(*Espr1*)⇒*espressione*

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

$$\begin{aligned} \text{getNum}\left(\frac{x+2}{y-3}\right) &= x+2 \\ \text{getNum}\left(\frac{2}{7}\right) &= 2 \\ \text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right) &= x+y \end{aligned}$$

GetStr

Menu Hub

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

Per degli esempi vedere **Get**.

GetStr[*promptString*,] *funk*(*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 87 e **Send**, pagina 168.

getType()

Catalogo >

getType(*var*)⇒*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}→ <i>temp</i>	{1,2,3}
getType(<i>temp</i>)	"LIST"
3· <i>i</i> → <i>temp</i>	3· <i>i</i>
getType(<i>temp</i>)	"EXPR"
DelVar <i>temp</i>	<i>Done</i>
getType(<i>temp</i>)	"NONE"

getVarInfo() (Ottieni informazioni variabile)

Catalog >

getVarInfo()⇒matrice o stringa

getVarInfo

(*StringaNomeLibreria*)⇒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()**.

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.

getVarInfo()	"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>
getVarInfo()	$\begin{bmatrix} x & \text{"NUM"} & \left[\begin{smallmatrix} \end{smallmatrix} \right] & 1 \\ y & \text{"LIST"} & \text{"LibPriv"} & 0 \\ z & \text{"FUNC"} & \text{"LibPub"} & 0 \end{bmatrix}$
getVarInfo("tmp3")	"Error: Argument must be a string"
getVarInfo("tmp3")	[volcyl2 "NONE" "LibPub" 0]

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

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 (Gradianti o Gradi centesimali)**Catalogo >** **Espr1 ► Grad⇒espressione**

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

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

In modalità angolo in gradi:

(1.5)►Grad(1.66667)^g

In modalità angolo in radianti:

(1.5)►Grad(95.493)^r

I

identity() (Identità)**Catalogo >** **identity(*Intero*)⇒matrice**

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

Intero deve essere un numero intero positivo.

identity(4)

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

If

If *Espressione booleana*
Istruzione

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

Done

4

If *Espressione booleana Then*
Blocco
EndIf

 $g(-2)$

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

Se il valore di *Espressione booleana* 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 *Espressione booleana Then*
Blocco1
Else
Blocco2
EndIf

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

Done

12

 $g(12)$

12

 $g(-12)$

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

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

Blocco1 e *Blocco2* possono essere una singola istruzione.

If

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

Permette il passaggio a un'altra funzione. Se il valore di *Espressione booleana1* è vero, viene eseguito *Blocco1*. Se il valore di *Espressione booleana1* è falso, viene calcolata *Espressione booleana2*, ecc.

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 corrisponde a qualsiasi posizione selezionata.

ifFn()**Catalogo >**

- Se il secondo, terzo o quarto argomento della funzione **ifFn()** 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.

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

Valore_se_falso non è specificato. Viene utilizzato *Undef*.

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

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

imag() (Parte immaginaria)**Catalogo >**

imag(*EsprI*) \Rightarrow *espressione*

Restituisce la parte immaginaria dell'argomento.

$$\text{imag}(1+2 \cdot i) \quad 2$$

$$\text{imag}(z) \quad 0$$

$$\text{imag}(x+i \cdot y) \quad y$$

Nota: tutte le variabili non definite vengono considerate come variabili reali. vedere anche **real()**, pagina 155

imag(*Listal*) \Rightarrow *lista*

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

Restituisce una lista delle parti immaginarie degli elementi.

imag(*MatriceI*) \Rightarrow *matrice*

Restituisce una matrice delle parti immaginarie degli elementi.

$$\text{imag} \begin{bmatrix} a & b \\ i \cdot c & i \cdot d \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$$

impDif() (Derivata implicita)**Catalogo >**

impDif(*Equazione*, *Var*, *varDipendente* [, *Ord*]) \Rightarrow *espressione*

$$\text{impDif}(x^2+y^2=100, x, y) \quad \frac{-x}{y}$$

dove l'impostazione predefinita dell'ordine *Ord* è 1.

Calcola la derivata implicita di equazioni in cui una variabile è definita implicitamente rispetto a un'altra.

inString() (All'interno della stringa)**Catalogo >**

inString(stringaRicerca, sottoStringa[, Inizio])⇒intero

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.

inString("Hello there","the")	7
inString("ABCEFG","D")	0

int() (Funzione parte intera)**Catalogo >**

int(Espr)⇒intero

int(-2.5)	-3.
-----------	-----

int(Lista1)⇒lista

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

int(Matrice1)⇒matrice

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() Divisione intera**Catalogo >**

intDiv(Numer01, Numer02)⇒intero

intDiv(-7,2)	-3
--------------	----

intDiv(Lista1, Lista2)⇒lista

intDiv(4,5)	0
-------------	---

intDiv(Matrice1, Matrice2)⇒matrice

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

Restituisce la parte intera di ($\text{Numero1} \div \text{Numero2}$).

Con liste e matrici, restituisce la parte intera di (argomento 1 \div argomento 2) per ciascuna coppia di elementi.

integral

Vedere παγ α ◆◆◆-

interpolate ()

interpolate(valoreX, listaX, listaY, listaYprime)⇒lista

Questa funzione consente di fare quanto segue:

Dati listaX , $\text{listaY}=\mathbf{f}(\text{listaX})$ e $\text{listaYprime}=\mathbf{f}'(\text{listaX})$ per una funzione \mathbf{f} non nota, viene utilizzato una interpolante cubica per approssimare la funzione \mathbf{f} al valoreX . 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 $[\text{listaX}[i], \text{listaX}[i+1]]$ che contenga valoreX . Se trova un intervallo di questo tipo, restituisce un valore interpolato per $\mathbf{f}(\text{valoreX})$; altrimenti, restituisce **undef**.

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

valoreX può essere una variabile indefinita, un numero o una lista di numeri.

Equazione differenziale:

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

```
rk:=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  10.
```

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

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.,2.67062,3.19499,4.02782,5.00394,6.00011}
```

invχ2()

invχ2(Area,gl)

invChi2(Area,gl)

invχ²()**Catalogo > **

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

invF() (Distribuzione F cumulativa inversa)**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.

invNorm() (Distribuzione normale cumulativa inversa)**Catalogo > ****invNorm(*area,[μ,σ]*)**

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 t di Student inversa, specificata dal grado di libertà, gl per una data *Area* sotto la curva.

iPart() (Parte intera)**Catalogo > ****iPart(*Numero*) \Rightarrow *intero*****iPart(*-1.234*)****-1.****iPart(*ListaL*) \Rightarrow *lista*****iPart($\left\{ \frac{3}{2}, -2.3, 7.003 \right\}$)****{1, -2, 7.}****iPart(*MatriceL*) \Rightarrow *matrice***

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()

irr(*CF0,CFLista [,CFFreq]*)⇒*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*.

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 **mirr()**, pagina 123.

<i>list1</i> := {6000,-8000,2000,-3000}	{6000, -8000, 2000, -3000}
<i>list2</i> := {2,2,2,1}	{2,2,2,1}
irr(5000, <i>list1</i> , <i>list2</i>)	-4.64484

isPrime() (Numero primo)

isPrime(*Numero*)⇒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.

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

isPrime(5)	true
isPrime(6)	false

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

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

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.

isVoid()

isVoid(Var)⇒espressione costante booleana

a:=_	-
isVoid(a)	true
isVoid({1,_,3})	{ false,true,false }

isVoid(Espr)⇒espressione costante booleana

isVoid(Lista)⇒lista di espressioni costanti booleane

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

Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 254.

L**Lbl (Etichetta)**

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

lcm() (Minimo comune multiplo)**Catalogo >** **lcm(*Numero1*, *Numero2*)** \Rightarrow espressione**lcm(*List1*, *Lista2*)** \Rightarrow lista**lcm(*Matrice1*, *Matrice2*)** \Rightarrow 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 divisore (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 $\left\{ \left[\frac{1}{3}, -14, 16 \right], \left[\frac{2}{15}, 7, 5 \right] \right\}$ { $\frac{2}{3}, 14, 80$ }**left() (Sinistra)****Catalogo >** **left(*stringaOrigine*[, *Num*])** \Rightarrow 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*]) \Rightarrow 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*) \Rightarrow espressioneleft($x < 3$)

x

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

libShortcut() (Collegamento a libreria)**Catalogo >** **libShortcut(*stringaNomeLibr*,
stringaNomeCollegamento[,
LibPrivFlag])** \Rightarrow lista di variabili

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

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 35.

Per eliminare un gruppo di variabili, vedere **DelVar**, pagina 57.

getVarInfo("linalg2")

[clearmat	"FUNC"	"LibPub "
gauss1	"PRGM"	"LibPriv "	
gauss2	"FUNC"	"LibPub "	

libShortcut("linalg2","la")

{la.clearmat,la.gauss2}

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

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

limit() o lim() (Limite)

**limit(*Espr1*, *Var*, *Punto* [,
Direzione])**⇒*espressione*

**limit(*Listal*, *Var*, *Punto* [,
Direzione])**⇒*lista*

**limit(*Matrice1*, *Var*, *Punto* [,
Direzione])**⇒*matrice*

Restituisce il limite richiesto.

Nota: vedere anche **Modello di limite**, pagina 11.

Direzione: negativa=da sinistra, positiva=da destra, altrimenti=entrambe. (se il valore è omesso, *Direzione* viene impostata per default su entrambe).

I limiti tendenti a $+\infty$ e a $-\infty$ sono sempre convertiti in limiti unilaterali dalla parte finita.

$$\lim_{x \rightarrow 5} (2 \cdot x + 3)$$

13

$$\lim_{x \rightarrow 0^+} \left(\frac{1}{x} \right)$$

∞

$$\lim_{x \rightarrow 0} \left(\frac{\sin(x)}{x} \right)$$

1

$$\lim_{h \rightarrow 0} \left(\frac{\sin(x+h) - \sin(x)}{h} \right)$$

$\cos(x)$

$$\lim_{n \rightarrow \infty} \left(\left(1 + \frac{1}{n} \right)^n \right)$$

e

A seconda dei casi, **limit()** restituisce il limite cercato oppure `undef` quando non può determinare un limite univoco. Ciò non significa necessariamente che non esista un limite univoco. `undef` significa che il risultato è un numero non conosciuto di grandezza finita o infinita, oppure è l'intero insieme di tali numeri.

limit() utilizza metodi quali la regola de L'Hopital, per cui vi sono limiti univoci che non possono essere determinati da questa funzione. Se *Expr1* contiene variabili indefinite diverse da *Var*, può risultare necessario ridurle per ottenere un risultato più conciso.

I limiti sono molto sensibili agli errori di arrotondamento. Pertanto, si consiglia di evitare nel calcolo dei limiti, per quanto possibile, l'impostazione Approssimato della modalità **Auto** o **Approssimato** e i numeri approssimati. In caso contrario, è possibile che i numeri, che dovrebbero essere zero o avere una grandezza infinita, si comportino in modo diverso e che i limiti, che dovrebbero avere una grandezza finita diversa da zero, non l'abbiano.

$\lim_{x \rightarrow \infty} (a^x)$	<code>undef</code>
$\lim_{x \rightarrow \infty} (a^x) a > 1$	∞
$\lim_{x \rightarrow \infty} (a^x) a > 0 \text{ and } a < 1$	0

LinRegBx (Regressione lineare)

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 189).

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 254.

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 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>

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 189).

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 254.

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 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>

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 189).

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 254.

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
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.ŷ	$a + b \cdot \text{ValX}$

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 ρ per l'equazione $y=\alpha+\beta x$. Viene verificata l'ipotesi nulla $H_0: \beta=0$ (in modo equivalente, $\rho=0$) 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 ($H_0: \beta=\rho=0$).

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

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

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

Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

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**

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

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

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

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

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

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

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

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”.

Δlist() (Differenza in una lista)

Δlist(*Listal*)⇒*lista*

$\Delta\text{List}(\{20,30,45,70\})$

{ 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 *Listal*. Ogni elemento di *Listal* viene sottratto dal successivo elemento di *Listal*. La lista risultante è sempre composta da un elemento in meno della *Listal* originale.

list►mat() (Da lista a matrice)

**list►mat(*Lista* [,
elementiPerRiga])**⇒*matrice*

$\text{list}\blacktriangleright\text{mat}(\{1,2,3\})$	[1 2 3]
$\text{list}\blacktriangleright\text{mat}(\{1,2,3,4,5\},2)$	[1 2 3 4 5 0]

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(...)**.

►In (Conversione in logaritmo naturale)

Catalogo >

Espr ►In⇒espressione

Causa la conversione dell'input *Espr* in un'espressione contenente solo logaritmi naturali (ln).

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

$$\left(\log_{10}(x) \right) \blacktriangleright \ln \frac{\ln(x)}{\ln(10)}$$

ln() (Logaritmo naturale)

Tasti

ln(*Espr1*)⇒espressione

$$\ln(2.) \quad 0.693147$$

ln(*Lista1*)⇒lista

Restituisce il logaritmo naturale dell'argomento.

In una lista, restituisce i logaritmi naturali degli elementi.

Se la modalità del formato complesso è Reale:

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

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:

$$\ln(\{-3,1,2,5\}) \quad \{ \ln(3)+\pi \cdot i, 0.182322, \ln(5) \}$$

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

$$\begin{aligned} \ln \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \\ \begin{matrix} 1.83145+1.73485 \cdot i & 0.009193-1.49086 \\ 0.448761-0.725533 \cdot i & 1.06491+0.623491 \\ -0.266891-2.08316 \cdot i & 1.12436+1.79018 \end{matrix} \end{aligned}$$

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

LnReg (Regressione logaritmica)

Catalogo >

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 189).

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 254.

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>
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>

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

rollcount()

16

rollcount()

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*.

Nota: Il comando **Blocca (Lock)** cancella la cronologia di Annulla/Ripeti quando è applicato a variabili sbloccate.

Vedere **unLock**, pagina 212 e **getLockInfo()**, pagina 89.

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

log() (Logaritmo)

Tasti **ctrl** **10^x**

log(*Espr1*[,*Espr2*]) \Rightarrow espressione

log(*Listal*[,*Espr2*]) \Rightarrow lista

Restituisce il logaritmo in base-*Espr2* dell'argomento.

Nota: vedere anche **Modello di logaritmo**, pagina 6.

In una lista, restituisce il logaritmo in base-*Espr2* degli elementi.

Se *Espr2* viene omesso, come base viene utilizzato 10.

$\log_{10}(2.)$	0.30103
$\log_4(2.)$	0.5
$\log_3(10) - \log_3(5)$	$\log_3(2)$

Se la modalità del formato complesso è Reale:

$$\log_{10}(\{-3,1,2,5\}) \quad \text{Error: Non-real result}$$

Se la modalità del formato complesso è Rettangolare:

$$\begin{aligned} \log_{10}(\{-3,1,2,5\}) \\ \left\{ \log_{10}(3) + 1.36438 \cdot i, 0.079181, \log_{10}(5) \right\} \end{aligned}$$

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

$$\begin{aligned} \log_{10} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \\ \begin{bmatrix} 0.795387 + 0.753438 \cdot i & 0.003993 - 0.6474 \cdot i \\ 0.194895 - 0.315095 \cdot i & 0.462485 + 0.2707 \cdot i \\ -0.115909 - 0.904706 \cdot i & 0.488304 + 0.7774 \cdot i \end{bmatrix} \end{aligned}$$

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

logbase(*matriceQuadrata1*[,*Espr2*]) \Rightarrow matriceQuadrata

Restituisce il logaritmo in base-*Espr2* della matrice di *matriceQuadrata1*. Ciò non equivale a calcolare il logaritmo in base-*Espr2* di ogni elemento. Per informazioni sul metodo di calcolo, vedere **cos()**.

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

Se l'argomento base viene omesso, come base viene utilizzato 10.

Catalogo >

Espr **logbase**(*Espr1*) \Rightarrow espressione

Causa la semplificazione dell'input *Espr* in un'espressione che utilizza la base *Espr1*.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando **@>logbase (...)**.

$$\frac{\log_3(10) - \log_5(5) \blacktriangleright \logbase(5)}{\log_5\left(\frac{10}{3}\right)} = \log_3\left(\frac{10}{3}\right)$$

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

Calcola la regressione logistica $y = c/(1+a \cdot e^{-bx})$ sulle liste *X* e *Y* con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189).

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 254.

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 189).

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 254.

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

Variabile di output	Descrizione
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>

Loop

Catalogo >

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)

Catalogo >

**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*.

MatriceL · *MatriceU* = *MatriceP* · *matrice*

$$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$$

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

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 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.

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

$\begin{bmatrix} m & n \\ 0 & p \end{bmatrix} \rightarrow mI$	$\begin{bmatrix} m & n \\ 0 & p \end{bmatrix}$
LU <i>mI,lower,upper,perm</i>	Done
lower	$\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \\ o & \end{bmatrix}$
upper	$\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$
perm	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

max() (Massimo)

max(*Espr1, Espr2*)⇒*espressione*

matlist([1 2 3])	{1,2,3}
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow mI$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
matlist(<i>mI</i>)	{1,2,3,4,5,6}

max(*List1, Lista2*)⇒*lista*

max(*Matrice1, Matrice2*)⇒*matrice*

max(2,3,1,4)	2,3
max({1,2},{-4,3})	{1,3}

max() (Massimo)

Catalogo >

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(Lista)⇒espressione

Restituisce l'elemento massimo di *lista*.

max(Matrice I)⇒matrice

Restituisce un vettore riga contenente l'elemento massimo di ciascuna colonna di *Matrice I*.

Gli elementi vuoti (nulli) vengono ignorati.
Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 254.

Nota: vedere anche **fMax()** e **min()**.

max({0,1,-7,1.3,0.5})

1.3

max[[1 -3 7], [-4 0 0.3]]

[1 0 7]

mean() Media

Catalogo >

mean(Lista[, listaFreq])⇒espressione

Restituisce la media degli elementi di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

mean(Matrice I[, matriceFreq])⇒matrice

Restituisce un vettore riga contenente le medie di tutte le colonne di *Matrice I*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Matrice I*.

Gli elementi vuoti (nulli) vengono ignorati.
Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 254.

mean({0.2,0.1,-0.3,0.4})

0.26

mean({1,2,3},{3,2,1})

5

3

In modalità formato vettore rettangolare:

mean[[0.2 0], [-1 3], [0.4 -0.5]] [-0.133333 0.833333]

mean[[1 0], [5 3], [2 -1], [5 2]] [-2 5] [15 6]

mean[[1 2][5 3], [3 4][4 1], [5 6][6 2]] [47 11] [15 3]

median() (Mediana)

Catalogo >

median(Lista[, listaFreq])⇒espressione

median({0.2,0.1,-0.3,0.4})

0.2

Restituisce la mediana degli elementi di *Lista*.

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

median(MatriceI[, matriceFreq])⇒matrice
$$\text{median} \begin{bmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{bmatrix} \quad [0.4 \quad -0.3]$$

Restituisce un vettore riga contenente le mediane degli elementi delle colonne di *MatriceI*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *MatriceI*.

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 254.

MedMed (Linea mediana-medianay)

Catalogo >

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 189).

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 254.

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 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>

mid() (In mezzo alla stringa)

**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*.

mid("Hello there",2)	"ello there"
mid("Hello there",7,3)	"the"
mid("Hello there",1,5)	"Hello"
mid("Hello there",1,0)	"[]"

mid() (In mezzo alla stringa)

Catalogo >

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*

Restituisce *Cont* stringhe della lista delle stringhe *listaStringheOrigine*, incominciano dal numero di elemento di *Inizio*.

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"}

min() (Minimo)

Catalogo >

min(*Espr1*, *Espr2*)⇒*espressione*

min(2,3,1,4)	1.4
min({1,2},{-4,3})	{-4,2}

min(*List1*, *List2*)⇒*lista*

min(*Matrice1*, *Matrice2*)⇒*matrice*

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({0,1,-7,1,3,0.5})	-7

min(*Matrice1*)⇒*matrice*

min([[1 -3 7], [-4 0 0.3]])	[-4 -3 0.3]

Restituisce un vettore di riga contenente l'elemento minimo di ciascuna colonna di *Matrice1*.

Nota: vedere anche **fMin()** e **max()**.

mirr()**Catalogo >**

```
mirr
(
tassoFinanziamento
,tassoReinvestimento,CF0,CFLista
,[CFFreq])
```

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.

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 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 100.

<i>list1</i> := { 6000,-8000,2000,-3000 }	{ 6000,-8000,2000,-3000 }
<i>list2</i> := { 2,2,2,1 }	{ 2,2,2,1 }
mirr(4.65,12,5000, <i>list1</i> , <i>list2</i>)	13.41608607

mod() (Modulo)**Catalogo >**

mod(*Espr1*, *Espr2*)⇒espressione

mod(*Lista1*, *Lista2*)⇒lista

mod(*Matrice1*, *Matrice2*)⇒matrice

Restituisce il primo argomento modulo secondo argomento, come definito dalle identità:

$$\text{mod}(x,0) = x$$

$$\text{mod}(x,y) = x - y \text{ floor}(x/y)$$

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 }

mod() (Modulo)

Catalogo >

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 158

mRow() (Operazione con righe di matrice)

Catalogo >

mRow(*Espr, Matrice1, Indice*) \Rightarrow matrice

Restituisce una copia di *Matrice1* dove ciascun elemento della riga *Indice* di *Matrice1* viene moltiplicato per *Espr*.

$$\text{mRow}\left(-1, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 2\right) \quad \begin{bmatrix} 1 & 2 \\ -1 & 4 \\ 3 & 3 \end{bmatrix}$$

mRowAdd() (Moltiplicazione e somma di righe di matrice)

Catalogo >

mRowAdd(*Espr, Matrice1, Indice1, Indice2*) \Rightarrow matrice

Restituisce una copia di *Matrice1* dove ciascun elemento della riga *Indice2* di *Matrice1* viene sostituito con:

$$\text{mRowAdd}\left(-3, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}$$

$$\text{mRowAdd}\left(n, \begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} a & b \\ a \cdot n + c & b \cdot n + d \end{bmatrix}$$

Espr \times riga *Indice1* + riga *Indice2*

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 189).

Tutte le liste devono avere la stessa dimensione.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.b0, stat.b1, ...	Coefficienti di regressione
stat.R ²	Coefficiente di determinazione multipla
stat.yList	$\hat{y} \text{ List} = b_0+b_1 \cdot x_1+ \dots$
stat.Resid	Residui della regressione

MultRegIntervals (Intervalli di confidenza della previsione di regressione multipla)

MultRegIntervals $Y, X1[, X2[, X3, \dots, X10]], listaValX[, livelloConfidenza]$

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 189).

Tutte le liste devono avere la stessa dimensione.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.y	Una stima del punto: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ 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

Variabile di output	Descrizione
stat.LowerPred,	Intervallo di previsione per una singola osservazione
stat.UpperrPred	
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 189).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

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 libertà della regressione

Variabile di output	Descrizione
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,...} 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
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

ctrl = 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 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.

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)

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}

nCr() (Combinazioni)

Catalogo >

nCr(Espr1, Espr2) \Rightarrow espressione

Date le espressioni intere *Espr1* e *Espr2* con $Espr1 \geq Espr2 \geq 0$, **nCr()** è il numero di combinazioni degli elementi di *Espr1* presi nel numero di *Espr2* per volta (questa procedura è nota anche come coefficiente binomiale). Entrambi gli argomenti possono essere numeri interi o espressioni simboliche.

nCr(z,3)	$\frac{z \cdot (z-1) \cdot (z-2)}{6}$
Ans z=5	10
nCr(z,c)	$\frac{z!}{c! \cdot (z-c)!}$
Ans	$\frac{1}{c!}$

nCr(Espr1, 0) \Rightarrow 1

nCr(Espr1, interoNeg) \Rightarrow 0

nCr(Espr1, interoPos) \Rightarrow Espr1 · (Espr1-1)...(Espr1-interoPos+1)/interoPos!

nCr(Espr1, nonInter) \Rightarrow espressione! / ((Espr1-nonInter)!) · nonInter!)

nCr(Lista1, Lista2) \Rightarrow lista

nCr({5,4,3},{2,4,2})	{10,1,3}
----------------------	----------

nCr() (Combinazioni)

Catalogo >

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*

Restituisce una matrice di combinazioni, sulla base delle coppie di elementi corrispondenti nelle due matrici. Le matrici degli argomenti devono avere le stesse dimensioni.

$$\text{nCr}\left[\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right] = \begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$$

nDerivative()

Catalogo >

nDerivative(*Espr1*, *Var*=*Valore*, [*Ordine*])⇒*valore*

nDerivative(*Espr1*, *Var*[,*Ordine*]) | *Var*=*Valore*⇒*valore*

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.

nDerivative(x ,x=1)	1
nDerivative(x ,x) x=0	undef
nDerivative(√(x-1),x) x=1	undef

newList() (Nuova lista)

Catalogo >

newList(*numElementi*)⇒*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*)⇒*matrice*

newMat(2,3) [0 0 0
0 0 0]

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*)⇒*valore*

$$\text{nfMax}\left(-x^2 - 2 \cdot x - 1, x\right) \quad -1.$$

nfMax(*Espr*, *Var*, *estremoInf*)⇒*valore*

$$\text{nfMax}\left(0.5 \cdot x^3 - x - 2, x, -5, 5\right) \quad 5.$$

**nfMax(*Espr*, *Var*, *estremoInf*,
estremoSup)**⇒*valore*

**nfMax(*Espr*, *Var*) | *estremoInf*≤*Var*
≤*estremoSup***⇒*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.

Nota: vedere anche **fMax()** e **d()**.

nfMin() (Minimo di una funzione calcolato numericamente)

Catalogo >

nfMin(*Espr*, *Var*)⇒*valore*

$$\text{nfMin}\left(x^2 + 2 \cdot x + 5, x\right) \quad -1.$$

nfMin(*Espr*, *Var*, *estremoInf*)⇒*valore*

$$\text{nfMin}\left(0.5 \cdot x^3 - x - 2, x, -5, 5\right) \quad -5.$$

**nfMin(*Espr*, *Var*, *estremoInf*,
estremoSup)**⇒*valore*

**nfMin(*Espr*, *Var*) | *estremoInf*≤*Var*
≤*estremoSup***⇒*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.

Nota: vedere anche **fMin()** and **d()**.

nInt() (Integrale numerico)**Catalogo > ****nInt(*EsprI*, *Var*, *Inferiore*,
Superiore)** \Rightarrow espressione

nInt(e^{-x^2} , *x*, -1, 1)

1.49365

Se la funzione da integrare *EsprI* 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 (*EsprI*, *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.

Nota: vedere anche **ʃ()**, pagina 226.

$$\begin{aligned} \text{nInt}(\cos(x), x, -\pi, \pi + 1.e-12) &= -1.04144e-12 \\ \int_{\pi + 10^{-12}}^{\pi} \cos(x) dx &\approx -\sin\left(\frac{1}{1000000000000}\right) \end{aligned}$$

$$\begin{aligned} \text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x\right), x, 0, 1\right) &= 3.30423 \end{aligned}$$

nom()**Catalogo > ****nom(*tassoEffettivo*, *CpY*)** \Rightarrow valore

nom(5.90398, 12)

5.75

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 66.

BooleanExpr1norBooleanExpr2 restituisce
Boolean expression

$x \geq 3 \text{ or } x \geq 4$	$x \geq 3$
$x \geq 3 \text{ nor } x \geq 4$	$x < 3$

BooleanList1norBooleanList2 restituisce
Boolean list

BooleanMatrix1norBooleanMatrix2
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.

Integer1norInteger2⇒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)⇒espressione

$$\text{norm}\begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \sqrt{a^2+b^2+c^2+d^2}$$

norm(Vettore)⇒espressione

$$\text{norm}\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \sqrt{30}$$

Restituisce la norma di Frobenius.

$$\text{norm}\begin{bmatrix} 1 & 2 \end{bmatrix} \quad \sqrt{5}$$

$$\text{norm}\begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \sqrt{5}$$

normalLine() (Retta normale)**Catalogo >**

normalLine
 $(Espr1, Var, Punto) \Rightarrow espressione$

normalLine
 $(Espr1, Var=Punto) \Rightarrow espressione$

Restituisce la retta normale alla curva rappresentata da *Espr1* nel punto specificato in *Var=Punto*.

Accertarsi che la variabile indipendente non sia definita. Per esempio, se $f1(x):=5$ e $x:=3$, allora **normalLine**($f1(x), x, 2$) restituisce "false".

$\text{normalLine}(x^2, x, 1)$	$\frac{3}{2} - \frac{x}{2}$
$\text{normalLine}((x-3)^2-4, x, 3)$	$x=3$
$\left\{ \frac{1}{x^3}, x=0 \right\}$	0
$\text{normalLine}(\sqrt{ x }, x=0)$	undefined

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.

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

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 *EsprBooleana* \Rightarrow espressione booleana

$\text{not}\{2 \geq 3\}$	true
$\text{not}\{x < 2\}$	$x \geq 2$
not not <i>innocent</i>	<i>innocent</i>

not

Restituisce vero o falso o una forma semplificata dell'argomento.

not *Interol* \Rightarrow *intero*

Restituisce il complemento a uno di un intero reale. Internamente, *Interol* 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 22.

In modalità base Esadecimale:

Importante: è zero, non la lettera O.

not 0h7AC36 0hFFFFFFFFFFFF853C9

In modalità base Bin:

0b100101 ► Base10

not 0b100101

not 0b100101►Base10

-38

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 >

nPr(Espr1, Espr2)⇒espressione

Per $Espr1$ ed $Espr2$ intere, dove $Espr1 \geq Espr2 \geq 0$, $nPr()$ corrisponde al numero di disposizioni semplici degli elementi di $Espr1$ prese nel numero di $Espr2$ alla volta. Entrambi gli argomenti possono essere numeri interi o espressioni simboliche.

nPr(Espr, 0)⇒1

nPr(Espresso, interoNeg) \Rightarrow 1 / ((Espresso + 1) · (Espresso + 2) · ... · (Espresso - interoNeg))

$$\mathbf{nPr}(Espr, interoPos) \Rightarrow Espr \cdot (Espr-1) \cdots (Espr - interoPos + 1)$$

nPr(*Expr*, *nonIntero*) \Rightarrow *Expr*! / (*Expr* - *nonIntero*)!

$nPr(z, 3)$	$z \cdot (z-1) \cdot (z-2)$
$Ans z=5$	60
$nPr(z, 3)$	$\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$
$nPr(z, c)$	$\frac{z!}{(z-c)!}$
$Ans \cdot nPr(z-c, -c)$	1

$|Valore2 \geq 0$, **nPr()** corrisponde al numero di disposizioni semplici di *Valore1* prese nel numero di *Valore2* alla volta.

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.

$\text{nPr}\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$
--	---

npv()

npv(tassoInteresse, CFO, CFLista [,CFFreq])

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

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.

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

tassoInteresse è il tasso a cui scontare i cash flow (il costo del denaro) di un dato periodo.

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

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

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

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

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

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.

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

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

nSolve() (Soluzione numerica)

Catalogo >

**nSolve(Equazione,Var
[=Campione])** ⇒ numero o stringa_erroro

**nSolve(Equazione,Var
[=Campione],estremoInf,estremoSup)** ⇒ numero o stringa_erroro

**nSolve(Equazione,Var[=Campione]) |
estremoInf≤Var≤estremoSup** ⇒ numero o stringa_erroro

Ricerca iterativamente una soluzione numerica reale approssimata dell'*Equazione* per la sua variabile. Specificare la variabile come:

variabile

– 0 –

variabile = numero reale

Ad esempio, x è valido come pure x=3.

nSolve() risulta spesso più veloce di **solve()** o di **zeros()**, soprattutto se viene utilizzato l'operatore “|” per limitare la ricerca in un intervallo piccolo che contenga esattamente una soluzione semplice.

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.”

Nota: vedere anche **cSolve()**, **cZeros()**, **solve()** e **zeros()**.

O

OneVar (Statistiche a una variabile)

Catalogo >

OneVar [1,]X,[Freq],[Categoria,Includi]]

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.

nSolve($x^2+5 \cdot x - 25 = 9, x < 0$)	-8.84429
nSolve($\frac{(1+r)^{24}-1}{r} = 26, r$)	$r > 0$ and $r < 0.25$
	0.006886

nSolve($x^2=-1, x$)	"No solution found"
-----------------------	---------------------

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 189).

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 254.

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

Variabile di output	Descrizione
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

Catalogo >

BooleanExpr1 or *BooleanExpr2* restituisce
Boolean expression

$x \geq 3$ or $x \geq 4$ $x \geq 3$

BooleanList1 or *BooleanList2* restituisce
Boolean list

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

If $x \leq 0$ or $x \geq 5$

Goto end

Return $x \cdot 3$

Lbl end

EndFunc

$g(3)$ 9

$g(0)$ A function did not return a value

BooleanMatrix1 or *BooleanMatrix2*
restituisce Boolean matrix

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.

Inter01 or *Inter02* \Rightarrow intero

In modalità base Esadecimale:

0h7AC36 or 0h3D5F 0h7BD7F

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.

Importante: è zero, non la lettera O.

In modalità base Bin:

0b100101 or 0b100 0b100101

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 22.

Nota: vedere xor.

ord() Codice numerico di carattere

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

►Rx() (Coordinata x rettangolare)

►Rx(*rEspr*, *θEspr*)⇒espressione

►Rx(*rLista*, *θLista*)⇒lista

►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.

In modalità angolo in radianti:

►Rx(<i>r</i> , <i>θ</i>)	$\cos(\theta) \cdot r$
►Rx(4,60°)	2
►Rx({ -3,10,1.3 }, { $\frac{\pi}{3}$, $\frac{-\pi}{4}$, 0 })	$\left\{ \frac{-3}{2}, 5\sqrt{2}, 1.3 \right\}$

P►Rx() (Coordinata x rettangolare)

Catalogo >

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

P►Ry() (Coordinata y rettangolare)

Catalogo >

P►Ry(*rEspr*, *θEspr*)⇒*espressione*

P►Ry(*rLista*, *θLista*)⇒*lista*

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. 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@>Ry (...)**.

In modalità angolo in radianti:

$$\begin{array}{ll} \text{P►Ry}(r, \theta) & \sin(\theta) \cdot r \\ \text{P►Ry}(4, 60^\circ) & 2 \cdot \sqrt{3} \\ \text{P►Ry}\left(\{-3, 10, 1.3\}, \left\{\frac{\pi}{3}, \frac{-\pi}{4}, 0\right\}\right) & \left\{\frac{-3 \cdot \sqrt{3}}{2}, -5 \cdot \sqrt{2}, 0\right\} \end{array}$$

PassErr

Catalogo >

PassErr

Per un esempio di **PassErr**, vedere l'esempio 2 del comando **Try**, pagina 205.

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 31, e **Try**, pagina 205.

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 7.

Define $p(x)=\begin{cases} x, & x>0 \\ \text{undef}, & x\leq 0 \end{cases}$	<i>Done</i>
$p(1)$	1
$p(-1)$	undef

poissCdf() (Probabilità cumulata per la distribuzione discreta di Poisson)

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(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)

Vettore ►Polar

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

Visualizza vettore in forma polare $[r \angle \theta]$. 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.

Nota: vedere anche ►Rect, pagina 156.

valoreComplesso ►Polar

Visualizza vettoreComplesso in forma polare.

- In modalità angolo in gradi, restituisce $(r \angle \theta)$.
- In modalità angolo in radianti, restituisce $r e^{i\theta}$.

valoreComplesso può avere una forma complessa qualsiasi. Tuttavia una voce $r e^{i\theta}$ causa un errore nella modalità di misurazione degli angoli in gradi.

Nota: è necessario usare le parentesi per un inserimento polare $(r \angle \theta)$.

[1 3.] ►Polar [3.16228 ∠ 1.24905]

[x y] ►Polar

$$\left[\sqrt{x^2+y^2} \angle \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right) \right]$$

In modalità angolo in radianti:

$(3+4 \cdot i)$ ►Polar $e^{i\left(\frac{\pi}{2}-\tan^{-1}\left(\frac{3}{4}\right)\right)}$

$\left(4 \angle \frac{\pi}{3}\right)$ ►Polar $e^{\frac{i\pi}{3}} \cdot 4$

In modalità angolo in gradi (gradi centesimali):

$(4 \cdot i)$ ►Polar $(4 \angle 100)$

In modalità angolo in gradi:

$(3+4 \cdot i)$ ►Polar $\left(5 \angle 90-\tan^{-1}\left(\frac{3}{4}\right)\right)$

polyCoeffs()

Catalogo >

polyCoeffs(Poli [,Var])⇒lista

Restituisce una lista di coefficienti di polinomio *Poli* in funzione della variabile *Var*.

Poli deve essere un'espressione polinomiale in *Var*. Si consiglia di non omettere *Var* a meno che *Poli* non sia un'espressione in una variabile singola.

polyCoeffs($4 \cdot x^2 - 3 \cdot x + 2, x$) {4,-3,2}

polyCoeffs($(x-1)^2 \cdot (x+2)^3$) {1,4,1,-10,-4,8}

Espande il polinomio e seleziona *x* per la *Var* omessa.

polyCoeffs($(x+y+z)^2, x$) {1,2·(y+z),(y+z)^2}

polyCoeffs($(x+y+z)^2, y$) {1,2·(x+z),(x+z)^2}

polyCoeffs($(x+y+z)^2, z$) {1,2·(x+y),(x+y)^2}

polyDegree()

Catalogo >

polyDegree(Poli [,Var])⇒valore

Restituisce il grado dell'espressione polinomiale *Poli* in funzione della variabile *Var*. Se si omette *Var*, la funzione **polyDegree()** seleziona un'impostazione predefinita dalle variabili contenute nel polinomio *Poli*.

Poli deve essere un'espressione polinomiale in *Var*. Si consiglia di non omettere *Var* a meno che *Poli* non sia un'espressione in una sola variabile.

polyDegree(5) 0

polyDegree(ln(2)+π,x) 0

Polinomi costanti

polyDegree($4 \cdot x^2 - 3 \cdot x + 2, x$) 2

polyDegree($(x-1)^2 \cdot (x+2)^3$) 5

polyDegree($(x+y^2+z^3)^2, x$) 2

polyDegree($(x+y^2+z^3)^2, y$) 4

polyDegree($(x-1)^{10000}, x$) 10000

Il grado può essere estratto, ma non i coefficienti. Questo perché è possibile estrarre il grado senza espandere il polinomio.

polyEval() (Calcola polinomio)**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.

<code>polyEval({{a,b,c}},x)</code>	$a \cdot x^2 + b \cdot x + c$
<code>polyEval({{1,2,3,4}},2)</code>	26
<code>polyEval({{1,2,3,4}},{{2,-7}})</code>	{26,-262}

polyGcd()**polyGcd(Espr1,Espr2)⇒espressione**

Restituisce il massimo comune divisore (gcd) dei due argomenti.

Espr1 e *Espr2* devono essere espressioni polinomiali.

Non sono ammessi come argomenti liste, matrici e booleani.

<code>polyGcd(100,30)</code>	10
<code>polyGcd(x^2-1,x-1)</code>	$x-1$
<code>polyGcd(x^3-6*x^2+11*x-6,x^2-6*x+8)</code>	$x-2$

polyQuotient()**polyQuotient(Poli1,Poli2
[,Var])⇒espressione**

Restituisce il quoziente del polinomio *Poli1* diviso per il polinomio *Poli2* in funzione della variabile specificata *Var*.

Poli1 e *Poli2* devono essere espressioni polinomiali in *Var*. Si consiglia di non omettere *Var* a meno che *Poli1* e *Poli2* non siano espressioni nella stessa variabile singola.

<code>polyQuotient(x-1,x-3)</code>	1
<code>polyQuotient(x-1,x^2-1)</code>	0
<code>polyQuotient(x^2-1,x-1)</code>	$x+1$
<code>polyQuotient(x^3-6*x^2+11*x-6,x^2-6*x+8)</code>	x

<code>polyQuotient((x-y)*(y-z),x+y+z,y)</code>	$y-z$
<code>polyQuotient((x-y)*(y-z),x+y+z,z)</code>	$2 \cdot x - y + 2 \cdot z$

polyQuotient()**Catalogo >**

$$\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, z) = -(x-y)$$

polyRemainder()**polyRemainder(Poli1,Poli2 [,Var])**⇒espressione

Restituisce il resto del polinomio *Poli1* diviso per il polinomio *Poli2* in funzione della variabile specificata *Var*.

Poli1 e *Poli2* devono essere espressioni polinomiali in *Var*. Si consiglia di omettere *Var* a meno che *Poli1* e *Poli2* non siano espressioni nella stessa variabile singola.

Catalogo >

polyRemainder($x-1, x-3$)	2
polyRemainder($x-1, x^2-1$)	$x-1$
polyRemainder($x^2-1, x-1$)	0

polyRemainder($(x-y) \cdot (y-z), x+y+z, x$)	$-(y-z) \cdot (2 \cdot y + z)$
polyRemainder($(x-y) \cdot (y-z), x+y+z, y$)	$-2 \cdot x^2 - 5 \cdot x \cdot z - 2 \cdot z^2$
polyRemainder($(x-y) \cdot (y-z), x+y+z, z$)	$(x-y) \cdot (x+2 \cdot y)$

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: {}.

Poli deve essere un polinomio in una variabile.

La seconda sintassi, **PolyRoots(ListaDiCoeff)**, restituisce una lista di radici reali per i coefficienti di *ListadiCoeff*.

Nota: vedere anche **cPolyRoots()**, pagina 43.

polyRoots(y^3+1, y)	{-1}
cPolyRoots(y^3+1, y)	$\left\{-1, \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i, \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i\right\}$
polyRoots($x^2+2 \cdot x+1, x$)	{-1,-1}
polyRoots({1,2,1})	{-1,-1}

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

Calcola la regressione su potenzay = (a · (x)^b) sulle liste X e Y con frequenza *Freq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189).

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 254.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: a · (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 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 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 240.

Product (PI) (Prodotto)

Vedere P(), pagina 240.

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*]])⇒*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 254.

product({1,2,3})	24
product({2,x,y})	2·x·y
product({4,5,8,9},2,3)	40

product{{1 2 3} 4 5 6 7 8 9})	[28 80 162]
product{{1 2 3} 4 5 6 7 8 9},1,2)	[4 10 18]

propFrac() (Frazione propria)

Catalogo >

propFrac(*Esp1*[, *Var*])⇒*espressione*

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.

Nelle espressioni razionali, **propFrac()** costituisce un'alternativa più veloce ma meno completa di **expand()**.

propFrac(4/3)	1+1/3
propFrac(-4/3)	-1-1/3

propFrac(x^2+x+1/x+1+y^2+y+1/y+1,x)	
	1/x+1+y^2+y+1/y+1
propFrac(Ans)	1/x+1+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.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

QR *m1,qm,rm* Done

<i>qm</i>	0.123091	0.904534	0.408248
	0.492366	0.301511	-0.816497
	0.86164	-0.301511	0.408248

<i>rm</i>	8.12404	9.60114	11.0782
	0.	0.904534	1.80907
	0.	0.	0.

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

QR *m1,qm,rm* Done

<i>qm</i>	$\frac{m}{\sqrt{m^2+o^2}}$	$\frac{\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}}$
	$\frac{o}{\sqrt{m^2+o^2}}$	$\frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}}$

<i>rm</i>	$\sqrt{m^2+o^2}$	$\frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}}$
	0	$\frac{ m \cdot p - n \cdot o }{\sqrt{m^2+o^2}}$

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

Calcola la regressione polinomiale
 $quadraticay = a \cdot x^2 + b \cdot x + c$ sulle liste *X* e *Y*
 con frequenza *Freq*. Il riepilogo dei risultati
 è memorizzato nella variabile *stat.results*.
 (pagina 189).

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 254.

Variabile di output	Descrizione
stat.RegEqn	Equazione di regressione: $a \cdot x^2 + b \cdot x + c$
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 189).

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 254.

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 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>

Variabile di output	Descrizione
stat.FreqReg	Lista di frequenze corrispondenti a stat.XReg e stat.YReg

R

R▶Pθ() (Coordinata polare)

Catalogo >

R▶Pθ (xEspr, yEspr)⇒espressione

In modalità angolo in gradi:

$$R▶Pθ(x,y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

R▶Pθ (xLista, yLista)⇒lista

R▶Pθ (xMatrice, yMatrice)⇒matrice

Restituisce la coordinata θ equivalente alla coppia di argomenti (x,y).

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 R@>Ptheta (...).

In modalità angolo in gradi (gradi centesimali):

$$R▶Pθ(x,y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

In modalità angolo in radianti:

$$R▶Pθ(3,2) \quad \tan^{-1}\left(\frac{2}{3}\right)$$

$$R▶Pθ\left[\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right] \\ \left[0 \quad \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} \quad 0.643501 \right]$$

R▶Pr() Coordinata polare

Catalogo >

R▶Pr (xEspr, yEspr)⇒espressione

In modalità angolo in radianti:

$$R▶Pr(3,2) \quad \sqrt{13}$$

$$R▶Pr(x,y) \quad \sqrt{x^2+y^2}$$

$$R▶Pr\left[\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right] \\ \left[3 \quad \sqrt{\frac{\pi^2+256}{4}} \quad 2.5 \right]$$

R▶Pr (xLista, yLista)⇒lista

R▶Pr (xMatrice, yMatrice)⇒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 (...).

►Rad in angolo radiante

Catalogo >

EsprI►*Rad*⇒*espressione*

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 $\{0.02618\}^r$

In modalità angolo in gradi (gradi centesimali):

(1.5)►Rad $\{0.023562\}^r$

rand() (Numero casuale)

Catalogo >

rand()⇒*espressione*

Imposta il seme dei numeri casuali.

rand(*numTentativi*)⇒*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.

RandSeed 1147	Done
rand(2)	$\{0.158206, 0.717917\}$

randBin() (Numero casuale da distribuzione binomiale)

Catalogo >

randBin(*n, p*)⇒*espressione*

randBin(80,0.5) 42

randBin(*n, p, numTentativi*)⇒*lista*

randBin(80,0.5,3) {41,32,39}

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

randInt(3,10) 5

(*EstremoInf*,*EstremoSup*)⇒*espressione*

randInt(3,10,4) {9,7,5,8}

randInt

randInt()
 $(EstremoInf, EstremoSup, numTentativi) \Rightarrow lista$

randInt(*EstremoInf*,*EstremoSup*)
 restituisce un intero casuale all'interno dell'intervallo specificato dagli numeri interi *EstremoInf* e *EstremoSup*.

**randInt(*EstremoInf*,*EstremoSup*,*numTe
ntativi*)** restituisce un elenco contenente *NumTentativi* interi casuali all'interno dell'intervallo specificato.

randMat() (Matrice casuale)

**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)

8	-3	6
-2	3	-6
0	4	-6

Nota: i valori di questa matrice cambiano ogni volta che si preme **enter**.

randNorm() (Normale casuale)

randNorm(μ , σ) $\Rightarrow espressione$

randNorm(μ , σ , *numTentativi*) $\Rightarrow lista$

Restituisce un numero decimale dalla distribuzione normale specifica. Può essere qualsiasi numero reale, anche se con maggiore probabilità sarà compreso nell'intervallo $[\mu - 3 \cdot \sigma, \mu + 3 \cdot \sigma]$.

randNorm(μ , σ , *numTentativi*) restituisce una lista contenente *numTentativi* valori decimali dalla distribuzione normale specificata.

RandSeed 1147

Done

randNorm(0,1)

0.492541

randNorm(3,4.5)

-3.54356

randPoly() (Polinomio casuale)**Catalogo >** **randPoly(Var, Ordine)⇒espressione**

Restituisce un polinomio in *Var* dell'*Ordine* specificato. I coefficienti sono numeri interi casuali compresi nell'intervallo 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])⇒lista**

Restituisce una lista contenente un campione casuale di *NumTentativi* da *Lista* con l'opzione di sostituire (*noSostituzione*=0) o meno (*noSostituzione*=1) il campione. L'impostazione predefinita prevede la sostituzione del campione.

Define list3={1,2,3,4,5}

Done

Define list4=randSamp(list3,6)

Done

list4

{2,3,4,3,1,2}

RandSeed (Seme numero casuale)**Catalogo >** **RandSeed Numero**

Se *Numero* = 0, imposta i semi ai valori predefiniti per il generatore di numeri casuali. Se *Numero* ≠ 0, viene utilizzato per generare due semi, memorizzati nelle variabili di sistema seed1 e seed2.

RandSeed 1147

Done

rand()

0.158206

real() (Reale)**Catalogo >** **real(Espr1)⇒espressione**

Restituisce la parte reale dell'argomento.

Nota: tutte le variabili non definite vengono considerate come variabili reali. vedere anche **imag()**, pagina 96.

real(Lista1)⇒lista

Restituisce le parti reali di tutti gli elementi.

real(Matrice1)⇒matrice

real(2+3·i)

2

real(z)

z

real(x+i·y)

x

real({a+i·b,3,i})

{a,3,0}

real([[a+i·b c], [3 i]])

\begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}

Restituisce le parti reali di tutti gli elementi.

►Rect (Visualizza come vettore rettangolare)

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 142.

valoreComplesso ►Rect

Visualizza *valoreComplesso* nella forma rettangolare a+bi. *valoreComplesso* può avere una forma complessa qualsiasi. Tuttavia un inserimento nella forma $re^{i\theta}$ causa un errore nella modalità di misurazione degli angoli in gradi.

Nota: è necessario usare le parentesi per un inserimento polare ($r \angle \theta$).

$$\begin{aligned} &\left[3 \angle \frac{\pi}{4} \angle \frac{\pi}{6} \right] \blacktriangleright \text{Rect} \\ &\left[\frac{3\sqrt{2}}{4} \quad \frac{3\sqrt{2}}{4} \quad \frac{3\sqrt{3}}{2} \right] \\ &\left[a \angle b \angle c \right] \\ &\left[a \cdot \cos(b) \cdot \sin(c) \quad a \cdot \sin(b) \cdot \sin(c) \quad a \cdot \cos(c) \right] \end{aligned}$$

In modalità angolo in radianti:

$$\begin{aligned} &\left(4 \cdot e^{\frac{\pi}{3}} \right) \blacktriangleright \text{Rect} && \frac{\pi}{3} \\ &\left(4 \angle \frac{\pi}{3} \right) \blacktriangleright \text{Rect} && 2 + 2\sqrt{3} \cdot i \end{aligned}$$

In modalità angolo in gradi (gradi centesimali):

$$\left(1 \angle 100 \right) \blacktriangleright \text{Rect} \quad i$$

In modalità angolo in gradi:

$$\left(4 \angle 60 \right) \blacktriangleright \text{Rect} \quad 2 + 2\sqrt{3} \cdot i$$

Nota: per inserire \angle dalla tastiera, selezionarlo nell'elenco dei simboli del Catalogo.

ref() (Forma a scalini per righe)

Catalogo >

ref(*Matrice1*[, *Tol*])⇒*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.

- 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:
 $5E-14 \cdot \max(\dim(\text{Matrice1})) \cdot \text{rowNorm}(\text{Matrice1})$

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:

$$\text{ref}\begin{pmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

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.

$$\text{ref}\begin{pmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{pmatrix} \quad \begin{pmatrix} 1 & -2 & -4 & 4 \\ 5 & 5 & 5 & 5 \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 7 & 7 & 7 & 7 \\ 0 & 0 & 1 & \frac{-62}{71} \\ 71 & 71 & 71 & 71 \end{pmatrix}$$

$$\begin{matrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow m1 \\ \text{ref}(m1) \end{matrix} \quad \begin{matrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\ \begin{bmatrix} 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix} \end{matrix}$$

ref() (Forma a scalini per righe)

Catalogo >

$$\text{ref} \left[\begin{array}{ccc} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right] | a=0 \quad \left[\begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right]$$

Nota: vedere anche **rref()**, pagina 166.**remain() (Resto)**

Catalogo >

remain(*Espr1, Espr2*)⇒espressione**remain(*Lista1, Lista2*)**⇒lista**remain(*Matrice1, Matrice2*)**⇒matrice

Restituisce il resto del primo argomento rispetto al secondo argomento, come definito dalle identità:

$$\text{remain}(x, 0) \quad x$$

$$\text{remain}(x, y) \quad x - y \cdot \text{iPart}(x/y)$$

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 123.

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}

$$\text{remain} \left[\begin{bmatrix} 9 & -7 \\ 6 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 3 \\ 4 & -3 \end{bmatrix} \right] \quad \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$$

Request

Catalogo >

Request(*stringaPrompt, var[, FlagDispl [, varStato]]*)

Definire un programma:

Define request_demo()=Prgm

Request "Radius: ", r

Disp "Area = ", pi*r^2

EndPrgm

Request(*stringaPrompt, funz(arg1, ...argn [, FlagDispl [, varStato]]*)

Eseguire il programma e digitare una risposta:

request_demo()

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.

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.



Risultato dopo aver selezionato **OK**:

Radius: 6/2

Area= 28.2743

Definire un programma:

Define polynomial()=Prgm

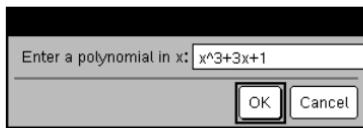
Request "Enter a polynomial in
x:",p(x)

Disp "Real roots are:",polyRoots(p
(x),x)

EndPrgm

Eseguire il programma e digitare una
risposta:

polynomial()



Risultato dopo aver selezionato **OK**:

Enter a polynomial in x: x^3+3x+1

Le radici reali sono: {-0.322185}

Per arrestare un programma che contiene un comando **Request in un loop infinito**:

- **Palmare:** Tenere premuto il tasto  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**, pagina 160.

RequestStr

RequestStr *stringaPrompt, var[, FlagDispl]*

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  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 **Request**, pagina 158.

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 *FlagDispl* 0 omette la richiesta e la risposta dalla cronologia):

```
requestStr_demo()
```

La risposta ha 5 caratteri.

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(Lista[, Num])⇒lista****right({1,3,-2,4},3)**

{3,-2,4}

Restituisce i *Num* elementi più a destra contenuti in *Lista*.

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

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, VarMax}, varDipendente0, incrVar [, tollErrore])⇒matrice**

Equazione differenziale:

 $y'=0.001*y*(100-y)$ e $y(0)=10$ **rk23(SistemaDiEspr, Var, ListaDiVarDipendenti, {Var0, VarMax}, ListaDiVarDipendenti0, incrVar [,****rk23(0.001·y·(100-y),t,y,{0,100},10,1)
[0. 1. 2. 3. 4.,
10. 10.9367 11.9493 13.042 14.2]**

tollErrore]) \Rightarrow matrice

rk23(*ListaDiEspr*, *Var*,
ListaDiVarDipendenti, {*Var0*, *VarMax*},
ListaDiVarDipendenti0, *incrVar* [,
tollErrore]) \Rightarrow matrice

Utilizza il metodo di Runge-Kutta per risolvere il sistema

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

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 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*.

ListaDiVarDipendenti0 è una lista di valori iniziali di variabili dipendenti.

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

Stessa equazione con *tollErrore* impostata su 1.E-6

$$\begin{aligned} \text{rk23}\left[0.001 \cdot y \cdot (100-y), t, y, \{0, 100\}, 10, 1, 1 \cdot 10^{-6}\right] \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9367 & 11.9495 & 13.0423 & 14.2189 \end{bmatrix} \end{aligned}$$

Confrontare il precedente risultato con la soluzione esatta del CAS ottenuta utilizzando *deSolve()* e *seqGen()*:

$$\begin{aligned} \text{deSolve}\left[y'=0.001 \cdot y \cdot (100-y) \text{ and } y(0)=10, t, y\right] \\ y=\frac{100 \cdot (1.10517)^t}{(1.10517)^t+9}. \end{aligned}$$

$$\begin{aligned} \text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t+9}, t, y, \{0, 100\}\right) \\ \{10., 10.9367, 11.9494, 13.0423, 14.2189, 15.48\} \end{aligned}$$

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 $y2(0)=5$

$$\begin{aligned} \text{rk23}\left[\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0.5\}, \{2.5\}, 1\right] \\ \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 2. & 1.94103 & 4.78694 & 3.25253 & 1.82848 \\ 5. & 16.8311 & 12.3133 & 3.51112 & 6.27245 \end{bmatrix} \end{aligned}$$

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()

Catalogo > 

root(*Espr*) \Rightarrow radice

$$\sqrt[3]{2}$$

root(*Espr1*, *Espr2*) \Rightarrow *radice*

$$\frac{1}{\sqrt[3]{2}}$$

root(*Espr*) restituisce la radice quadrata di *Espr*.

$$\begin{array}{r} & 3^3 \\ \hline 3\sqrt[3]{2} & 1.44225 \end{array}$$

root(*Espr1*, *Espr2*) restituisce la radice *Espr2*-esima di *Espr1*. *Espr1* può essere una costante reale o complessa a virgola mobile, una costante razionale intera o complessa o un'espressione simbolica generale.

Nota: vedere anche **Modello di radice ennesima**, pagina 5.

rotate() (Ruota)

Catalogo >

rotate(*Interol[,numRotazioni]***)** \Rightarrow *intero*

In modalità base Bin:

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 22.

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

rotate() (Ruota)

Catalogo >

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.

0b0000000000000001111010110000110101

L'ultimo bit a destra diventa l'ultimo 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 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).

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)	"bcd"

round() (Arrotondamento)

Catalogo >

round(Espresso[, cifre])⇒espressione

round(1.234567,3)	1.235
-------------------	-------

Restituisce l'argomento arrotondato ad un numero specifico di cifre dopo la virgola decimale.

round() (Arrotondamento)

Catalogo >

cifre deve essere un numero intero compreso tra 0 e 12. Se *cifre* non viene incluso, la funzione restituisce l'argomento arrotondato a 12 cifre significative.

Nota: la visualizzazione dipende dalla modalità selezionata.

round(Lista1[, cifre])⇒lista

Restituisce una lista degli elementi arrotondati al numero specifico di cifre.

$$\text{round}(\{\pi, \sqrt{2}, \ln(2)\}, 4) \\ \{3.1416, 1.4142, 0.6931\}$$

round(Matrice1[, cifre])⇒matrice

Restituisce una matrice degli elementi arrotondati al numero specifico di cifre.

$$\text{round}\left[\begin{array}{cc} \ln(5) & \ln(3) \\ \pi & e^1 \end{array}\right], 1 \\ \begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$$

rowAdd() (Somma di righe di matrice)

Catalogo >

rowAdd(Matrice1, rIndice1, rIndice2)⇒matrice

Restituisce una copia di *Matrice1* nella quale la riga *rIndice2* è sostituita dalla somma delle righe *rIndice1* e *rIndice2*.

$$\text{rowAdd}\left[\begin{array}{cc} 3 & 4 \\ -3 & 2 \end{array}\right], 1, 2 \\ \begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$$

$$\text{rowAdd}\left[\begin{array}{cc} a & b \\ c & d \end{array}\right], 1, 2 \\ \begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$$

rowDim() (Dimensione righe matrice)

Catalogo >

rowDim(Matrice)⇒espressione

Restituisce il numero di righe di *Matrice*.

Nota: vedere anche **colDim()**, pagina 32.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1 \\ \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

$$\text{rowDim}(m1) \\ 3$$

rowNorm() (Norma righe matrice)

Catalogo >

rowNorm(Matrice)⇒espressione

Restituisce 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 32.

$$\text{rowNorm}\left[\begin{array}{ccc} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{array}\right] \\ 25$$

rowSwap() (Inverti righe matrice)

Catalogo >

**rowSwap(*Matrice1*, *rIndice1*,
rIndice2)** \Rightarrow matriceRestituisce *Matrice1* con le righe *rIndice1* e *rIndice2* scambiate.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \xrightarrow{\text{rowSwap}} mat$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
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*.

$\text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$
--	---

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.

rref($\begin{bmatrix} a & b \\ c & d \end{bmatrix}$)	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
--	--

- 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 ·max(dim(*Matrice1*)) ·rowNorm (*Matrice1*)

Nota: vedere anche **ref()**, pagina 157.**S****sec() (Secante)**

Tasto

sec(*Espr1*) \Rightarrow espressione

In modalità angolo in gradi:

sec(*List1*) \Rightarrow lista

sec() (Secante)

Tasto

Restituisce la secante di *espressione1* oppure restituisce una lista contenente le secanti di tutti gli elementi in *Lista1*.

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.

$$\begin{array}{ll} \sec(45) & \sqrt{2} \\ \sec(\{1,2,3,4\}) & \left\{ \frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)} \right\} \end{array}$$

sec⁻¹⁽⁾ (Secante inversa)

Tasto **sec⁻¹(Espr1) ⇒ espressione**

In modalità angolo in gradi:

sec⁻¹(Lista1) ⇒ lista

$$\begin{array}{ll} \sec^{-1}(1) & 0 \end{array}$$

Restituisce l'angolo la cui secante è *Espr1* oppure restituisce una lista contenente le secanti inverse di ciascun elemento di *Lista1*.

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 **arcsec(...)**.

In modalità angolo in gradi (gradi centesimali):

$$\begin{array}{ll} \sec^{-1}(\sqrt{2}) & 50 \end{array}$$

In modalità angolo in radianti:

$$\begin{array}{ll} \sec^{-1}(\{1,2,5\}) & \left\{ 0, \frac{\pi}{3}, \cos^{-1}\left(\frac{1}{5}\right) \right\} \end{array}$$

sech() (Secante iperbolica)

Catalogo > **sech(Espr1) ⇒ espressione**

$$\begin{array}{ll} \text{sech}(3) & \frac{1}{\cosh(3)} \end{array}$$

sech(Lista1) ⇒ lista

$$\begin{array}{ll} \text{sech}(\{1,2,3,4\}) & \left\{ \frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)} \right\} \end{array}$$

Restituisce la secante iperbolica di *Espr1* oppure restituisce una lista contenente le secanti iperboliche di tutti gli elementi di *Lista1*.

sech⁻¹(*Espr1*) ⇒ *espressione***sech⁻¹(*Listal*)** ⇒ *lista*

Restituisce la secante iperbolica inversa di *Espr1* oppure restituisce una lista contenente le secanti iperboliche inverse di ciascun elemento di *Listal*.

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

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

sech ⁻¹ (1)	0
sech ⁻¹ ({1,-2,2,1})	$\left\{0, \frac{2\pi}{3} \cdot i, 8 \cdot 10^{-15} + 1.07448 \cdot i\right\}$

Send

Menu Hub

Send *exprOrString1[, exprOrString2] ...*

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.

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 87), **GetStr** (pagina 91) e **eval()** (pagina 70).

Esempio: attivare l'elemento blu del LED RGB integrato per 0,5 secondi.

Send "SET COLOR.BLUE ON TIME .5"

Done

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 "READ BRIGHTNESS" *Done*

Get *lightval* *Done*

lightval 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.

n:=50 50

m:=4 4

Send "SET SOUND eval(*m*·*n*)" *Done*

iostr.SendAns "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.

$\text{seq}\left(n^2, n, 1, 6\right)$	$\{1, 4, 9, 16, 25, 36\}$
$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$	$\left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	$\frac{1968329}{1270080}$

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

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

Generare i primi 5 termini della sequenza *u*(*n*) = *u*(*n*-1)²/2, con *u*(1)=2 e *incrVar*=1.

$\text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right)$	$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$
---	---

Esempio in cui *Var0*=2:

$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right)$	$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$
---	--

Esempio in cui il termine iniziale è simbolico :

$\text{seqGen}\left(u(n-1)+2, n, u, \{1, 5\}, \{a\}\right)$	$\{a, a+2, a+4, a+6, a+8\}$
---	-----------------------------

seqGen()

Catalogo >

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.

Sistema di due successioni:

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u_2(n-1)}{2} + u_1(n-1)\right\}, n, \{u1, u2\}, \{1, 5\}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix}$$

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()

Catalogo >

seqn(*Espr{u, n [, ListaDiTermIniziali[, nMax [, ValArrotPerEccesso]]}]*) \Rightarrow *lista*

Genera una lista di termini per una successione $u(n)=\text{Espr}(u, n)$ 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 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 *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

Genera i primi 6 termini della successione $u(n)=u(n-1)/2$, con $u(1)=2$.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{ 2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360} \right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{ 1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36} \right\}$$

series() (Serie)**Catalogo > **

series[*Espr1*, *Var*, *Ordine* [, *Punto*]]⇒espressione

series[*Espr1*, *Var*, *Ordine* [, *Punto*]] | *Var*>*Punto*⇒espressione

series[*Espr1*, *Var*, *Ordine* [, *Punto*]] | *Var*<*Punto*⇒espressione

series $\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1 \right)$	$\frac{1}{2} - \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$
series $\left(\frac{-1}{e^{z_-}}, z \rightarrow 1 \right)$	z_-^{-1}
series $\left(\left(1 + \frac{1}{n} \right)^n, n, 2, \infty \right)$	$e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$

Restituisce una rappresentazione generalizzata in serie di potenze troncate di *Espr1* sviluppate intorno a *Punto* nel grado di *Ordine*. *Ordine* può essere qualsiasi numero razionale. Le potenze risultanti di (*Var* – *Punto*) possono includere esponenti negativi e/o frazionari. I coefficienti di queste potenze possono includere logaritmi di (*Var* – *Punto*) e altre funzioni di *Var* che sono dominate da tutte le potenze di (*Var* – *Punto*) aventi lo stesso segno esponenziale.

Il valore predefinito di Punto è 0. Punto può essere ∞ o –∞, in questi casi lo sviluppo è nel grado di Ordine in 1/(Var – Punto).

series(...) restituisce “**series(...)**” se non è in grado di determinare una tale rappresentazione, come ad esempio per singolarità essenziali quali $\sin(1/z)$ in corrispondenza di $z=0$ e $e^{-1/z}$ in corrispondenza di $z=0$ oppure $0 \cdot e^z$ in corrispondenza di $z = \infty$ o $-\infty$.

series $\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5 \right) x > 0$	$\frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$
series $\left(\int \frac{\sin(x)}{x} dx, x, 6 \right)$	$x - \frac{x^3}{18} + \frac{x^5}{600}$
series $\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7 \right)$	$\frac{x^3}{2} - \frac{x^5}{24} - \frac{29 \cdot x^7}{720}$

series $\left((1+e^x)^2, x, 2, 1 \right)$	$(e+1)^2 + 2 \cdot e \cdot (e+1) \cdot (x-1) + e \cdot (2 \cdot e+1) \cdot (x-1)^2$
--	---

Se la serie o una delle sue derivate presenta una discontinuità in corrispondenza del *Punto*, il risultato potrebbe contenere sottoespressioni della forma `sign(...)` o `abs(...)` per una variabile di espansione reale o della forma `(-1)^floor(...angle(...))` per una variabile di espansione complessa, che termina con `_`. Se si pensa di utilizzare le serie solo per i valori da una sola parte di *Punto*, aggiungere quello appropriato tra `"| Var > Punto"`, `"| Var < Punto"`, `"| Var ≥ Punto"` o `"Var ≤ Punto"` per ottenere un risultato più semplice.

series() può fornire approssimazioni simboliche a integrali indefiniti e integrali definiti per i quali non è possibile ottenere altrimenti soluzioni simboliche.

series() si distribuisce su liste e matrici indicate come primo argomento.

series() è una versione generalizzata di **taylor()**.

Come mostrato nell'ultimo esempio a destra, le routine di visualizzazione dopo il risultato prodotto da `series(...)` potrebbero ridisporre i termini in modo che il termine dominante non sia quello più a sinistra.

Nota: vedere anche **dominantTerm()**, pagina 63.

setMode()Catalogo > 

**setMode(*interoNomeModo*,
interoImpostazione)** ⇒ *intero*

setMode(*lista*) ⇒ *lista interi*

Valido solo all'interno di una funzione o di un programma.

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()

**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.

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 90.

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.

Define <i>prog1()</i> =Prgm Disp approx(π) setMode(1,16) Disp approx(π) EndPrgm	Done
<i>prog1()</i>	3.14159
	3.14

Done

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 =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, 3 =Esatto
Formato vettoriale (Vector Format)	6	1 =Rettangolare, 2 =Cilindrico, 3 =Sferico
Base	7	1 =Decimale, 2 =Esa, 3 =Binario
Sistema unità di misura (Unit system)	8	1 =SI, 2 =Eng/US

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 22.

In modalità base Bin:

shift(0b1111010110000110101)	0b111101011000011010
shift(256,1)	0b1000000000

In modalità base Esadecimale:

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.

`0b000000000000000111101011000011010`

Inserisce 0 se il bit più a sinistra è 0,
oppure 1 se il bit più a sinistra è 1.

Dà:

`0b000000000000000111101011000011010`

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".

<code>shift(0h78E)</code>	0h3C7
<code>shift(0h78E,-2)</code>	0h1E3
<code>shift(0h78E,2)</code>	0h1E38

Importante: per inserire un numero binario o esadecimale, utilizzare sempre il prefisso 0b o 0h (è uno zero, non la lettera O).

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 }

shift() (Sposta)**Catalogo > **

shift(*Stringa1*,
[,*numSpostamenti*]) \Rightarrow *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.

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

sign() (Segno)**Catalogo > **

sign(*Espr1*) \Rightarrow *espressione*

sign(-3.2)	-1.
sign({2,3,4,-5})	{1,1,1,-1}
sign(1+ x)	1

sign(*Listal*) \Rightarrow *lista*

sign(*Matrice1*) \Rightarrow *matrice*

Quando *Espr1* è reale o complessa, restituisce *Espr1*/abs(*Espr1*) se *Espr1* \neq 0.

Restituisce 1 se *Espr1* è positiva.

Restituisce -1 se *Espr1* è negativa.

sign(0) restituisce ± 1 se la modalità Formato complesso è Reale; altrimenti restituisce se stesso.

sign(0) rappresenta la circonferenza unitaria del dominio complesso.

Con liste o matrici, restituisce i segni di tutti gli elementi.

Se la modalità del formato complesso è Reale:

sign([-3 0 3])	[-1 ±1 1]
----------------	-----------

simult() Sistema di equazioni simultanee**Catalogo > **

simult(*matriceCoeff*, *vettoreCost*,
[*Tol*]) \Rightarrow *matrice*

Risolvere rispetto a x e y:

$$x + 2y = 1$$

$$3x + 4y = -1$$

simult() Sistema di equazioni simultanee

Catalogo >

Restituisce un vettore colonna che contiene le soluzioni di un sistema di equazioni lineari.

Nota: vedere anche **linSolve()**, pagina 109.

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*]) \Rightarrow *matrice*

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.

$$\text{simult}\left[\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right] = \begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

La soluzione è $x=-3$ e $y=2$.

Risolvere:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{array}{c} \left[\begin{array}{cc} a & b \\ c & d \end{array} \right] \rightarrow \text{matr1} \quad \left[\begin{array}{c} a & b \\ c & d \end{array} \right] \\ \text{simult}\left(\text{matr1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) = \begin{bmatrix} \frac{-(2 \cdot b - d)}{a \cdot d - b \cdot c} \\ \frac{2 \cdot a - c}{a \cdot d - b \cdot c} \end{bmatrix} \end{array}$$

Risolvere:

$$x + 2y = 1$$

$$3x + 4y = -3$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left[\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right] = \begin{bmatrix} -3 & -7 \\ 2 & 9/2 \end{bmatrix}$$

Per il primo sistema, $x=-3$ e $y=2$. Per il secondo sistema, $x=-7$ e $y=9/2$.

Espr ►sin

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

Rappresenta *Espr* rispetto al seno. È un operatore di conversione della visualizzazione. Può essere utilizzato solo alla fine della riga di introduzione.

►sin riduce tutte le potenze di cos(...) modulo 1-sin(...)^2 in modo che qualsiasi potenza rimanente di sin(...) abbia esponenti compresi nell'intervallo (0, 2). Pertanto, il risultato non conterrà cos(...) se e solo se cos(...) si presenta nell'espressione data solamente con esponenti pari.

Nota: questo operatore di conversione non è supportato nelle modalità di misurazione degli angoli in Gradi o Gradianti (gradi centesimali). Prima di utilizzarlo, accertarsi che la modalità angolare sia impostata su Radiani e che *Espr* non contenga riferimenti esplicativi ad angoli in gradi o gradianti.

sin() (Seno)

Tasto **sin(Espr1)⇒espressione**

In modalità angolo in gradi:

sin(Lista1)⇒lista

$$\sin\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

sin(Espr1) restituisce il seno dell'argomento sotto forma di espressione.

$$\sin(45) \quad \frac{\sqrt{2}}{2}$$

sin(Lista1) restituisce una lista di seni di tutti gli elementi di *Lista1*.

$$\sin(\{0,60,90\}) \quad \left\{0, \frac{\sqrt{3}}{2}, 1\right\}$$

Nota: l'argomento è interpretato come angolo in gradi, gradianti o radienti, a seconda della modalità angolo correntemente impostata. Si può usare °, G o † per escludere tale impostazione provvisoriamente.

In modalità angolo in gradianti (gradi centesimali):

$$\sin(50) \quad \frac{\sqrt{2}}{2}$$

In modalità angolo in radianti:

$$\frac{\sin\left(\frac{\pi}{4}\right)}{\sin(45^\circ)} = \frac{\frac{\sqrt{2}}{2}}{\frac{\sqrt{2}}{2}}$$

sin(*matriceQuadrata1*)⇒*matriceQuadrata*

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()**.

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

sin⁻¹() (Arcoseno)**sin⁻¹(*Espr1*)⇒*espressione*****sin⁻¹(*Listal*)⇒*lista***

sin⁻¹(*Espr1*) restituisce sotto forma di espressione l'angolo il cui seno è *Espr1*.

sin⁻¹(*Listal*) restituisce una lista contenente l'inversa del seno di ciascun elemento di *Listal*.

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()**.

In modalità angolo in gradi:

$$\sin^{-1}(1) = 90$$

In modalità angolo in gradi (gradi centesimali):

$$\sin^{-1}(1) = 100$$

In modalità angolo in radianti:

$$\sin^{-1}(\{0,0.2,0.5\}) = \{0.0, 0.201358, 0.523599\}$$

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

$$\begin{aligned} \sin^{-1}\left[\begin{matrix} 1 & 5 \\ 4 & 2 \end{matrix}\right] = & -0.174533 - 0.12198 \cdot i & 1.74533 - 2.35591 \cdot i \\ & 1.39626 - 1.88473 \cdot i & 0.174533 - 0.593162 \cdot i \end{aligned}$$

$\sin^{-1}()$ (Arcoseno)

Tasto

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

 $\sinh()$ (Seno iperbolico)

Catalogo >

 $\sinh(Espr1) \Rightarrow espressione$ $\sinh(Lista1) \Rightarrow lista$

$\sinh(Espr1)$ restituisce sotto forma di espressione il seno iperbolico dell'argomento.

$\sinh(Lista1)$ restituisce una lista del seno iperbolico di ciascun elemento di *Lista1*.

 \sinh $(matriceQuadrata1) \Rightarrow matriceQuadrata$

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.

$\sinh(1.2)$	1.50946
$\sinh(\{0,1,2,3\})$	$\{0,1.50946,10.0179\}$

In modalità angolo in radianti:

$\sinh\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$	$\begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$
---	---

 $\sinh^{-1}()$ (Arcoseno iperbolico)

Catalogo >

 $\sinh^{-1}(Espr1) \Rightarrow espressione$ $\sinh^{-1}(Lista1) \Rightarrow lista$

$\sinh^{-1}(Espr1)$ restituisce sotto forma di espressione l'inversa del seno iperbolico dell'argomento.

$\sinh^{-1}(Lista1)$ 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 **$\text{arcsinh}(...)$** .

 \sinh^{-1} $(matriceQuadrata1) \Rightarrow matriceQuadrata$

$\sinh^{-1}(0)$	0
$\sinh^{-1}(\{0,2,1,3\})$	$\{0,1.48748,\sinh^{-1}(3)\}$

In modalità angolo in radianti:

sinh⁻¹() (Arcoseno iperbolico)

Catalogo >

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.

$$\text{sinh}^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{pmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{pmatrix}$$

SinReg (Regressione sinusoidale)

Catalogo >

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 189).

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 254.

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 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>

solve() (Risolvi)

solve(Equazione, Var)⇒espressione booleana

$$\begin{aligned} &\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x) \\ &x = \frac{-b - \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \text{ or } x = \frac{-b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a} \end{aligned}$$

**solve(Equation,
Var=Campione)⇒espressione booleana**

solve(Disequazione, Var)⇒espressione booleana

Restituisce le possibili soluzioni reali di un'equazione o di una disequazione rispetto a *Var*. L'obiettivo è quello di restituire valori possibili per tutte le soluzioni.

Tuttavia, per alcune equazioni o disequazioni il numero di soluzioni può risultare infinito.

I valori possibili di soluzione possono non essere soluzioni finite reali per alcune combinazioni di valori delle variabili non definite.

Ans|a=1 and b=1 and c=1

$$x = \frac{-1 + \sqrt{3}}{2} \cdot i \text{ or } x = \frac{-1 - \sqrt{3}}{2} \cdot i$$

solve() (Risolvi)

Catalogo >

Nell'impostazione Auto della modalità **Auto o Approssimato**, l'obiettivo è di produrre soluzioni esatte quando semplici e supportate da ricerche iterative con approssimazione aritmetica quando le soluzioni esatte non sono determinabili.

Il massimo comune divisore del numeratore e del denominatore delle frazioni viene cancellato per impostazione predefinita; per questo motivo le soluzioni possono essere tali solo come limite da una o da entrambe le parti.

Per le disequazioni del tipo \geq , \leq , $<$ o $>$, è difficile ottenere soluzioni esplicite, tranne qualora la disequazione sia lineare e contenga solo *Var*.

Nell'impostazione Esatto della modalità **Auto o Approssimato**, le parti che non possono essere risolte vengono restituite come equazione implicita o disequazione.

L'uso dell'operatore ("|") di vincolo permette di limitare l'intervallo di soluzione e/o altre variabili presenti nell'equazione o nella disequazione. Quando viene trovata una soluzione in un intervallo, è possibile utilizzare gli operatori di disequazione per escludere lo stesso intervallo da ricerche successive.

false viene restituito quando non è stata trovata nessuna soluzione reale, mentre. true viene restituito se mediante **solve()** tutti i valori reali finiti di *Var* soddisfano l'equazione o la disequazione.

Poiché **solve()** restituisce sempre un risultato booleano, è possibile utilizzare gli operatori "and," "or" e "not" per combinare i risultati ottenuti con **solve()** reciprocamente o con altre espressioni booleane.

Le soluzioni possono contenere una nuova costante non definita univoca nella forma n_j , dove *j* rappresenta un numero intero compreso nell'intervallo tra 1 e 255. Tali variabili indicano un intero arbitrario.

solve $((x-a) \cdot e^x = -x \cdot (x-a), x)$

$x=a$ or $x=-0.567143$

$(x+1) \cdot \frac{x-1}{x-1} + x - 3$

$2 \cdot x - 2$

solve $(5 \cdot x - 2 \geq 2 \cdot x, x)$

$x \geq \frac{2}{3}$

exact $(\text{solve}((x-a) \cdot e^x = -x \cdot (x-a), x))$

$e^x + x = 0$ or $x = a$

In modalità angolo in radianti:

solve $(\tan(x) = \frac{1}{x}, x)$

$| x > 0 \text{ and } x < 1$

$x = 0.860334$

solve $(x = x + 1, x)$

false

solve $(x = x, x)$

true

$2 \cdot x - 1 \leq 1 \text{ and solve}(x^2 = 9, x)$

$x \neq -3 \text{ and } x \leq 1$

In modalità angolo in radianti:

solve $(\sin(x) = 0, x)$

$x = n1 \cdot \pi$

In modalità Reale, le potenze frazionarie con denominatori dispari mostrano solo la parte reale. Altrimenti, le espressioni con parti multiple quali le potenze frazionarie, i logaritmi e le funzioni trigonometriche inverse mostrano solo la parte principale. Pertanto, la funzione **solve()** restituisce solo soluzioni corrispondenti a tali parti reali o principali.

Nota: vedere anche **cSolve()**, **cZeros()**, **nSolve()** e **zeros()**.

**solve(Eqn1 and Eqn2 [and...],
VarOCampione1, VarOCampione2 [, ...
])**⇒espressione booleana

**solve(SistemaDiEquazioni,
VarOCampione1, VarOCampione2 [, ...
])**⇒espressione booleana

**solve({Eqn1, Eqn2 [...]} {VarOCampione1,
VarOCampione2 [, ...
]})** ⇒espressione booleana

Restituisce possibili soluzioni reali di un sistema di equazioni, dove ogni *varOCampione* specifica una variabile in base alla quale risolvere l'equazione.

È possibile separare le equazioni con l'operatore *and* oppure è possibile inserire un sistema di equazioni utilizzando un modello del Catalogo. Il numero di argomenti di *varOCampione* deve corrispondere al numero di equazioni. In alternativa è possibile specificare un valore campione iniziale per la variabile. Ogni *varOCampione* deve avere la forma:

variabile

– o –

variabile = numero reale o non reale

Ad esempio, *x* è valido come pure *x=3*.

$\text{solve}\left(\frac{1}{x^3} = 1, x\right)$	$x = -1$
$\text{solve}(\sqrt{x} = -2, x)$	false
$\text{solve}(-\sqrt{x} = -2, x)$	$x = 4$

$\text{solve}(y = x^2 - 2 \text{ and } x + 2 \cdot y = -1, \{x, y\})$	
$x = -\frac{3}{2}$ and $y = \frac{1}{4}$ or $x = 1$ and $y = -1$	

Se tutte le equazioni sono polinomiali e NON si indica alcun valore campione iniziale, **solve()** utilizza il metodo di eliminazione lessicale di Gröbner/Buchberger per determinare tutte le soluzioni reali.

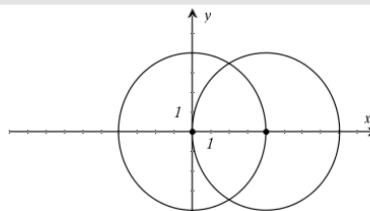
Supponiamo ad esempio di avere una circonferenza con raggio r all'origine e un'altra con raggio r con centro nel punto in cui la prima circonferenza interseca l'asse positivo x . Utilizzare **solve()** per trovare le intersezioni.

Come indicato da r nell'esempio a destra, le equazioni polinomiali simultanee possono avere variabili aggiuntive senza valori, ma che rappresentano valori numerici dati che potrebbero essere sostituiti successivamente.

È possibile, in aggiunta o in alternativa, includere variabili risolutorie che non compaiono nelle equazioni. Ad esempio è possibile includere z quale variabile risolutoria per estendere l'esempio precedente a due cilindri paralleli con raggio r che si intersecano.

Le soluzioni dei cilindri mostrano come le famiglie di soluzioni possano contenere costanti arbitrarie della forma ck , dove k è un suffisso intero compreso tra 1 e 255

Per i sistemi polinomiali, i tempi di calcolo o l'insufficienza di memoria possono dipendere in gran parte dall'ordine in cui sono elencate le variabili risolutorie. Se la scelta iniziale esaurisce la memoria (o la pazienza), provare a ridisporre le variabili all'interno delle equazioni e/o della lista *varOCampione*.



$$\begin{aligned} \text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right) \\ x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3} \cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=\frac{-\sqrt{3} \cdot r}{2} \end{aligned}$$

$$\begin{aligned} \text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right) \\ x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3} \cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y= \end{aligned}$$

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

solve() (Risolvi)

Catalogo >

Se non viene indicato alcun valore campione e se anche una sola equazione è non polinomiale per una qualsiasi variabile ma tutte le equazioni sono lineari per le variabili risolutorie, **solve()** utilizza l'eliminazione gaussiana per tentare di determinare tutte le soluzioni reali.

Se un sistema non è né polinomiale per tutte le sue variabili né lineare per le variabili risolutorie, **solve()** determina al più una soluzione tramite un metodo di approssimazione iterativa. A tale scopo, il numero di variabili risolutorie deve essere uguale al numero di equazioni e tutte le altre variabili delle equazioni devono poter essere semplificate in numeri.

Ogni variabile risolutoria inizia con il valore campione se ce n'è uno; altrimenti inizia con 0.0.

Utilizzare più valori campione per provare altre soluzioni una alla volta. Per ottenere una convergenza, può essere necessario che il valore campione debba essere abbastanza prossimo alla soluzione.

$$\text{solve}\left(x+e^z \cdot y=1 \text{ and } x-y=\sin(z), \{x,y\}\right)$$

$$x = \frac{e^z \cdot \sin(z) + 1}{e^z + 1} \text{ and } y = \frac{-(\sin(z) - 1)}{e^z + 1}$$

$$\text{solve}\left(e^z \cdot y=1 \text{ and } -y=\sin(z), \{y,z\}\right)$$

$$y=2.812e^{-10} \text{ and } z=21.9911 \text{ or } y=0.001871$$

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

$$\text{solve}\left(e^z \cdot y=1 \text{ and } -y=\sin(z), \{y,z\}\right)$$

$$y=0.001871 \text{ and } z=6.28131$$

SortA (Ordinamento ascendente)

Catalogo >

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.

Tutti gli argomenti devono essere nomi di liste o di vettori e devono avere le stesse dimensioni.

$$\{2,1,4,3\} \rightarrow list1$$
$$\{2,1,4,3\}$$

SortA *list1*

Done

list1

$$\{1,2,3,4\}$$

$$\{4,3,2,1\} \rightarrow list2$$
$$\{4,3,2,1\}$$

SortA *list2,list1*

Done

list2

$$\{1,2,3,4\}$$

list1

$$\{4,3,2,1\}$$

Gli elementi vuoti (nulli) nel primo argomento si spostano in fondo. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 254.

SortD (Ordinamento discendente)

SortD *List1[, 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 254.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
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)

Vettore ►Sphere

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

Visualizza il vettore di riga o colonna nel formato sferico [$p \angle\theta \angle\phi$].

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.

Nota: Per forzare un risultato approssimativo,

Palmare: Premere **ctrl** **enter**.

Windows®: Premere **Ctrl+Invio**.

Macintosh®: Premere **⌘+Invio**.

iPad®: Tenere premuto **Invio** e selezionare .

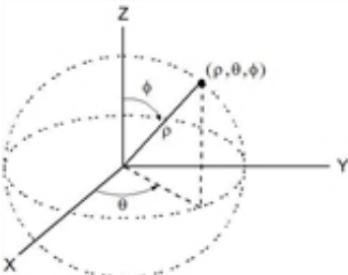
[1 2 3]►Sphere
[3.74166 \angle 1.10715 \angle 0.640522]

$\left(2 \angle \frac{\pi}{4} 3\right)$ ►Sphere
[3.60555 \angle 0.785398 \angle 0.588003]

Premere **enter**

$$\left(\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix} \right) \rightarrow \text{Sphere}$$

$$\left[\sqrt{13} & \angle \frac{\pi}{4} & \angle \sin^{-1} \left(\frac{2 \cdot \sqrt{13}}{13} \right) \right]$$



sqrt() (Radice quadrata)

sqrt(*Espr1*)⇒*espressione*

$$\sqrt{4} \quad 2$$

sqrt(*Listal*)⇒*lista*

$$\sqrt{\{9,a,4\}} \quad \{3,\sqrt{a},2\}$$

Restituisce la radice quadrata dell'argomento.

In una lista, restituisce le radici quadrate di tutti gli elementi di *Listal*.

Nota: vedere anche **Modello di radice quadrata**, pagina 5.

stat.results (Risultati dell'analisi statistica)

Catalogo >

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.

xlist:= {1,2,3,4,5}	{1,2,3,4,5}
ylist:= {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"	"{...}"

stat.values	"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.p̂	stat.Sx ²	stat.X̄1
stat.b10	stat.Fcol	stat.p̂1	stat.Sxy	stat.X̄2
stat.bList	stat.FInteract	stat.p̂2	stat.Sy	stat.X̄Diff
stat.χ ²	stat.FreqReg	stat.p̂Diff	stat.Sy 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])⇒espressione

In modalità angolo in radianti:

Restituisce la deviazione standard degli elementi di *Lista*.

$$\text{stDevPop}(\{a,b,c\})$$

$$\sqrt{\frac{2(a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}{3}}$$

Ogni elemento di *listaFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Lista*.

$$\text{stDevPop}(\{1,2,5,-6,3,-2\})$$

$$\sqrt{\frac{465}{6}}$$

Nota: *Lista* deve avere almeno due elementi. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 254.

$$\text{stDevPop}(\{1.3,2.5,-6.4\}, \{3,2,5\}) = 4.11107$$

stDevPop() (Deviazione standard della popolazione)

Catalogo >

stDevPop(*MatriceI*[,
matriceFreq]) \Rightarrow *matrice*

Restituisce un vettore di riga delle deviazioni standard della popolazione delle colonne di *MatriceI*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *MatriceI*.

Nota: *MatriceI* deve avere almeno due righe. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 254.

$$\begin{array}{c} \text{stDevPop} \begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{pmatrix} \begin{pmatrix} 4\sqrt{6} & \sqrt{78} & 2\sqrt{6} \\ 3 & 3 & 3 \end{pmatrix} \\ \hline \text{stDevPop} \begin{pmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{pmatrix} \begin{pmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{pmatrix} \\ \hline [2.52608 \quad 5.21506] \end{array}$$

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 254.

$$\begin{array}{c} \text{stDevSamp}\{\{a,b,c\}\} \\ \frac{\sqrt{3 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3} \\ \hline \text{stDevSamp}\{\{1,2,5,-6,3,-2\}\} \quad \frac{\sqrt{62}}{2} \\ \hline \text{stDevSamp}\{\{1,3,2,5,-6,4\}, \{3,2,5\}\} \quad 4.33345 \end{array}$$

stDevSamp(*MatriceI*[,
matriceFreq]) \Rightarrow *matrice*

Restituisce un vettore di riga delle deviazioni standard del campione delle colonne di *MatriceI*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *MatriceI*.

Nota: *MatriceI* deve avere almeno due righe. Gli elementi vuoti (nulli) vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 254.

$$\begin{array}{c} \text{stDevSamp} \begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{pmatrix} \begin{bmatrix} 4 & \sqrt{13} & 2 \end{bmatrix} \\ \hline \text{stDevSamp} \begin{pmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{pmatrix} \begin{pmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{pmatrix} \\ \hline [2.7005 \quad 5.44695] \end{array}$$

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>	<i>Done</i>
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 251.****string() (Da espressione a stringa)****Catalogo >**

string(*Espr*)⇒*stringa*

Semplifica *Espr* e restituisce il risultato come una stringa di caratteri.

<i>string(1.2345)</i>	"1.2345"
<i>string(1+2)</i>	"3"
<i>string(cos(x)+sqrt(3))</i>	"cos(x)+sqrt(3)"

subMat() (Sottomatrice)**Catalogo >**

subMat(*Matrice1*[, *rigaInizio*] [, *colInizio*] [, *rigaFine*] [, *collFine*])⇒*matrice*

Restituisce la sottomatrice specificata di *Matrice1*.

Impostazioni predefinite: *rigaInizio*=1, *colInizio*=1, *rigaFine*=ultima riga, *colFine*=ultima colonna.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
<i>subMat(m1,2,1,3,2)</i>	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
<i>subMat(m1,2,2)</i>	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)**Vedere Σ(), pagina 241.**

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 254.

sum(MatriceI[, Inizio[, Fine]])⇒matrice

Restituisce un vettore di riga contenente le somme degli elementi nelle colonne della *MatriceI*.

Inizio e *Fine* sono opzionali e specificano un intervallo di elementi.

Un argomento vuoto produce un risultato vuoto. Gli elementi vuoti (nulli) di *MatriceI* vengono ignorati. Per ulteriori informazioni sugli elementi vuoti, vedere a pagina 254.

sum({1,2,3,4,5})	15
sum({a,2·a,3·a})	6·a
sum(seq(n,n,1,10))	55
sum({1,3,5,7,9},3)	21

sumIf()

Catalogo >

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

sum({1 2 3 4 5 6})	[5 7 9]
sum({1 2 3 4 5 6 7 8 9})	[12 15 18]
sum({1 2 3 4 5 6 7 8 9},2,3)	[11 13 15]

sumIf({1,2,e,3,π,4,5,6},2.5<?<4.5)	e+π+7
sumIf({1,2,3,4},2<?<5,{10,20,30,40})	70

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 254.

Nota: vedere anche **countIf()**, pagina 42.

sumSeq()

Vedere $\Sigma()$, pagina 241

system() (Sistema)

Catalogo > 

system(*Espr1 [, Espr2 [, Espr3 [, ...]]]*)

`solve(` $\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y$ `)` $x=4$ and $y=-4$

system(*Equ1 [, Equ2 [, Equ3 [, ...]]]*)

Restituisce un sistema di equazioni, formattato come una lista. È inoltre possibile creare un sistema utilizzando un modello.

Nota: vedere anche **Sistema di equazioni**, pagina 7.

T (Trasposizione)**Catalogo >** *Matrice1***T** \Rightarrow *matrice*

Restituisce la trasposta dei complessi coniugati di *Matrice1*.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando **@t**.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$
$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\intercal$	$\begin{bmatrix} a & c \\ b & d \end{bmatrix}$
$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^\intercal$	$\begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$

tan() (Tangente)**Tasto** **tan**(*Espr1*) \Rightarrow *espressione***tan**(*Listal*) \Rightarrow *lista*

tan(*Espr1*) restituisce la tangente dell'argomento nella forma di espressione.

tan(*Listal*) restituisce una lista delle tangenti di tutti gli elementi di *Listal*.

Nota: l'argomento è interpretato come angolo in gradi, gradianti o radienti, a seconda della modalità angolo correntemente impostata. Si può usare **°**, **G** o **'** per escludere tale impostazione provvisoriamente.

In modalità angolo in gradi:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45)$	1
$\tan(\{0,60,90\})$	$\{0,\sqrt{3},\text{undef}\}$

In modalità angolo in gradianti (gradi centesimali):

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(50)$	1
$\tan(\{0,50,100\})$	$\{0,1,\text{undef}\}$

In modalità angolo in radianti:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45^\circ)$	1
$\tan\left(\left\{\pi, \frac{\pi}{3}, -\pi, \frac{\pi}{4}\right\}\right)$	$\{0,\sqrt{3},0,1\}$

In modalità angolo in radianti:

tan
(matriceQuadrata1) \Rightarrow *matriceQuadrata*

tan() (Tangente)

Tasto 

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.

$$\tan \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{pmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{pmatrix}$$

tan⁻¹() (Arcotangente)

Tasto 

tan⁻¹(Espr1)⇒espressione

tan⁻¹(List1)⇒lista

tan⁻¹(Espr1) restituisce nella forma di espressione l'angolo la cui tangente è Espr1.

tan⁻¹(List1) restituisce una lista dell'inversa della tangente di ciascun elemento di List1.

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 **arctan (...)**.

tan⁻¹

(matriceQuadrata1)⇒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.

In modalità angolo in gradi:

$$\tan^{-1}(1) = 45$$

In modalità angolo in gradi (gradi centesimali):

$$\tan^{-1}(1) = 50$$

In modalità angolo in radianti:

$$\tan^{-1}(\{0,0,2,0,0,5\}) = \{0,0.197396,0.463648\}$$

In modalità angolo in radianti:

$$\tan^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{pmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{pmatrix}$$

tangentLine() (Retta tangente)**Catalogo >** **tangentLine** $(Espr1,Var,Punto) \Rightarrow espressione$ **tangentLine** $(Espr1,Var=Punto) \Rightarrow espressione$

Restituisce la retta tangente alla curva rappresentata da *Espr1* in corrispondenza del punto specificato in *Var=Punto*.

Accertarsi che la variabile indipendente non sia definita. Per esempio, se $f1(x):=5$ e $x:=3$, allora **tangentLine(f1(x),x,2)** restituisce "false".

$\text{tangentLine}(x^2,x,1)$	$2 \cdot x - 1$
$\text{tangentLine}((x-3)^2-4,x=3)$	-4
$\text{tangentLine}\left(\frac{1}{x^3},x=0\right)$	$x=0$
$\text{tangentLine}(\sqrt{x^2-4},x=2)$	undef
$x:=3: \text{tangentLine}(x^2,x,1)$	5

tanh() (Tangente iperbolica)**Catalogo >** $\text{tanh}(Espr1) \Rightarrow espressione$ $\text{tanh}(Lista1) \Rightarrow lista$

$\text{tanh}(Espr1)$ restituisce nella forma di espressione la tangente iperbolica dell'argomento.

$\text{tanh}(Lista1)$ restituisce una lista delle tangenti iperboliche di ciascun elemento di *Lista1*.

tanh $(matriceQuadrata1) \Rightarrow matriceQuadrata$

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.

$\text{tanh}(1.2)$	0.833655
$\text{tanh}(\{0,1\})$	$\{0,\tanh(1)\}$

In modalità angolo in radianti:

$\text{tanh}\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$	$\begin{bmatrix} -0.079766 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$
---	---

tanh⁻¹() (Arcotangente iperbolica)**Catalogo >** $\text{tanh}^{-1}(Espr1) \Rightarrow espressione$ $\text{tanh}^{-1}(Lista1) \Rightarrow lista$

In modalità formato rettangolare complesso:

tanh⁻¹() (Arcotangente iperbolica)

Catalogo >

tanh⁻¹(Espr1) restituisce nella forma di espressione l'inversa della tangente iperbolica dell'argomento.

tanh⁻¹(List1) restituisce una lista dell'inversa della tangente iperbolica di ciascun elemento di *List1*.

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()**.

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

tanh ⁻¹ (0)	0
tanh ⁻¹ ({1,2,1,3})	{undef,0.518046-1.5708·i,ln(2)/2-π/2·i}

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

tanh ⁻¹ ($\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$)	[-0.099353+0.164058·i 0.267834-1.4908·i -0.087596-0.725533·i 0.479679-0.94730·i 0.511463-2.08316·i -0.878563+1.7901]
---	--

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

taylor()

Catalogo >

taylor(Espr1, Var, Ordine[, Punto])⇒espressione

Restituisce il polinomio di Taylor richiesto. Tale polinomio comprende i termini diversi da zero di grado intero compreso tra zero e *Ordine* in (*Var* meno *Punto*). **taylor()** restituisce la funzione stessa se non vi sono serie di potenze troncate di questo ordine, oppure se sono richiesti esponenti negativi o frazionari. Utilizzare la sostituzione e/o la moltiplicazione provvisoria per una potenza di (*Var* meno *Punto*) per determinare serie di potenze più generali.

Il valore predefinito di Punto è zero e corrisponde al punto di espansione.

taylor($e^{\sqrt{x}}$,x,2)	taylor($e^{\sqrt{x}}$,x,2,0)
taylor(e^t ,t,4) $t=\sqrt{x}$	$\frac{x^2}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1$
taylor($\frac{1}{x(x-1)}$,x,3)	taylor($\frac{1}{x(x-1)}$,x,3,0)
expand($\frac{\text{taylor}(\frac{x}{x(x-1)},x,4)}{x}$)	$-x^3 - x^2 - x - \frac{1}{x} - 1$

tCdf() (Probabilità di distribuzione t di Student)

Catalogo >

tCdf(*estremoInf*,*estremoSup*,*gl*)⇒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.

Per $P(X \leq valoreSuperiore)$, impostare
valoreInferiore = $-\infty$.

tCollect() (Riduzione trigonometrica)

Catalogo >

tCollect(*EsprI*)⇒espressione

Restituisce un'espressione nella quale i prodotti e le potenze intere dei seni e dei coseni vengono convertiti in una combinazione lineare di seni e coseni di angoli multipli, di somme e di differenze di angoli. La trasformazione converte i polinomi trigonometrici in combinazioni lineari delle rispettive armoniche.

A volte la funzione **tCollect()** permette di ottenere i risultati desiderati, non raggiunti mediante la semplificazione trigonometrica predefinita. **tCollect()** tende ad invertire le trasformazioni eseguite mediante **tExpand()**. A volte è possibile semplificare un'espressione mediante l'applicazione del comando **tExpand()** ad un risultato di **tCollect()**, o viceversa, in due fasi successive.

$$\begin{aligned} t\text{Collect}((\cos(\alpha))^2) &= \frac{\cos(2\cdot\alpha)+1}{2} \\ t\text{Collect}(\sin(\alpha)\cdot\cos(\beta)) &= \frac{\sin(\alpha-\beta)+\sin(\alpha+\beta)}{2} \end{aligned}$$

tExpand() (Espansione trigonometrica)

Catalogo >

tExpand(*EsprI*)⇒espressione

$$\begin{aligned} t\text{Expand}(\sin(3\cdot\phi)) &= 4\cdot\sin(\phi)\cdot(\cos(\phi))^2 - \sin(\phi) \\ t\text{Expand}(\cos(\alpha-\beta)) &= \cos(\alpha)\cdot\cos(\beta) + \sin(\alpha)\cdot\sin(\beta) \end{aligned}$$

Restituisce un'espressione nella quale vengono espansi i seni ed i coseni di angoli multipli di interi, di somme e di differenze di angoli. L'identità di $(\sin(x))^2 + (\cos(x))^2 = 1$ determina molti risultati possibili equivalenti. Pertanto un risultato può essere differente da quello riportato in altre pubblicazioni.

A volte la funzione **tExpand()** permette di ottenere i risultati desiderati, non raggiunti mediante la semplificazione trigonometrica predefinita. **tExpand()** tende ad invertire le trasformazioni eseguite mediante **tCollect()**. A volte è possibile semplificare un'espressione mediante l'applicazione del comando **tCollect()** ad un risultato di **tExpand()**, o viceversa, in due fasi successive.

Nota: la riduzione in scala nella modalità Gradi di $\pi/180$ influisce sulla capacità di **tExpand()** di riconoscere le forme da espandere. I migliori risultati si ottengono quando **tExpand()** viene utilizzata nella modalità di visualizzazione degli angoli in radianti.

Text

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.

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
```

Se il programma richiede di digitare una risposta, vedere **Request, pagina 158** oppure **RequestStr, pagina 160**.

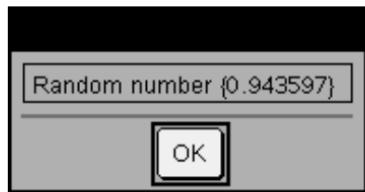
Nota: è possibile utilizzare questo comando all'interno di un programma definito dall'utente, ma non di una funzione.

EndPrgm

Eseguire il programma:

text_demo()

Esempio di finestra di dialogo:

**tInterval (Intervallo di confidenza t)****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 189).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

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

Variabile di output	Descrizione
stat.df	Gradi di libertà
stat.ox	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 *Lista1,Lista2[,Freq1
[,Freq2[,livelloConfidenza[,Aggregata]]]]*

(Input lista dati)

tInterval_2Samp *$\bar{x}_1,sx1,n1,\bar{x}_2,sx2,n2$
[,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 189).

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 254.

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.df	Gradi di libertà
stat. \bar{x}_1 , stat. \bar{x}_2	Medie dei campioni delle successioni di dati dalla distribuzione casuale normale
stat.ox1, stat.ox2	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.sp	Deviazione standard aggregata. Calcola quando <i>Aggregata</i> = sì.

tmpCnv() (Converti temperatura)

Catalogo > 

tmpCnv(Espr °unitàTempo, _ °unitàTempo2)

\Rightarrow espressione _ °unitàTempo2

Converte un valore relativo alla temperatura specificato da *Espr* da una unità di misura in un'altra. Le unità di misura valide per la temperatura sono:

_ °C Celsius

_ °F Fahrenheit

_ °K Kelvin

_ °R Rankine

Per inserire ° dalla tastiera, selezionarlo dai simboli del Catalogo.

Per inserire _ dalla tastiera, premere

 .

Ad esempio, 100_°C viene convertito in 212_°F.

Per convertire un intervallo di temperature, utilizzare invece ΔtmpCnv().

tmpCnv(100·_°C,_°F)	212·_°F
tmpCnv(32·_°F,_°C)	0·_°C
tmpCnv(0·_°C,_°K)	273.15·_°K
tmpCnv(0·_°F,_°R)	459.67·_°R

Nota: è possibile utilizzare il Catalogo per selezionare le unità di misura della temperatura.

ΔtmpCnv() (Converti un intervallo di temperature)

Catalogo > 

ΔtmpCnv(Espr °unitàTempo, _ °unitàTempo2) \Rightarrow espressione _ °unitàTempo2

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

Converte un intervallo di temperature (la differenza tra due valori di temperatura) specificato da *Espr* da una unità di misura in un'altra. Le unità di misura valide per la temperatura sono:

_ °C Celsius

_ °F Fahrenheit

Per inserire Δ dalla tastiera, selezionarlo dai simboli del Catalogo.

ΔtmpCnv(100·_°C,_°F)	180·_°F
ΔtmpCnv(180·_°F,_°C)	100·_°C
ΔtmpCnv(100·_°C,_°K)	100·_°K
ΔtmpCnv(100·_°F,_°R)	100·_°R
ΔtmpCnv(1·_°C,_°F)	1.8·_°F

Nota: è possibile utilizzare il Catalogo per selezionare le unità di misura della temperatura.

$\Delta\text{tmpCnv}()$ (Converti un intervallo di temperature)

Catalogo >

_°K Kelvin

_°R Rankine

Per inserire °, selezionarlo dalla tavolozza Simboli o digitare @d.

Per inserire _ dalla tastiera, premere

.

1_°C e 1_°K hanno grandezza uguale, come pure 1_°F e 1_°R. Tuttavia, la grandezza di 1_°C è 9/5 di quella di 1_°F.

Ad esempio, l'intervallo 100_°C (da 0_°C a 100_°C) equivale all'intervallo 180_°F.

Per convertire un particolare valore di temperatura invece di un intervallo, usare **tmpCnv()**.

tPdf() (Densità di probabilità t di Student)

Catalogo >

tPdf(*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 t di Student in corrispondenza di un valore *x* specificato con i gradi di libertà *gl* specificati.

trace() (Traccia)

Catalogo >

trace(*matriceQuadrata*)⇒espressione

Restituisce la traccia (somma di tutti gli elementi sulla diagonale principale) di *matriceQuadrata*.

trace $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	15
trace $\begin{bmatrix} a & 0 \\ 1 & a \end{bmatrix}$	$2 \cdot a$

Try*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 260.

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

Per vedere i comandi **Try**, **ClrErr** e **PassErr** in funzione, il programma *eigenvals()* riportato sulla destra. Avviare il programma eseguendo ciascuna delle seguenti espressioni.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

$$\text{eigenvals}\left(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

```
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
```

Nota: vedere anche **ClrErr**, pagina 31 e **PassErr**, pagina 140.

```
Define eigenvals(a,b)=Prgm
© Questo programma visualizza gli
autovalori di A·B
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 (Verifica t)**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 189).

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*=0Per $H_a: \mu > \mu0$, impostare *Ipotesi*>0

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

Variabile di output	Descrizione
stat.t	$(\bar{x} - \mu0) / (\text{stdev} / \sqrt{n})$
stat.PVal	Livello minimo di significatività in corrispondenza del quale l'ipotesi nulla può essere rifiutata

Variabile di output	Descrizione
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)

Catalogo > 

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 189).

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

Aggregata=0 non aggrega le varianze

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

Variabile di output	Descrizione
stat.t	Valore normale standard calcolato per la differenze delle medie

Variabile di output	Descrizione
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>Lista 1</i> e <i>Lista 2</i>
stat.sx1, stat.sx2	Deviazioni standard dei campioni delle sequenze di dati di <i>Lista 1</i> e <i>Lista 2</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,[PpY],[CpY],[PmtAt]*)⇒*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 209. Vedere anche **amortTbl()**, pagina 12.

tvmI()

Catalogo >

tvmI(*N,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*)⇒*valore*

tvmI(240,100000,-1000,0,0,12,12)

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 209. Vedere anche **amortTbl()**, pagina 12.

tvmN()

Catalogo >

tvmN(*I,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*)⇒*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 209. Vedere anche **amortTbl()**, pagina 12.

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 209. Vedere anche **amortTbl()**, pagina 12.

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 209. Vedere anche **amortTbl()**, pagina 12.

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
PpY	Rate all'anno, impostazione predefinita=1	Numero intero > 0
CpY	Periodi di capitalizzazione all'anno, impostazione predefinita=1	Numero intero > 0

Argomento TVM *	Descrizione	Tipo di dati
<i>PmtAt</i>	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 189).

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 254.

Variabile di output	Descrizione
stat. \bar{X}	Media dei valori X
stat. x	Somma dei valori X
stat. x2	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. \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 y
stat.MedY	Mediana di y
stat.Q ₃ Y	3° quartile di y
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)****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.**Catalogo >**

$\text{unitV}([a \ b \ c])$	$\begin{bmatrix} a \\ \sqrt{a^2+b^2+c^2} \end{bmatrix}$	$\begin{bmatrix} b \\ \sqrt{a^2+b^2+c^2} \end{bmatrix}$	$\begin{bmatrix} c \\ \sqrt{a^2+b^2+c^2} \end{bmatrix}$
$\text{unitV}([1 \ 2 \ 1])$	$\begin{bmatrix} \sqrt{6} \\ 6 \end{bmatrix}$	$\begin{bmatrix} \sqrt{6} \\ 3 \end{bmatrix}$	$\begin{bmatrix} \sqrt{6} \\ 6 \end{bmatrix}$
$\text{unitV}\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$	$\begin{bmatrix} \frac{\sqrt{14}}{14} \\ \frac{14}{14} \\ \frac{\sqrt{14}}{14} \end{bmatrix}$	$\begin{bmatrix} \frac{7}{14} \\ \frac{3\cdot\sqrt{14}}{14} \end{bmatrix}$	$\begin{bmatrix} \frac{14}{14} \\ \frac{14}{14} \\ \frac{7}{14} \end{bmatrix}$

Per vedere l'intero risultato, premere ,
quindi utilizzare e per spostare il cursore.**unLock****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 113 e **getLockInfo()**, pagina 89.**Catalogo >**

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

V**varPop() (Varianza della popolazione)****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*.**Catalogo >**

$\text{varPop}(\{5,10,15,20,25,30\})$	875
	12
<i>Ans</i> .1.	72.9167

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 254.

varSamp() (Varianza campione)

varSamp(Lista[, listaFreq])⇒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 254.

varSamp(Matrice1[, matriceFreq])⇒matrice

Restituisce un vettore riga contenente la varianza campione di ciascuna colonna di *Matrice1*.

Ogni elemento di *matriceFreq* conta il numero di occorrenze consecutive dell'elemento corrispondente di *Matrice1*.

Nota: *Matrice1* 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 254.

varSamp({{a,b,c}})

$$\frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$$

varSamp({{1,2,5,-6,3,-2}})

31

2

varSamp({{1,3,5},{4,6,2}})

68

33

varSamp({{1 2 5}, {-3 0 1}, {.5 .7 3}})

[4.75 1.03 4]

varSamp({{-1.1 2.2}, {3.4 5.1}, {-2.3 4.3}})

[6 3, 2 4, 5 1]

[3.91731 2.08411]

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 e premere 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*) \Rightarrow 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.

warnCodes(solve(sin(10·x)=\frac{x^2}{x},x),warn)
 $x=0.84232 \text{ or } x=-0.706817 \text{ or } x=-0.2852$
warn {10007,10009}

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

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 269.

when() (Quando)

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($x < 0, x + 3$) $x = 5$	undef
---------------------------------	-------

when($n > 0, n \cdot factorial(n - 1), 1$) → factorial(n)	Done
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* $1 \rightarrow i$ $0 \rightarrow tempsum$ While $i \leq n$ $tempsum + \frac{1}{i} \rightarrow tempsum$ $i+1 \rightarrow i$

EndWhile

Return *tempsum*

EndFunc

Done

*sum_of_recip(3)*116**X****xor**

BooleanExpr1 xor BooleanExpr2 restituisce
Boolean expression

true xor truefalse5>3 xor 3>5true

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 138.

Intero1 xor Intero2 \Rightarrow *intero*

In modalità base Esadecimale:

Importante: è zero, non la lettera O.

0h7AC36 xor 0h3D5F

0h79169

In modalità base Bin:

xor**Catalogo >**

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 22.

Nota: vedere **or**, pagina 138.

Z**zeros() (Zeri)****Catalogo >**

zeros(*Expr*, *Var*)⇒*lista*

zeros(*Expr*, *Var*=*Campione*)⇒*lista*

Restituisce una lista di valori reali possibili di *Var* che danno *Expr*=0. La funzione **zeros()** esegue questa operazione calcolando **exp>list(solve(*Expr*=0, *Var*), *Var*)**.

In alcuni casi, il risultato derivante dall'applicazione della funzione **zeros()** è preferibile a quello che si può ottenere mediante **solve()**. Tuttavia, il risultato di **zeros()** non può esprimere soluzioni implicite, soluzioni che richiedono disequazioni, oppure soluzioni che non implicano *Var*.

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.

zeros(*a*·*x*²+*b*·*x*+*c*,*x*)

$$\left\{ \frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a}, \frac{-\sqrt{b^2-4\cdot a\cdot c}+b}{2\cdot a} \right\}$$

a·*x*²+*b*·*x*+*c*|*x*=Ans[2] 0

exact(zeros(*a*·(e^{*x*}+*x*)·(sign(*x*)-1),*x*)) {[]}

exact(solve(*a*·(e^{*x*}+*x*)·(sign(*x*)-1)=0,*x*))

$$e^x+x=0 \text{ or } x>0 \text{ or } a=0$$

Nota: vedere anche `cSolve()`, `cZeros()` e `solve()`.

zeros({Espr1, Espr2}, {VarOCampione1, VarOCampione2 [, ...]})⇒matrice

Restituisce possibili zeri reali delle espressioni algebriche simultanee, dove ogni *VarOCampione* specifica una variabile di cui si cerca il valore.

In alternativa è possibile specificare un valore campione iniziale per la variabile. Ogni *VarOCampione* deve avere la forma:

variabile

– oppure –

variabile = numero reale o non reale

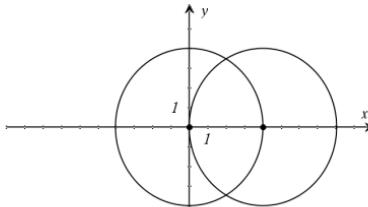
Ad esempio, x è valido come pure $x=3$.

Se tutte le espressioni sono polinomiali e NON si indica alcun valore campione iniziale, **zeros()** utilizza il metodo di eliminazione lessicale di Gröbner/Buchberger per determinare tutti gli zeri reali.

Supponiamo ad esempio di avere una circonferenza con raggio r all'origine e un'altra con raggio r con centro nel punto in cui la prima circonferenza interseca l'asse positivo x . Utilizzare **zeros()** per trovare le intersezioni.

Come indicato da r nell'esempio a destra, le espressioni polinomiali simultanee possono avere variabili aggiuntive senza valori, ma rappresentano valori numerici dati che potrebbero essere sostituiti successivamente.

Ciascuna riga della matrice risultante rappresenta uno zero alternativo, con i componenti ordinati come nella lista *varOCampione*. Per estrarre una riga, indicizzare la matrice per [riga].



$$\begin{aligned} \text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x,y\}\right) \\ \left[\begin{array}{cc} \frac{r}{2} & \frac{-\sqrt{3} \cdot r}{2} \\ \frac{r}{2} & \frac{2}{2} \\ \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \end{array} \right] \end{aligned}$$

Estrarre la riga 2:

$$\text{Ans}[2] \quad \left[\begin{array}{cc} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \end{array} \right]$$

zeros() (Zeri)

Catalogo >

È possibile, in aggiunta o in alternativa, includere incognite che non compaiono nelle espressioni. Ad esempio è possibile includere z quale incognita per estendere l'esempio precedente a due cilindri paralleli con raggio r che si intersecano. Gli zeri dei cilindri mostrano come le famiglie di zeri possano contenere costanti arbitrarie della forma ck, dove k è un suffisso intero compreso tra 1 e 255.

Per i sistemi polinomiali, i tempi di calcolo o l'insufficienza di memoria possono dipendere in gran parte dall'ordine in cui sono elencate le incognite. Se la scelta iniziale esaurisce la memoria (o la pazienza), provare a ridisporre le variabili all'interno delle espressioni e/o della lista *VarOCampione*.

Se non viene indicato alcun valore campione e se anche una sola espressione è non polinomiale per una qualsiasi variabile ma tutte le espressioni sono lineari per le incognite, **zeros()** utilizza l'eliminazione gaussiana per tentare di determinare tutte gli zeri reali.

Se un sistema non è né polinomiale per tutte le sue variabili né lineare per le incognite, **zeros()** determina al più uno zero utilizzando un metodo di approssimazione iterativa. A tale scopo, il numero di incognite deve essere uguale al numero di espressioni e tutte le altre variabili delle espressioni devono poter essere semplificate in numeri.

Ogni incognita inizia con il valore campione se ce n'è uno, altrimenti inizia con 0.0.

Utilizzare più valori campione per provare altri zeri uno alla volta. Per ottenere una convergenza, può essere necessario che il valore campione debba essere abbastanza prossimo a uno zero.

$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x,y,z\}\right)$$

$$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3} \cdot r}{2} & c1 \\ \frac{r}{2} & \frac{2}{\sqrt{3} \cdot r} & c1 \end{bmatrix}$$

$$\text{zeros}\left(\left\{x+e^z \cdot y - 1, x - y - \sin(z)\right\}, \{x,y\}\right)$$

$$\begin{bmatrix} \frac{e^z \cdot \sin(z) + 1}{e^z + 1} & \frac{-(\sin(z) - 1)}{e^z + 1} \end{bmatrix}$$

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \{y,z\}\right)$$

$$\begin{bmatrix} 0.041458 & 3.18306 \\ 0.001871 & 6.28131 \\ 4.76 \cdot 10^{-11} & 1796.99 \\ 2 \cdot 10^{-13} & 254.469 \end{bmatrix}$$

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \{y,z=2 \cdot \pi\}\right)$$

$$\begin{bmatrix} 0.001871 & 6.28131 \end{bmatrix}$$

zInterval (Intervallo di confidenza Z)

Catalogo >

zInterval σ , *Lista*[, *Freq*[, *livelloConfidenza*]]

zInterval (Intervallo di confidenza Z)

Catalogo >

(Input lista dati)

zInterval $\sigma, \bar{x}, n [,livelloConfidenza]$

(Input statistiche riepilogo)

Calcola un intervallo di confidenza z . Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189)

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

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 189).

x è un intero non negativo.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

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

Variabile di output	Descrizione
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]**

Calcola l'intervallo di confidenza z per due proporzioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189).

$x1$ e $x2$ sono interi non negativi.

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

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)

zInterval_2Samp (Intervallo di confidenza z su due campioni)

Catalogo > 

Calcola l'intervallo di confidenza z su due campioni. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189).

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

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. σ_x1 , stat. σ_x2	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>Lista 1</i> e <i>Lista 2</i>

zTest (Verifica z)

Catalogo > 

zTest $\mu0,\sigma,Lista,[Freq,Ipotesi]$

(Input lista dati)

zTest $\mu0,\sigma,\bar{x},n,[,Ipotesi]$

(Input statistiche riepilogo)

Esegue una verifica z con frequenza *listaFreq*. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189).

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 254.

Variabile di output	Descrizione
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$
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)

zTest_1Prop *p0,x,n[Ipotesi]*

Esegue una verifica *z* per una proporzione. Il riepilogo dei risultati è memorizzato nella variabile *stat.results*. (pagina 189).

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 254.

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 189).

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 $H_a: p < p0$, impostare *Ipotesi<0*

Per informazioni sull'effetto di elementi vuoti in una lista, vedere "Elementi vuoti (nulli)", pagina 254.

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

Variabile di output	Descrizione
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 189).

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 254.

Variabile di output	Descrizione
stat.z	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. \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)

Espr1 + Espr2⇒espressione

Restituisce la somma dei due argomenti.

Tasto	+
56	56
56+4	60
60+4	64
64+4	68
68+4	72

Listal + Listal2⇒lista

Matrice1 + Matrice2⇒matrice

Restituisce una lista (o matrice) contenente le somme degli elementi corrispondenti di *Listal1* e *Listal2* (oppure di *Matrice1* e *Matrice2*).

Gli argomenti devono avere uguali dimensioni.

Espr + Listal1⇒lista

Listal1 + Espr⇒lista

Restituisce una lista contenente le somme di *Espr* e di ciascun elemento di *Listal1*.

Espr + Matrice1⇒matrice

Matrice1 + Espr⇒matrice

Restituisce una matrice nella quale *Espr* viene sommata a ciascun elemento sulla diagonale di *Matrice1*. *Matrice1* deve essere quadrata.

Nota: utilizzare *.+* (punto più) per aggiungere un'espressione a ciascun elemento.

$\left\{ 22,\pi, \frac{\pi}{2} \right\} \rightarrow l1$	$\left\{ 22,\pi, \frac{\pi}{2} \right\}$
$\left\{ 10,5, \frac{\pi}{2} \right\} \rightarrow l2$	$\left\{ 10,5, \frac{\pi}{2} \right\}$
$l1+l2$	$\{ 32,\pi+5,\pi \}$
$Ans+\{ \pi,-5,-\pi \}$	$\{ \pi+32,\pi,0 \}$
$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$
$15+\{ 10,15,20 \}$	$\{ 25,30,35 \}$
$\{ 10,15,20 \}+15$	$\{ 25,30,35 \}$

$20+\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
---	--

- (sottrazione)

Tasto -

Espr1 - Espr2⇒espressione

Restituisce *Espr1* meno *Espr2*.

6-2	4
$\pi - \frac{\pi}{6}$	$\frac{5\cdot\pi}{6}$

-(sottrazione)**Tasto** *Listal - Listal2⇒lista**Matrice1 - Matrice2⇒matrice*

Sottrae ciascun elemento di *Listal2* (o di *Matrice2*) dal corrispondente elemento di *Listal1* (o di *Matrice1*) e restituisce il risultato.

Gli argomenti devono avere uguali dimensioni.

*Espr - Listal1⇒lista**Listal1 - Espr⇒lista*

Sottrae ciascun elemento di *Listal1* da *Espr* oppure sottrae *Espr* da ciascun elemento di *Listal1*, quindi restituisce una lista contenente i risultati.

*Espr - Matrice1⇒matrice**Matrice1 - Espr⇒matrice*

Espr - Matrice1 restituisce una matrice di numero volte *Espr* la matrice di identità, meno *Matrice1*. *Matrice1* deve essere quadrata.

Matrice1 - Espr restituisce una matrice di numero volte *Espr* la matrice di identità, sottratta da *Matrice1*. *Matrice1* deve essere quadrata.

Nota: Utilizzare **.** (punto meno) per sottrarre un'espressione da ciascun elemento.

$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10,5, \frac{\pi}{2} \right\}$	$\{ 12, \pi - 5, 0 \}$
$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 2 \end{bmatrix}$

$15 - \{ 10, 15, 20 \}$	$\{ 5, 0, -5 \}$
$\{ 10, 15, 20 \} - 15$	$\{ -5, 0, 5 \}$

$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$
---	--

·(moltiplicazione)**Tasto** *Espr1 · Espr2⇒espressione*

Restituisce il prodotto di due argomenti.

Listal1 · Listal2⇒lista

Restituisce una lista contenente i prodotti degli elementi corrispondenti di *Listal1* e *Listal2*.

Le liste devono avere uguali dimensioni.

$2 \cdot 3 \cdot 45$	6.9
$x \cdot y \cdot x$	$x^2 \cdot y$

$\{ 1, 2, 3 \} \cdot \{ 4, 5, 6 \}$	$\{ 4, 10, 18 \}$
$\left[\frac{2}{a}, \frac{3}{2} \right] \cdot \left[a^2, \frac{b}{3} \right]$	$\left[2 \cdot a, \frac{b}{2} \right]$

·(moltiplicazione)

Tasto

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*.

Espr · *Listal* ⇒ *lista*

Listal · *Espr* ⇒ *lista*

Restituisce una lista contenente i prodotti di *Espr* per ciascun elemento di *Listal*.

Espr · *Matrice1* ⇒ *matrice*

Matrice1 · *Espr* ⇒ *matrice*

Restituisce una lista contenente i prodotti di *Espr* per ciascun elemento di *Matrice1*.

Nota: Utilizzare . ·(punto moltiplicazione) per moltiplicare un'espressione per ciascun elemento.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} = \begin{bmatrix} a+2\cdot b+3\cdot c & d+2\cdot e+3\cdot f \\ 4\cdot a+5\cdot b+6\cdot c & 4\cdot d+5\cdot e+6\cdot f \end{bmatrix}$$

/ (divisione)

Tasto

Espr1 / *Espr2* ⇒ *espressione*

Restituisce il quoziente di *Espr1* diviso per *Espr2*.

Nota: vedere anche **Modello di frazione**, pagina 5.

Listal / *Listal2* ⇒ *lista*

Restituisce una lista contenente i quozienti di *Listal* diviso per *Listal2*.

Le liste devono avere uguali dimensioni.

Espr / *Listal* ⇒ *lista*

Listal / *Espr* ⇒ *lista*

Restituisce una lista contenente i quozienti di *Espr* diviso per *Listal* o quelli di *Listal* diviso per *Espr*.

$$\begin{array}{ll} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 & \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix} \\ \text{1·identity}(3) & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$\begin{array}{ll} \frac{2}{3.45} & .57971 \\ \frac{x^3}{x} & x^2 \end{array}$$

$$\begin{array}{ll} \left\{ \frac{1}{4}, \frac{2}{5}, \frac{3}{6} \right\} & \left\{ 0.25, \frac{2}{5}, \frac{1}{2} \right\} \end{array}$$

$$\begin{array}{ll} \frac{a}{\{3,a,\sqrt{a}\}} & \left\{ \frac{a}{3}, 1, \sqrt{a} \right\} \\ \frac{\{a,b,c\}}{a\cdot b\cdot c} & \left\{ \frac{1}{b\cdot c}, \frac{1}{a\cdot c}, \frac{1}{a\cdot b} \right\} \end{array}$$

/ (divisione)

Tasto

Matrice1 / Espr \Rightarrow matrice

Restituisce una matrice contenente i quozienti di *Matrice1 / Espr*.

$$\frac{\begin{bmatrix} a & b & c \\ \end{bmatrix}}{a \cdot b \cdot c} \quad \begin{bmatrix} \frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b} \\ \end{bmatrix}$$

Nota: utilizzare . / (punto divisione) per dividere un'espressione per ciascun elemento.

\wedge (elevamento a potenza)

Tasto

Espr1 ^ Espr2 \Rightarrow espressione

Listal ^ Lista2 \Rightarrow lista

$$\frac{4^2}{\{a,2,c\}^{\{1,b,3\}}} \quad \begin{bmatrix} 16 \\ a,2^b,c^3 \end{bmatrix}$$

Restituisce il primo argomento elevato alla potenza del secondo argomento.

Nota: vedere anche **Modello di esponente**, pagina 5.

In una lista, restituisce gli elementi di *Listal* 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.

Espr ^ Lista1 \Rightarrow lista

$$\frac{p^{\{a,2,-3\}}}{\left\{ p^a, p^2, \frac{1}{p^3} \right\}}$$

Restituisce *Espr* elevata alla potenza degli elementi di *Lista1*.

Lista1 ^ Espr \Rightarrow lista

$$\frac{\{1,2,3,4\}^{-2}}{\left\{ 1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16} \right\}}$$

Restituisce gli elementi di *Listal* elevati alla potenza di *espressione*.

matriceQuadrata1 ^ intero \Rightarrow matrice

$$\frac{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2}{\begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}}$$

Restituisce *matriceQuadrata1* elevata alla potenza di *intero*.

matriceQuadrata1 deve essere una matrice quadrata.

$$\frac{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1}}{\begin{bmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{bmatrix}}$$

Se *intero* = -1, viene calcolata l'inversa della matrice.

$$\frac{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2}}{\begin{bmatrix} \frac{11}{4} & -\frac{5}{4} \\ 2 & 2 \\ -\frac{15}{4} & \frac{7}{4} \end{bmatrix}}$$

\wedge (elevamento a potenza)

Tasto

Se $intero < -1$, viene calcolata l'inversa della matrice a una potenza positiva appropriata.

x^2 (quadrato)

Tasto

*Espr1*² \Rightarrow espressione

Restituisce il quadrato dell'argomento.

Listal $\wedge 2$ \Rightarrow lista

Restituisce una lista contenente i quadrati degli elementi di *Listal*.

*matriceQuadrata1*² \Rightarrow matrice

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.

4^2	16
$\{2,4,6\}^2$	$\{4,16,36\}$
$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$	$\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$
$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} .^2$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$

$\cdot+$ (punto addizione)

Tasti

Matrice1 $\cdot+$ *Matrice2* \Rightarrow matrice

Espr $\cdot+$ *Matrice1* \Rightarrow matrice

Matrice1 $\cdot+$ *Matrice2* restituisce una matrice che è la somma di ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

Espr $\cdot+$ *Matrice1* restituisce una matrice che è la somma di *Espr* e di ciascun elemento di *Matrice1*.

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} .+\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$
$x .+\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$

$\cdot-$ (punto sottrazione).

Tasti

Matrice1 $\cdot-$ *Matrice2* \Rightarrow matrice

Espr $\cdot-$ *Matrice1* \Rightarrow matrice

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} .-\begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$
$x .-\begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$

. - (punto sottrazione).

Tasti

Matrice1 .- Matrice2 restituisce una matrice che è la differenza tra ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

Espr .- Matrice1 restituisce una matrice che è la differenza di *Espr* e di ciascun elemento di *Matrice1*.

. · (punto moltiplicazione)

Tasti

Matrice1 .· Matrice2 \Rightarrow matrice

Espr .· Matrice1 \Rightarrow matrice

Matrice1 .· Matrice2 restituisce una matrice che è il prodotto di ciascuna coppia di elementi corrispondenti di *Matrice1* e *Matrice2*.

Espr .· Matrice1 restituisce una matrice contenente i prodotti di *Espr* per ciascun elemento di *Matrice1*.

$$\begin{array}{l} \left[\begin{matrix} a & 2 \\ b & 3 \end{matrix} \right] \cdot \left[\begin{matrix} c & 4 \\ 5 & d \end{matrix} \right] \\ \hline x \cdot \left[\begin{matrix} a & b \\ c & d \end{matrix} \right] \end{array} \quad \begin{array}{l} \left[\begin{matrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{matrix} \right] \\ \hline \left[\begin{matrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{matrix} \right] \end{array}$$

. / (punto divisione)

Tasti

Matrice1 . / Matrice2 \Rightarrow matrice

Espr . / Matrice1 \Rightarrow matrice

Matrice1 . / Matrice2 restituisce una matrice che è il quoziente di ciascuna coppia di elementi corrispondenti di *Matrice1* e di *Matrice2*.

Espr . / Matrice1 restituisce una matrice che è il quoziente di *Espr* diviso per ciascun elemento di *Matrice1*.

$$\begin{array}{l} \left[\begin{matrix} a & 2 \\ b & 3 \end{matrix} \right] \cdot \left[\begin{matrix} c & 4 \\ 5 & d \end{matrix} \right] \\ \hline x \cdot \left[\begin{matrix} c & 4 \\ 5 & d \end{matrix} \right] \end{array} \quad \begin{array}{l} \left[\begin{matrix} a & 1 \\ c & 2 \\ b & 3 \\ 5 & d \end{matrix} \right] \\ \hline \left[\begin{matrix} x & x \\ c & 4 \\ x & x \\ 5 & d \end{matrix} \right] \end{array}$$

.^ (punto elevato a potenza)

Tasto

Matrice1 .^ Matrice2 \Rightarrow matrice

Espr .^ Matrice1 \Rightarrow matrice

$$\begin{array}{l} \left[\begin{matrix} a & 2 \\ b & 3 \end{matrix} \right] \cdot \left[\begin{matrix} c & 4 \\ 5 & d \end{matrix} \right] \\ \hline x \cdot \left[\begin{matrix} c & 4 \\ 5 & d \end{matrix} \right] \end{array} \quad \begin{array}{l} \left[\begin{matrix} a^c & 16 \\ b^5 & 3^d \end{matrix} \right] \\ \hline \left[\begin{matrix} x^c & x^4 \\ x^5 & x^d \end{matrix} \right] \end{array}$$

= (uguale)

Tasto

$Espr1 = Espr2 \Rightarrow$ espressione booleana

$List1 = Lista2 \Rightarrow$ lista booleana

$Matrice1 = Matrice2 \Rightarrow$ 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.

Esempio di funzione che usa gli operatori relazionali: $=$, \neq , $<$, \leq , $>$, \geq

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

If $x \leq -5$ 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

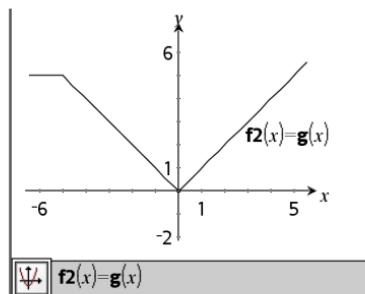
Return 3

EndIf

EndFunc

Done

Risultato della rappresentazione grafica $g(x)$



\neq (diverso)

Tasti

$Espr1 \neq Espr2 \Rightarrow$ espressione booleana

Vedere l'esempio di " $=$ " (uguale).

$List1 \neq Lista2 \Rightarrow$ 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).

$List1 < Lista2 \Rightarrow$ 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).

$List1 \leq Lista2 \Rightarrow$ 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).

$List1 > Lista2 \Rightarrow$ 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).

$List1 \geq Lista2 \Rightarrow$ 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 =>

\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 <=>

! (fattoriale)**Tasto** *Espr1!* \Rightarrow espressione

$5!$	120
$\{\{5,4,3\}\}!$	$\{120,24,6\}$
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}!$	$\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

Listal! \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.

& (aggiunge)**Tasti** *Stringa1 & Stringa2* \Rightarrow stringa

"Hello " & "Nick"

"Hello Nick"

Restituisce una stringa di testo, aggiungendo *Stringa2* a *Stringa1*.**d() (derivata)****Catalogo** > *d(**Espr1, Var[, Ordine]**)* \Rightarrow espressione

$$\frac{d}{dx}(f(x) \cdot g(x)) = \frac{d}{dx}(f(x)) \cdot g(x) + \frac{d}{dx}(g(x)) \cdot f(x)$$

*d(**Listal, Var[, Ordine]**)* \Rightarrow lista

$$\frac{d}{dy}\left(\frac{d}{dx}(x^2 \cdot y^3)\right) = 6 \cdot y^2 \cdot x$$

*d(**Matrice1, Var[, Ordine]**)* \Rightarrow matrice

$$\frac{d}{dx}\left(\begin{bmatrix} x^2 & x^3 & x^4 \end{bmatrix}\right) = \begin{bmatrix} 2 \cdot x & 3 \cdot x^2 & 4 \cdot x^3 \end{bmatrix}$$

Restituisce la derivata prima in funzione della variabile *Var*.*Ordine*, se incluso, deve essere un numero intero. Se ordine è minore di zero, il risultato è una primitiva.**Nota:** è possibile inserire questa funzione dalla tastiera del computer digitando **derivative (...)**.**d()** non segue il normale meccanismo di calcolo che prevede la semplificazione di tutti gli argomenti e la successiva applicazione delle definizioni di funzioni a tali argomenti completamente semplificati. **d()** esegue invece la procedura seguente:

1. Semplifica il secondo argomento solo se non porta a una non variabile.
2. Semplifica il primo argomento solo se non richiama nessun valore

memorizzato per la variabile determinata al punto 1.

3. Determina la derivata simbolica del risultato del punto 2 relativamente alla variabile di cui al punto 1.

Se la variabile dalla fase 1 ha un valore memorizzato o un valore specificato mediante un operatore (“|”) di vincolo, sostituire tale valore nel risultato dalla fase 3.

Nota: Vedere anche **Derivata prima**, pagina 9, **Derivata seconda**, pagina 10 o **Derivata n-esima**, pagina 10.

J() (integrale)

$\int(Espr1, Var[, Inferiore, Superiore]) \Rightarrow espressione$

$\int(Espr1, Var[, Costante]) \Rightarrow espressione$

Restituisce l'integrale di *Espr1* rispetto alla variabile *Var* da *Inferiore* a *Superiore*.

Nota: vedere anche **Modello di integrale definito o indefinito**, pagina 10.

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

Restituisce una primitiva se *Inferiore* e *Superiore* vengono omessi. Una costante simbolica di integrazione viene omessa se non si fornisce l'argomento della *Costante*.

$$\int_a^b x^2 dx = \frac{b^3}{3} - \frac{a^3}{3}$$

$$\int(a \cdot x^2, x, c) = \frac{a \cdot x^3}{3} + c$$

Primitive ugualmente valide possono differire per una costante numerica che può essere nascosta, soprattutto quando una primitiva contiene logaritmi o funzioni trigonometriche inverse. Inoltre, a volte vengono aggiunte espressioni piecewise costanti in modo che una primitiva possa essere valida per un intervallo maggiore rispetto alla formula tradizionale.

La funzione $\int()$ restituisce se stessa per quelle parti di *Espr1* che non può determinare come combinazione finita esplicita delle funzioni interne e degli operatori.

Quando si forniscono sia *Inferiore* sia *Superiore*, viene effettuato il tentativo di individuare eventuali discontinuità o derivate discontinue nell'intervallo *Inferiore* < *Var* < *Superiore* e di suddividere l'intervallo in tali punti.

Nell'impostazione Auto della modalità **Auto** o **Approssimato** l'integrazione numerica viene utilizzata, per quanto possibile, quando non si può determinare una primitiva o un limite

Nell'impostazione Approssimato, viene subito eseguito un tentativo di integrazione numerica, se possibile. Le primitive vengono cercate solamente qualora tale integrazione numerica non sia applicabile o dia esiti negativi.

$$\int b \cdot e^{-x^2} + \frac{a}{x^2+a^2} dx = b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$$

Nota: Per forzare un risultato approssimativo,

Palmare: Premere **ctrl enter**.

Windows®: Premere **Ctrl+Invio**.

Macintosh®: Premere **⌘+Invio**.

iPad®: Tenere premuto **Invio** e selezionare **≈**.

$$\int_{-1}^1 e^{-x^2} dx = 1.49365$$

$\int()$ (integrale)

Catalogo >

La funzione $\int()$ può essere nidificata per eseguire integrali multipli. I limiti di integrazione possono dipendere da variabili di integrazione esterne ad essi.

Nota: vedere anche **nint()**, pagina 131.

$$\int_0^a \int_0^x \ln(x+y) dy dx$$

$$\frac{a^2 \cdot \ln(a)}{2} + \frac{a^2 \cdot (4 \cdot \ln(2) - 3)}{4}$$

$\sqrt()$ (radice quadrata)

Tasti

$\sqrt{Espr1} \Rightarrow espressione$

$$\sqrt{4} \quad 2$$

$\sqrt{List1} \Rightarrow lista$

$$\sqrt{\{9,a,4\}} \quad \{3,\sqrt{a},2\}$$

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 5.

$\prod()$ (prodSeq)

Catalogo >

$\prod(Var, Basso, Alto) \Rightarrow espressione$

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

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

$$\prod_{k=1}^n (k^2) \quad (n!)^2$$

Calcola *Espr1* per ciascun valore di *Var* da *Basso* ad *Alto* e restituisce il prodotto dei risultati.

$$\prod_{n=1}^5 \left(\left\{\frac{1}{n}, n, 2\right\}\right) \quad \left\{\frac{1}{120}, 120, 32\right\}$$

Nota: vedere anche **Modello di prodotto** (\prod), pagina 9.

$$\prod_{k=4}^3 (k) \quad 1$$

$\prod(Var, Basso, Basso-1) \Rightarrow 1$

$\prod(Var, Basso, Alto) \Rightarrow 1 / \prod(Var, Alto+1, Basso-1)$ if *Alto* < *Basso-1*

$\Pi()$ (prodSeq)

Catalogo >

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.

$$\prod_{k=4}^1 \left(\frac{1}{k} \right)$$

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) \cdot \prod_{k=2}^4 \left(\frac{1}{k} \right)$$

6

4

$\Sigma()$ (sumSeq)

Catalogo >

$\Sigma(Espr1, Var, Bass0, Alto) \Rightarrow$ espressione

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

Calcola *Espr1* per ciascun valore di *Var* da *Bass0* ad *Alto* e restituisce la somma dei risultati.

Nota: vedere anche **Modello di somma**, pagina 9.

$\Sigma(Espr1, Var, Bass0, Bass0-1) \Rightarrow 0$

$\Sigma(Espr1, Var, Bass0, Alto) \Rightarrow -\Sigma(Espr1, Var, Alto+1, Bass0-1)$ if *Alto* < *Bass0-1*

$$\sum_{n=1}^5 \left(\frac{1}{n} \right)$$

$$\sum_{k=1}^n \left(k^2 \right)$$

$$\sum_{n=1}^{\infty} \left(\frac{1}{n^2} \right)$$

$$\sum_{k=4}^3 \left(k \right)$$

137
60 $\frac{n \cdot (n+1) \cdot (2 \cdot n + 1)}{6}$

6

6

6

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.

$$\sum_{k=4}^1 \left(k \right)$$

$$\sum_{k=4}^1 \left(k \right) + \sum_{k=2}^4 \left(k \right)$$

-5

4

$\Sigma\text{Int()}$

Catalogo >

$\Sigma\text{Int}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [valoreArrotondato]) \Rightarrow$ valore

 $\Sigma\text{Int}(1,3,12,4.75,20000,,12,12)$

-213.48

ΣInt

$\Sigma\text{Int}()$

Catalogo >

(
 $NPmt1$,
 $NPmt2$,
 $\text{tabellaAmmortamento}$) \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 209.

- 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.

ΣInt

($NPmt1$,
 $NPmt2$,
 $\text{tabellaAmmortamento}$)
calcola la somma degli interessi sulla base della tabella di ammortamento
 $\text{tabellaAmmortamento}$. L'argomento
 $\text{tabellaAmmortamento}$ deve essere una matrice avente la forma descritta in
 $\text{amortTbl}()$, pagina 12.

Nota: vedere anche $\Sigma\text{Prn}()$, sotto, e $\text{Bal}()$, pagina 21.

$tbl:=\text{amortTbl}(12,12,4.75,20000,,12,12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\text{Int}(1,3,tbl)$

-213.48

$\Sigma\text{Prn}()$

Catalogo >

$\Sigma\text{Prn}(NPmt1, NPmt2, N, I, PV, [Pmt],$
 $[FV], [PpY], [CpY], [PmtAt],$
 $[\text{valoreArrotondato}])\Rightarrow$ valore

$\Sigma\text{Prn}(1,3,12,4.75,20000,,12,12)$

-4916.28

ΣPrn

(
 $NPmt1$,
 $NPmt2$,
 $\text{tabellaAmmortamento}$) \Rightarrow 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 209.

- 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
(*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 12.

Nota: vedere anche **ΣInt()**, sopra, e **Bal()**, pagina 21.

tbl:=amortTbl([12,12,4.75,20000,,12,12])

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

ΣPrn(1,3,tbl) -4916.28

(conversione indiretta)

Tasti  

stringaNomeVariabile

#("x"&"y"&"z")

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.

Crea o fa riferimento alla variabile *xyz*.

10→r	10
"r"→sI	"r"
#sI	10

Restituisce il valore della variabile (r) il cui nome è memorizzato nella variabile s1.

E (notazione scientifica)***mantissaEesponente***

Permette di inserire un numero in notazione scientifica. Il numero viene interpretato come *mantissa* × 10^{*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.

Tasto 

23000.	23000.
2300000000.+4.1E15	4.1E15
3·10 ⁴	30000

g (gradanti)***Espr1g* ⇒ *espressione*****Tasto **

In modalità di misurazione degli angoli in gradi, gradianti o radianti:

$\cos(50^\circ)$	$\frac{\sqrt{2}}{2}$
$\cos(\{0,100^\circ,200^\circ\})$	{1,0,-1}

Espr1g* ⇒ *espressione***Listalg* ⇒ *lista******Matrice1g* ⇒ *matrice***

Questa funzione consente di utilizzare un angolo espresso in gradianti 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 gradianti, restituisce *Espr1* senza modifiche.

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando @g.

r (radianti)

Tasto

Espr1 $r \Rightarrow$ espressione

Listal $r \Rightarrow$ lista

Matrice l $r \Rightarrow$ matrice

Questa funzione consente di utilizzare un angolo espresso in radianti pur essendo attiva la modalità di misurazione degli angoli in gradi o in radianti.

In modalità angolo in gradi, moltiplica l'argomento per $180/\pi$.

In modalità angolo in radianti, restituisce l'argomento immutato.

In modalità angolo in gradi, 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.

$^\circ$ (gradi)

Tasto

Espr1 $^\circ \Rightarrow$ espressione

Listal $^\circ \Rightarrow$ lista

Matrice l $^\circ \Rightarrow$ 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 $10/9$.

In modalità angolo in gradi, gradi o radianti:

$$\cos\left(\frac{\pi}{4^r}\right) = \frac{\sqrt{2}}{2}$$

$$\cos\left(0^r, \frac{\pi}{12}, -\{\pi\}^r\right) = \left\{1, \frac{(\sqrt{3}+1)\cdot\sqrt{2}}{4}, -1\right\}$$

In modalità angolo in gradi, gradi o radianti:

$$\cos(45^\circ) = \frac{\sqrt{2}}{2}$$

In modalità angolo in radianti:

Nota: Per forzare un risultato approssimativo,

Palmare: Premere **ctrl enter**.

Windows®: Premere **Ctrl+Invio**.

Macintosh®: Premere **⌘+Invio**.

iPad®: Tenere premuto **Invio** e selezionare .

° (gradi)

Tasto

Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando @d.

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) \\ \{1., 0.707107, 0., 0.864976\}$$

°, ', " (gradi/primi/secondi)

Tasti

gg°pp'ss.ss"⇒espressione

In modalità angolo in gradi:

gg un numero positivo o negativo

25°13'17.5"

25.2215

pp un numero non negativo

25°30'

51

2

ss.ss un 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 ("").

∠ (angolo)

Tasti

[Raggio,∠θ_Angolo]⇒vettore

In modalità angolo in radianti e con il formato vettore impostata su:

(formato polare)

rettangolare

[Raggio,∠θ_Angolo,Z_Coordinate]⇒vettore

$$[5 \angle 60^\circ \angle 45^\circ] \begin{bmatrix} 5\sqrt{2} & 5\sqrt{6} & 5\sqrt{2} \\ 4 & 4 & 2 \end{bmatrix}$$

(formato cilindrico)

[Raggio,∠θ_Angolo,∠θ_Angolo]⇒vettore

cilindrico

(formato sferico)

$$[5 \angle 60^\circ \angle 45^\circ] \begin{bmatrix} \frac{5\sqrt{2}}{2} & \angle \frac{\pi}{3} & \frac{5\sqrt{2}}{2} \end{bmatrix}$$

Restituisce le coordinate nella forma di un vettore a seconda dell'impostazione della modalità formato vettore: rettangolare, cilindrico o sferico.

sferico

\angle (angolo)

Tasti ctrl



Nota: è possibile inserire questo simbolo dalla tastiera del computer digitando $\text{@}<$.

$[5 \angle 60^\circ \angle 45^\circ]$

$[5 \angle \frac{\pi}{3} \angle \frac{\pi}{4}]$

(Grandezza \angle Angolo) \Rightarrow valoreComplesso
(formato polare)

Inserisce un valore complesso in formato polare ($r\angle\theta$). *Angolo* viene interpretato conformemente alla modalità di misurazione degli angoli impostata.

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

$5+3\cdot i\left(10 \angle \frac{\pi}{4}\right) \quad 5-5\cdot \sqrt{2}+(3-5\cdot \sqrt{2})\cdot i$

Nota: Per forzare un risultato approssimativo,

Palmare: Premere ctrl enter.

Windows®: Premere Ctrl+Invio.

Macintosh®: Premere ⌘+Invio.

iPad®: Tenere premuto **Invio** e selezionare ≈.

$5+3\cdot i\left(10 \angle \frac{\pi}{4}\right) \quad -2.07107-4.07107\cdot i$

' (primo)

Tasto ?!

variabile'

variabile''

Inserisce il simbolo di primo in un'equazione differenziale. Un simbolo di primo indica un'equazione differenziale del primo ordine, due simboli di primo ne indicano una del secondo ordine e così via.

deSolve $y''=y$ and $y(0)=0$ and $y'(0)=0, t,y$
$$\frac{3}{2 \cdot y^{\frac{4}{3}}}=t$$

_ (trattino basso come elemento vuoto)

Vedere “Elementi vuoti (nulli)”,
pagina 254.

_ (trattino basso come designatore di unità)

Tasti ctrl _

Espr_Unità

$3\cdot \underline{\text{m}} \blacktriangleright \underline{\text{ft}}$

$9.84252\cdot \underline{\text{ft}}$

Indica le unità di misura di un'*Espr*. Tutti i nomi delle unità di misura devono iniziare con un trattino basso.

(trattino basso come designatore di unità)

Tasti

È possibile utilizzare unità predefinite oppure crearne di proprie. Per un elenco delle unità predefinite, aprire il Catalogo e visualizzare la scheda Conversione unità di misura. È possibile selezionare i nomi delle unità di misura dal Catalogo oppure digitarli direttamente dalla tastiera.

Variabile_

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.

Se a *Variabile* è associato un valore, il carattere *_* viene ignorato e *Variabile* conserverà il tipo di dati originale.

Nota: è possibile memorizzare un numero complesso in una variabile senza utilizzare il carattere *_*. Tuttavia, per ottenere i migliori risultati in calcoli come **cSolve()** e **cZeros()**, è consigliabile usare il carattere *_* (trattino basso).

Nota: il simbolo di conversione, \blacktriangleright , si trova nel Catalogo. Fare clic su , quindi su **Operatori matematici**.

Assumendo che *z* non sia definito:

real(<i>z</i>)	<i>z</i>
real(<i>z_</i>)	real(<i>z_</i>)
imag(<i>z</i>)	0
imag(<i>z_</i>)	imag(<i>z_</i>)

\blacktriangleright (converti)

Tasti

Espr_Unità1 \blacktriangleright *_Unità2* \Rightarrow *Espr_Unità2*

3·_m \blacktriangleright ft 9.84252·_ft

Converte un'espressione da un'unità di misura ad un'altra.

Il carattere “*_*” indica le unità di misura. Le unità di misura devono appartenere alla stessa categoria, come ad esempio Lunghezza o Area.

Per un elenco delle unità predefinite, aprire il Catalogo e visualizzare la scheda Conversione unità di misura:

- È possibile selezionare un nome di unità dall'elenco.
- È possibile selezionare l'operatore di conversione, \blacktriangleright , dalla cima dell'elenco.

In alternativa, è possibile digitare i nomi delle unità manualmente. Per digitare il carattere “_” quando si inseriscono manualmente i nomi di unità nel palmare, premere .

Nota: per convertire le unità di misura della temperatura, utilizzare **tmpCnv()** e **ΔtmpCnv()**. L'operatore di conversione ► non gestisce le unità di misura della temperatura.

10^()**Catalogo >**

10^ (Espr1)⇒espressione

$$10^{1.5} \quad 31.6228$$

10^ (Listal)⇒lista

$$10^{\{0,-2,2,a\}} \quad \left\{1, \frac{1}{100}, 100, 10^a\right\}$$

Restituisce 10 elevato alla potenza dell'argomento.

In una lista, restituisce 10 elevato alla potenza degli elementi di *Listal*.

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)**Catalogo >**

Espr1 ^-1⇒espressione

$$(3.1)^{-1} \quad 0.322581$$

Listal ^-1⇒lista

$$\{a, 4, -0.1, x, -2\}^{-1} \quad \left\{\frac{1}{a}, \frac{1}{4}, -10, \frac{1}{x}, \frac{-1}{2}\right\}$$

Restituisce il reciproco dell'argomento.

In una lista, restituisce i reciproci degli elementi di *Listal*.

matriceQuadrata1 $\wedge^{-1} \Rightarrow$ matriceQuadrata

Restituisce l'inversa di matriceQuadrata1.

matriceQuadrata1 deve essere una matrice quadrata non singolare.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{-2}{a-2} & \frac{1}{a-2} \\ \frac{a}{2 \cdot (a-2)} & \frac{-1}{2 \cdot (a-2)} \end{bmatrix}$$

| (operatore di vincolo)

tasti

Expr | BooleanExpr1

$x+1|x=3$ 4

[andBooleanExpr2]...

$x+y|x=\sin(y)$ $\sin(y)+y$

Expr | BooleanExpr1 [orBooleanExpr2]...

$x+y|\sin(y)=x$ $x+y$

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.

$x^3-2 \cdot x+7 \rightarrow f(x)$ Done

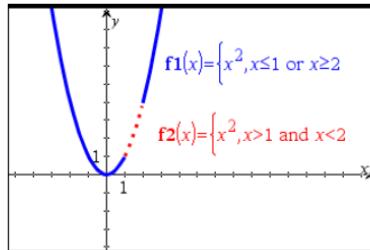
$f(x)|x=\sqrt{3}$ $\sqrt{3}+7$

$(\sin(x))^2+2 \cdot \sin(x)-6|\sin(x)=d$ $d^2+2 \cdot d-6$

$\text{solve}(x^2-1=0,x)|x>0 \text{ and } x<2$ x=1

$\sqrt{x} \cdot \sqrt{\frac{1}{x}}|x>0$ 1

$\sqrt{x} \cdot \sqrt{\frac{1}{x}}$ $\sqrt{\frac{1}{x}} \cdot \sqrt{x}$



Le esclusioni utilizzano l'operatore relazionale "diverso da" (\neq or \neq) per escludere dalla valutazione un valore specifico. Vengono impiegate principalmente per escludere una soluzione esatta usando **cSolve()**, **cZero()**, **fMax()**, **fMin()**, **solve()**, **zero()**, ecc.

`solve(x^2 - 1 = 0, x)|x ≠ 1` $x = -1$

→ (memorizza)

Tasti ctrl var

Espr → Var

$\frac{\pi}{4} \rightarrow myvar$ $\frac{\pi}{4}$

Lista → Var

$2 \cdot \cos(x) \rightarrow y1(x)$ Done

Matrice → Var

$\{1, 2, 3, 4\} \rightarrow lst5$ $\{1, 2, 3, 4\}$

Espr → Funzione(Param1,...)

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow matg$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
---	--

Lista → Funzione(Param1,...)

"Hello" → str1 "Hello"

Matrice → Funzione(Param1,...)

Se la variabile *Var* non esiste, *Var* viene creata e inizializzata in *Espr*, *Lista* o *Matrice*.

Se la variabile *Var* esiste già e non è bloccata o protetta, l'operatore ne sostituisce il contenuto con *Espr*, *Lista* o *Matrice*.

Suggerimento: se si prevede di eseguire calcoli simbolici mediante variabili non definite, si consiglia di evitare di memorizzare qualsiasi cosa con variabili di una sola lettera comunemente utilizzate, quali a, b, c, x, y, z, ecc.

Nota: è possibile inserire questo operatore dalla tastiera del computer digitando =: come scelta rapida. Per esempio, digitare **pi/4 =: Miavar.**

:= (assegna)*Var := Expr**Var := Lista**Var := Matrice**Funzione(Param1,...) := Expr**Funzione(Param1,...) := Lista**Funzione(Param1,...) := Matrice*

Se la variabile *Var* non esiste, *Var* viene creata e inizializzata in *Expr*, *Lista* o *Matrice*.

Se la variabile *Var* esiste già e non è bloccata o protetta, l'operatore ne sostituisce il contenuto con *Expr*, *Lista* o *Matrice*.

Suggerimento: se si prevede di eseguire calcoli simbolici mediante variabili non definite, si consiglia di evitare di memorizzare qualsiasi cosa con variabili di una sola lettera comunemente utilizzate, quali a, b, c, x, y, z, ecc.

Tasti ctrl int₂

<i>myvar:=</i>	$\frac{\pi}{4}$
----------------	-----------------

<i>yI(x):=</i>	$2 \cdot \cos(x)$	<i>Done</i>
----------------	-------------------	-------------

<i>lst5:=</i>	$\{1, 2, 3, 4\}$	$\{1, 2, 3, 4\}$
---------------	------------------	------------------

<i>matg:=</i>	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
---------------	--	--

<i>str1:="Hello"</i>	"Hello"
----------------------	---------

© [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

Tasti , tasti

0b numeroBinario

In modalità base Dec:

0b10+0hF+10

27

0h numeroEsadecimale

In modalità base Bin:

0b10+0hF+10

0b11011

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 Esadecimale:

0b10+0hF+10

0h1B

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™ CAS 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 57 e **isVoid()**, pagina 101.

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.

$[-]$	=
$\text{gcd}(100, -)$	=
$3 + -$	=
$\{5, -, 10\} - \{3, 6, 9\}$	$\{2, -, 1\}$

Argomenti di lista contenenti 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 statistici.

$\text{sum}(\{2, -, 3, 5, 6, 6\})$	16.6
$\text{median}(\{1, 2, -, -, -, 3\})$	2
$\text{cumulativeSum}(\{1, 2, -, 4, 5\})$	$\{1, 3, -, 7, 12\}$
$\text{cumulativeSum}\begin{bmatrix} 1 & 2 \\ 3 & - \\ 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 \\ 4 & - \\ 9 & 8 \end{bmatrix}$

SortA e **SortD** spostano in fondo tutti gli elementi vuoti del primo argomento.

$\{5, 4, 3, -, 1\} \rightarrow \text{list1}$	$\{5, 4, 3, -, 1\}$
$\{5, 4, 3, 2, 1\} \rightarrow \text{list2}$	$\{5, 4, 3, 2, 1\}$
SortA $\text{list1}, \text{list2}$	Done
list1	$\{1, 3, 4, 5, -\}$
list2	$\{1, 3, 4, 5, 2\}$

Argomenti di lista contenenti elementi vuoti

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
$list1$	$\{5,3,2,1,_$
$list2$	$\{5,3,2,1,4\}$
$l1:=\{1,2,3,4,5\}; l2:=\{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
LinRegMx $l1, l2$	Done
$stat.Resid$	$\{0.434286,_, -0.862857, -0.011429, 0.44\}$
$stat.XReg$	$\{1,_, 3, 4, 5,\}$
$stat.YReg$	$\{2,_, 3, 5, 6, 6\}$
$stat.FreqReg$	$\{1,_, 1, 1, 1,\}$

Una categoria omessa nelle regressioni introduce un elemento vuoto corrispondente nel residuo.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
$cat:=\{"M", "M", "F", "F"\}; incl:=\{"F"\}$	$\{"F"\}$
LinRegMx $l1, l2, 1, cat, incl$	Done
$stat.Resid$	$\{_, _, 0, 0, _\}$
$stat.XReg$	$\{_, _, 4, 5, _\}$
$stat.YReg$	$\{_, _, 5, 6, 6\}$
$stat.FreqReg$	$\{_, _, 1, 1, _\}$

Una frequenza 0 nelle regressioni introduce un elemento vuoto corrispondente nel residuo.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx $l1, l2, \{1, 0, 1, 1\}$	Done
$stat.Resid$	$\{0.069231, _, -0.276923, 0.207692\}$
$stat.XReg$	$\{1, _, 4, 5, _\}$
$stat.YReg$	$\{2, _, 5, 6, 6\}$
$stat.FreqReg$	$\{1, _, 1, 1, _\}$

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	pi
θ	theta
∞	infinity
\leq	<=
\geq	>=
\neq	/=
\Rightarrow (implicazione logica)	=>
\Leftrightarrow (doppia implicazione logica, XNOR)	<=>
\rightarrow (memorizza operatore)	=:
$ $ (valore assoluto)	abs(...) (Valore assoluto)
$\sqrt()$	sqrt(...) (Radice quadrata)
d()	derivative (...)
$\int()$	integral (...) (Integrale)
$\Sigma()$ (Modello di somma)	sumSeq(...)
$\Pi()$ (Modello di prodotto)	prodSeq(...)
$\sin^{-1}()$, $\cos^{-1}()$, ...	arcsin(...) , arccos(...) , ...
Δ Lista()	deltaList(...)
Δ tmpCnv() (Converti un intervallo di temperature)	deltaTmpCnv(...)

Dalla tastiera del computer

Per inserire questo:	digitare questa scelta rapida:
c1, c2, ... (costanti)	@c1, @c2, ...

Per inserire questo:	digitare questa scelta rapida:
n1, n2, ... (costanti valori interi)	<code>@n1, @n2, ...</code>
i (unità immaginaria)	<code>@i</code>
e (logaritmo naturale in base e)	<code>@e</code>
E (notazione scientifica)	<code>@E</code>
T (trasposizione)	<code>@t</code>
r (radiani)	<code>@r</code>
° (Gradi)	<code>@d</code>
G gradianti	<code>@g</code>
∠ (angolo)	<code>@<</code>
► (conversione)	<code>@></code>
►Decimal, ►approxFraction (), ecc.	<code>@>Decimal, @>approxFraction(), ecc.</code>

EOS™ (Equation Operating System) gerarchia

Questa sezione descrive il sistema EOS™ (Equation Operating System) utilizzato da TI-Nspire™ CAS 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 $/=$), minore di (<), minore di o uguale a (\leq o \leqslant), maggiore di (>), maggiore di o uguale a (\geq o \geqslant)
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™ CAS 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.

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 205).

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 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 virgolette 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	Interruzione È stato premuto il tasto [esc] o [fn on] durante un calcolo lungo o durante l'esecuzione del programma.
190	Definizione circolare 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.
200	Condizione non valida 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."
210	Tipo di dati non valido Un argomento è un tipo di dati sbagliato.
220	Limite dipendente
230	Dimensione 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.
235	Errore di dimensione. Elementi insufficienti nelle liste.
240	Dimensioni non corrispondenti 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.
250	Divisione per zero
260	Argomento errato Deve essere presente un argomento in un dominio specificato. Ad esempio <code>rand(0)</code> non è valido.
270	Nome di variabile duplicato
280	Else ed Elself non validi fuori dal blocco If..Endif
290	EndTry non trova la corrispondente istruzione Else
295	Numero eccessivo di iterazioni

Codice errore	Descrizione
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.
565	Non valido fuori da un programma
570	Nome di percorso non valido

Codice errore	Descrizione
	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 diagonalizzabile
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 }
730	Manca inizio o fine del blocco
740	Manca Then nel blocco If..EndIf
750	Il nome non è una funzione o un programma

Codice errore	Descrizione
765	Nessuna funzione selezionata
780	Nessuna soluzione trovata
800	Risultato non reale Ad esempio, se il software è impostato su Real, $\sqrt{(-1)}$ non è valido. Per consentire risultati complessi, modificare l'impostazione di modalità "Real or Complex" in RETTANGOLARE (RECTANGULAR) o POLARE (POLAR).
830	Superamento della memoria
850	Programma non trovato Impossibile trovare una chiamata di programma all'interno di un altro programma nel percorso specificato durante l'esecuzione.
855	Funzione di tipo Rand non consentita nel grafico
860	Ricorsione troppo profonda
870	Nome riservato o variabile di sistema
900	Argomento errato Impossibile applicare al set di dati il modello mediana-mediana.
910	Errore di sintassi
920	Testo non trovato
930	Argomenti mancanti Nella funzione o nel comando mancano uno o più argomenti.
940	Troppi argomenti L'espressione o equazione contiene un numero eccessivo di argomenti e non può essere calcolata.
950	Troppi indici
955	Troppe variabili non definite
960	La variabile non è definita Nessun valore assegnato alla variabile. Utilizzare uno dei seguenti comandi: <ul style="list-style-type: none">• sto \rightarrow• :=• Define per assegnare valori alle variabili.

Codice errore	Descrizione
965	SO privo di licenza
970	Variabile in uso, di conseguenza non sono consentite chiamate e modifiche
980	La variabile è protetta
990	Nome variabile non valido Accertarsi che il nome non superi i limiti di lunghezza
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	La funzione non è definita. Utilizzare uno dei seguenti comandi: <ul style="list-style-type: none">• Define• :=• sto \rightarrow per definire una funzione.
1100	Calcolo non reale Ad esempio, se il software è impostato su "Real" $\sqrt{(-1)}$ non è valido. 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

Codice errore	Descrizione
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 sto \rightarrow. • 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 MieLibrerie. • Aggiornare le librerie. <p>Per ulteriori informazioni, vedere il capitolo Libreria nella documentazione.</p>
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.

Codice errore	Descrizione
	<ul style="list-style-type: none"> • 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>
1290	<p>Argomento errato</p> <p>Utilizzare un polinomio in una sola variabile.</p>
1300	<p>Argomento errato</p> <p>I coefficienti del polinomio devono dare come risultato valori numerici.</p>

Codice errore	Descrizione
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 funzione **warnCodes()** 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 dei codici di avvertenza, vedere **warnCodes()**, pagina 214.

Codice di avvertenza	Messaggio
10000	L'operazione può introdurre soluzioni false
10001	Derivare un'equazione può produrre un'equazione falsa
10002	Soluzione dubbia
10003	Accuratezza dubbia
10004	L'operazione può far perdere soluzioni
10005	cSolve può specificare più zeri
10006	Solve può specificare più zeri
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>
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 undef^0 sostituito da 1
10014	undef^0 sostituito da 1
10015	1^∞ o 1^undef sostituito da 1
10016	1^undef 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

Codice di avvertenza	Messaggio
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 identica.
10024	Risultato ottenuto con approssimazione aritmetica
10025	L'equivalenza non può essere verificata in modalità EXACT (Esatta).
10026	Il vincolo potrebbe essere ignorato. Specificare il vincolo nella forma "\' Variable MathTestSymbol Constant' o da un insieme di queste due forme, per esempio 'x<3 and x>12'

Informazioni generali

Informazioni sul servizio di manutenzione e riparazione del prodotto TI e sulla garanzia

Informazioni sul prodotto e sui servizi TI Per ulteriori informazioni sui prodotti e servizi TI, potete contattare TI via e-mail o visiti l'indirizzo Internet di TI.

Indirizzo e-mail: ti-cares@ti.com

Indirizzo internet: education.ti.com

Informazioni sul servizio di manutenzione e riparazione e sulla garanzia Per informazioni sulla durata e le condizioni della garanzia o sul servizio di manutenzione e riparazione del prodotto, fate riferimento alla dichiarazione di garanzia allegata al presente prodotto oppure contattate il vostro rivenditore/distributore Texas Instruments locale.

Indice

-	
-, sottrazione[*]	226
!	
!, fattoriale	237
"	
", notazione in secondi	246
#	
#, conversione indiretta	243
#, operatore conversione indiretta	259
%	
%, percentuale	232
&	
&, aggiunge	237
*	
*, moltiplicazione	227
,	
, notazione in primi	246
, primo	247
.	
.-, punto sottrazione	230
.*, punto moltiplicazione	231
./, punto divisione	231
.^, punto elevato a potenza	231
.+, punto addizione	230
:	
:=, assegna	252

^	
\wedge^{-1} , reciproco	249
\wedge , elevamento a potenza	229
-	
\lrcorner designazione di unità	247
, operatore di vincolo	250
+	
+ ₁ , addizione	226
/	
/ ₁ , divisione[*]	228
=	
\neq , diverso[*]	233
= ₁ , uguale	233
>	
> ₁ , maggiore di	235
\Pi	
\Pi ₁ , prodotto[*]	240
\Sigma	
$\Sigma()$, somma[*]	241
\Int()	241
\Prn()	242
\vee	
\vee ₁ , radice quadrata[*]	240
\int	
\int ₁ , integrale[*]	238

≤	
≤, minore di o uguale a	234
≥	
≥, maggiore di o uguale a	235
▶	
►, converti in angolo in gradi[Grad]	93
►, converti unità[*]	248
►approxFraction()	18
►Base10, visualizza come numero decimale[Base10]	24
►Base16, visualizza come esadecimale[Base16]	24
►Base2, visualizza come binario[Base2]	22
►cos, Visualizza rispetto al coseno[cos]	36
►Cylind, visualizza come vettore in forma cilindrica[Cylind]	50
►DD, visualizza angolo decimale[DD]	54
►Decimal, visualizza il risultato nella forma decimale[Decimal]	54
►DMS, visualizza come gradi/primi/secondi[DMS]	62
►exp, vis. rispetto ad e[exp]	72
►Polar, visualizza come vettore polare[Polare]	142
►Rad, converti in radianti[Rad]	153
►Rect, visualizza come vettore rettangolare[Rect]	156
►sin, visualizza rispetto al seno[sin (Seno)]	178
►Sphere, visualizza come vettore sferico[Sphere]	187
⇒	
⇒, implicazione logica[*]	236, 256
→	
→, memorizza	251
↔	
↔, doppia implicazione logica[*]	236
©	
©, commento	253
°	
°, gradi/primi/secondi[*]	246
°, notazione in gradi[*]	245

0	
0b, indicatore binario	253
0h, indicatore esadecimale	253
1	
10^(), potenza di dieci	249
A	
abs(), valore assoluto	12
addizione, +	226
affianca/concatena, augment()	20
aggiunge, &	237
all'interno della stringa, inString()	97
altro, Else	94
amortTbl(), tabella di ammortamento	12, 21
and, operatore boolean	13
angle(), angolo	14
angolo, angle()	14
ANOVA, analisi della varianza a una variabile	15
ANOVA2way, analisi della varianza a due dimensioni	15
Ans, ultimo risultato	17
approssima, approx()	18
approx(), approssima	18
approxRational()	18
arccos()	19
arccosh()	19
arccot()	19
arccoth()	19
arccsc()	19
arccsch()	19
arcLen(), lunghezza arco	19
arcocoseno, $\cos^{-1}()$	38
arcoseno, $\sin^{-1}()$	179
arcotangente, $\tan^{-1}()$	196
arcsec()	19
arcsech()	20
arcsin()	20
arcsinh()	20
arctan()	20
arctanh()	20
argomenti nelle funzioni TVM	209

argomenti TVM	209
arrotondamento, round()	164
arrotondato per difetto, floor()	80
arrotondato per eccesso, ceiling()	26, 43
augment(), affianca(concatena)	20
autovalore, eigVl()	67
autovettore, eigVc()	66
avgRC(), tasso di variazione media	21

B

binario	
indicatore, 0b	253
visualizza, ►Base2	22
binomCdf() (Funzione della probabilità cumulativa per la distribuzione binomiale)	25
binomPdf()	25
blocco di variabile e gruppi di variabili	113

C

calcola polinomio, polyEval()	144
calcolo, ordine di	258
campione casuale	155
cancella	
errore, ClrErr	31
caratteri	
codice di carattere, ord()	139
stringa, char()	28
casuale	
matrice, randMat()	154
normale, randNorm()	154
polinomio, randPoly()	155
seme numero, RandSeed	155
Cdf()	78
ceiling(), arrotondato per eccesso	26
centralDiff()	26
cFactor(), fattore complesso	27
char(), stringa di caratteri	28
charPoly() (Polinomio caratteristico)	28
χ^2 2way	29
ClearAZ	31
ClrErr, cancella errore	31
codici di errore e messaggi	269
colAugment	32

colDim(), dimensione colonna di matrice	32
colNorm(), norma colonna di matrice	32
comando Stop	192
Comando Wait	214
combinazioni, nCr()	128
comDenom(), denominatore comune	33
commento, ©	253
complesso	
coniugato, conj()	35
fattore, cFactor()	27
risolvi, cSolve()	45
zeri, cZeros()	51
completeSquare(), complete square	34
con, 	250
conj(), complesso coniugato	35
constructMat(), Costruisci matrice	35
conteggio condizionato elementi in una lista, countif()	42
conteggio elementi in una lista, count()	42
conteggio giorni tra le date, dbd()	53
conversione indiretta, #	243
converti	
4Grad	93
4Rad	153
unità	248
coordinata x rettangolare, P►Rx()	139
coordinata y rettangolare, P►Ry()	140
copia variabile o funzione, CopyVar	35
corrMat(), matrice di correlazione	36
cos ⁻¹ , arcocoseno	38
cos(), coseno	37
coseno	
visualizza l'espressione rispetto a	36
coseno, cos()	37
cosh ⁻¹ (), arcocoseno iperbolico	39
cosh(), coseno iperbolico	39
costante	
in solve()	183
costanti	
in cSolve()	47
in cZeros()	52
in deSolve()	58
in solve()	185
tasti di scelta rapida per	256

Costruisci matrice, constructMat()	35
cot ⁻¹ (), arcocotangente	40
cot(), cotangente	40
cotangente, cot()	40
coth ⁻¹ (), arcocotangente iperbolica	41
coth(), cotangente iperbolica	41
count(), conteggio elementi in una lista	42
countif(), conteggio condizionato elementi in una lista	42
cPolyRoots()	43
crossP(), prodotto vettoriale	43
csc ⁻¹ (), cosecante inversa	44
csc(), cosecante	44
csch ⁻¹ (), cosecante iperbolica inversa	45
csch(), cosecante iperbolica	44
cSolve(), risovi in campo complesso	45
CubicReg, regressione cubica	48
cumulativeSum(), somma cumulativa	49
Cycle, ripeti	50
cZeros(), zeri complessi	51

D

d(), prima derivata	237
da lista a matrice, list►mat()	110
da matrice a lista, mat►list()	118
dbd(), giorni tra le date	53
decimale	
visualizza angolo, ►DD	54
visualizza intero, ►Base10	24
Define	54
Define LibPriv	56
Define LibPub	56
Define, definisci	54
definisci, Define	54
definizione	
funzione o programma privato	56
programma o funzione pubblica	56
deltaList()	57
deltaTmpCnv()	57
DelVar, elimina variabile	57
delVoid(), rimuovi elementi vuoti	57
denominatore	33
denominatore comune, comDenom()	33
densità di probabilità t di Student, tPdf()	204

densità di probabilità, normPdf()	133
derivata implicita, Impdif()	96
derivata o derivata ennesima	
modello di	10
derivata prima	
modello di	9
derivata seconda	
modello di	10
derivata()	58
derivate	
derivata numerica, nDeriv()	130
derivata numerica, nDerivative()	129
prima derivata, d()	237
deSolve(), soluzione di equazioni differenziali	58
destra, right()	161
det(), determinante matrice	60
deviazione standard, stdDev()	190-191, 212
diag(), diagonale matrice	61
dim(), dimensione	61
dimensione, dim()	61
Disp, visualizza dati	62
disposizioni semplici, nPr()	134
distribuzione normale cumulativa inversa (invNorm())	99
diverso, ≠	233
divisione intera, intDiv()	97
divisione, P	228
domain(), dominio della funzione	63
dominantTerm(), termine dominante	63
dominio della funzione, domain()	63
doppia implicazione logica, ⇔	236
dot	
prodotto, dotP()	65
dotP(), prodotto scalare	65

E

e alla potenza, e^()	72
E, esponente	244
e, vis. espressione rispetto a	72
e^(), funzione esponenziale	65
eff(), converti da tasso nominale a effettivo	66
eigVc(), autovettore	66
eigVL(), autovalore	67
elementi vuoti	254

elementi vuoti (nulli)	254
elementi vuoti, rimuovi	57
elevamento a potenza, ^	229
elimina	
elementi vuoti dalla lista	57
eliminazione	
variabile, DelVar	57
else if, Elseif	67
Elseif, else if	67
end	
loop, EndLoop	117
per, EndFor	81
se, EndIf	94
EndTry, fine tentativo	205
EndWhile, fine mentre	216
EOS (Equation Operating System)	258
Equation Operating System (EOS)	258
equazioni simultanee, simult()	176
errori e soluzione dei problemi	
cancella errore, ClrErr	31
passa errore, PassErr	140
esadecimale	
indicatore, Oh	253
visualizza, ►Base16	24
esatto, exact()	71
esci, Exit	71
esclusione con l'operatore " "	250
espandi, expand()	73
espansione trigonometrica, tExpand()	199
esponente, E	244
esponenti	
modello di	5
espressioni	
da espressione a lista, exp►list()	73
da stringa a espressione, expr()	75, 114
etichetta, Lbl	101
euler(), Euler function	68
exact(), esatto	71
Exit, esci	71
exp(), e alla potenza	72
exp►list(), da espressione a lista	73
expand(), espandi	73
expr(), da stringa a espressione	75, 114
ExpReg, regressione esponenziale	75

F

F Test su due campioni	84
factor(), fattorizza	76
fattoriale, !	237
fattorizza, factor()	76
Fill, riempimatrice	78
fine	
funzione, EndFunc	85
mentre, EndWhile	216
tentativo, EndTry	205
fine funzione, EndFunc	85
fine loop, EndLoop	117
fine mentre, EndWhile	216
fine se, EndIf	94
FiveNumSummary	79
floor(), arrotondato per difetto	80
fMax(), massimo funzione	80
fMin(), minimo funzione	81
For	81
For, per	81
forma a scalini per righe, ref()	157
forma a scalini ridotta per righe, rref()	166
format(), stringa formato	82
fpart(), funzione parte frazionaria	83
frazione propria, propFrac	148
frazioni	
frzProp	148
modello di	5
freqTable()	83
frequency()	84
Func, funzione	85
Func, funzione programma	85
funzione esponenziale e	
modello di	6
funzione esponenziale, e^()	65
funzione parte intera, int()	97
funzione piecewise a 2 tratti	
modello di	6
funzione piecewise a N tratti	
modello di	7
funzioni	
definite dall'utente	54
funzione programma, Func	85

massimo, fMax()	80
minimo, fMin()	81
parte frazionaria, fpart()	83
funzioni definite dall'utente	54
funzioni di distribuzione	
binomCdf() (Funzione della probabilità cumulativa per la distribuzione binomiale)	25
binomPdf()	25
invNorm()	99
invt()	99
Invχ ² () (Funzione di probabilità chi ² cumulativa inversa)	98
normCdf()	133
normPdf()	133
poissCdf()	141
poissPdf()	142
tCdf()	199
tPdf()	204
χ ² 2way()	29
χ ² Cdf()	29
χ ² GOF()	30
χ ² Pdf()	31
funzioni e programmi definiti dall'utente	56
funzioni e variabili	
copia	35
funzioni finanziarie, tvmFV()	208
funzioni finanziarie, tvml()	208
funzioni finanziarie, tvmN()	208
funzioni finanziarie, tvmPmt()	209
funzioni finanziarie, tvmPV()	209

G

G, gradianti	244
gcd(), massimo comun divisore	86
geomCdf()	87
geomPdf()	87
Get	87
getDenom(), ottieni/restituisci denominatore	88
getLangInfo(), ottieni/restituisci informazioni sulla lingua	89
getLockInfo(), testa lo stato bloccato di una variabile o di un gruppo di variabili	89
getMode(), impostazioni di modo	90
getNum(), ottieni/restituisci numeratore	91
GetStr	91
getType(), get type of variable	91
getVarInfo(), ottieni/restituisci informazioni variabile	92

giorni tra le date, <code>dbd()</code>	53
<code>Goto, vai a</code>	93
gruppi, blocco e sblocco	113, 212
gruppi, test stato di blocco	89

I

<code>identity()</code> , matrice identità	93
<code>If, se</code>	94
<code>ifFn()</code>	95
<code>imag()</code> , parte immaginaria	96
<code>ImpDif()</code> , derivata implicita	96
implicazione logica, \Rightarrow	236, 256
imposta	
modo, <code>setMode()</code>	172
impostazioni di modo, <code>getMode()</code>	90
impostazioni, visualizza correnti	90
in mezzo alla stringa, <code>mid()</code>	121
indice interno di rendimento modificato, <code>mirr()</code>	123
<code>input, Input</code>	96
<code>Input, input</code>	96
<code>inString()</code> , all'interno della stringa	97
<code>int()</code> , funzione parte intera	97
<code>intDiv()</code> , divisione intera	97
integrale definito	
modello di	10
integrale indefinito	
modello di	10
integrale, %o	238
<code>interpolate(), interpolate</code>	98
inversa, \wedge^{-1}	249
<code>invF()</code>	99
<code>invNorm()</code> , distribuzione normale cumulativa inversa	99
<code>invt()</code>	99
<code>Invx^2()</code>	98
<code>iPart()</code> , parte intera	99
iperbolico	
arcocoseno, $\cosh^{-1}()$	39
arcoseno, $\sinh^{-1}()$	180
arcotangente, $\tanh^{-1}()$	197
coseno, $\cosh()$	39
seno, $\sinh()$	180
tangente, $\tanh()$	197

irr(), indice di rendimento interno	100
indice di rendimento interno, irr()	100
isPrime(), verifica numero primo	100
isVoid(), test per nullità	101

L

Lbl, etichetta	101
lcm, minimo comune multiplo	102
left(), sinistra	102
LibPriv	56
LibPub	56
libreria	
crea collegamenti a oggetti	102
libShortcut(), crea collegamenti a oggetti libreria	102
limit() o lim(), limite	103
limite	
lim()	103
limit()	103
modello di	11
linea di regressione mediana-mediana, MedMed	120
lingua	
ottiene informazioni sulla lingua	89
LinRegBx, regressione lineare	104
LinRegMx, regressione lineare	105
LinRegtIntervals, regressione lineare	106
LinRegtTest (t Test regressione lineare)	108
linSolve()	109
list►mat(), da lista a matrice	110
lista, conteggio condizionato	42
lista, conteggio elementi	42
liste	
affianca/concatena, augment()	20
da espressione a lista, exp►list()	73
da lista a matrice, list►mat()	110
da matrice a lista, mat►list()	118
differenza, @list()	110
differenze in una lista, @list()	110
elementi vuoti in	254
in mezzo alla stringa, mid()	121
massimo, max()	118
minimo, min()	122
nuova, newList()	129
ordinamento ascendente, SortA	186

ordinamento discendente, SortD	187
prodotto scalare, dotP()	65
prodotto vettoriale, crossP()	43
prodotto, product()	148
somma cumulativa, cumulativeSum()	49
somma, sum()	193
In(), logaritmo naturale	111
LnReg, regressione logaritmica	111
Local, variabile locale	113
locale, Local	113
Lock, blocca variabile o gruppo di variabili	113
Log	
modello di	6
logaritmi	111
logaritmo naturale, ln()	111
Logistic, regressione logistica	115
LogisticD, regressione logistica	116
loop, Loop	117
Loop, loop	117
LU, scomposizione matrice inferiore-superiore	117
lunghezza arco, arcLen()	19
lunghezza della stringa	61

M

maggiore di o uguale a, 	235
maggiore di, >	235
massimo comune divisore, gcd()	86
massimo, max()	118
mat►list(), da matrice a lista	118
matrice (1×2)	
modello di	8
matrice (2×1)	
modello di	8
matrice (2×2)	
modello di	8
matrice ($m \times n$)	
modello di	8
matrice di correlazione, corrMat()	36
matrice identità, identity()	93
matrici	
affianca(concatena, augment())	20
autovalore, eigVl()	67
autovettore, eigVc()	66

casuale, randMat()	154
da lista a matrice, list•mat()	110
da matrice a lista, mat•list()	118
determinante, det()	60
diagonale, diag()	61
dimensione colonna, colDim()	32
dimensione righe, rowDim()	165
dimensione, dim()	61
forma a scalini per righe, ref()	157
forma a scalini ridotta per righe, rref()	166
identità, identity()	93
inverte righe, rowSwap()	166
massimo, max()	118
minimo, min()	122
moltiplicazione e somma di righe, mRowAdd()	124
norma colonna, colNorm()	32
norma riga, rowNorm()	165
nuova, newMat()	129
operazione con righe, mRow()	124
prodotto, product()	148
punto addizione, .+	230
punto divisione, .P	231
punto elevato a potenza, ^	231
punto moltiplicazione, *	231
punto sottrazione, .N	230
riempimento, Fill	78
scomposizione inferiore-superiore, LU	117
scomposizione QR, QR	149
somma cumulativa, cumulativeSum()	49
somma di righe, rowAdd()	165
somma, sum()	193
sottomatrice, subMat()	192, 194
trasposizione, T	195
max(), massimo	118
mean(), media	119
media, mean()	119
median(), mediana	120
mediana, median()	120
MedMed, linea di regressione mediana-mediana	120
memorizzazione simbolo, &	251-252
mentre, While	216
mid(), in mezzo alla stringa	121

min(), minimo	122
minimo comune multiplo, lcm	102
minimo, min()	122
minore di o uguale a, {	234
mirr(), indice interno di rendimento modificato	123
mod(), modulo	123
modelli	
derivata o derivata ennesima	10
derivata prima	9
derivata seconda	10
esponente	5
frazione	5
funzione esponenziale e	6
funzione piecewise a 2 tratti	6
funzione piecewise a N tratti	7
integrale definito	10
integrale indefinito	10
limite	11
Log	6
matrice (1×2)	8
matrice (2×1)	8
matrice (2×2)	8
matrice (m × n)	8
prodotto (P)	9
radice ennesima	5
radice quadrata	5
sistema di equazioni (2 equazioni)	7
sistema di equazioni (N equazioni)	7
sommatoria (G)	9
valore assoluto	7-8
modi	
impostazione, setMode()	172
modulo, mod()	123
moltiplicazione, *	227
mRow(), operazione con righe di matrice	124
mRowAdd(), moltiplicazione e somma di righe di matrice	124
MultReg (Regressione lineare multipla)	124
MultRegIntervals() (Intervalli di confidenza della previsione di regressione multipla)	125
MultRegTests() (Verifica sulla regressione lineare multipla)	126
N	
nand, operatore booleano	127
nCr(), combinazioni	128

nDerivative(), derivata numerica	129
newList(), nuova lista	129
newMat(), nuova matrice	129
nfMax(), massimo di una funzione calcolato numericamente	130
nfMin(), minimo di una funzione calcolato numericamente	130
nInt(), integrale numerico	131
nom), converti tasso effettivo in nominale	131
nor, operatore boolean	132
norm(), norma di Frobenius	132
norma di Frobenius, norm()	132
normalLine()(Retta normale)	133
normCdf()	133
normPdf()	133
not, operatore booleano	133
notazione in gradi, -	245
notazione in gradi/primi/secondi	246
notazione in gradienti, G	244
notazione in primi,	246
notazione in secondi, "	246
nPr(), disposizioni semplici	134
npv(), valore presente netto	135
nSolve(), soluzione numerica	136
nullo, test	101
numerica	
derivata, nDeriv()	130
derivata, nDerivative()	129
integrale, nInt()	131
soluzione, nSolve()	136
nuova	
lista, newList()	129
matrice, newMat()	129

o

oggetti	
crea collegamenti a libreria	102
OneVar, statistiche a una variabile	136
operatore conversione indiretta (#)	259
operatore di vincolo " "	250
operatore di vincolo. ordine di valutazione	258
operatori	
ordine di calcolo	258
operatori boolean	
and	13

nor	132
operatori booleani	
\Rightarrow	236
not	133
or	138
xor	216
Operatori booleani	
\Rightarrow	256
\Leftrightarrow	236
nand	127
or(Boolean), oppure	138
or, operatore booleano	138
ord(), codice numerico di carattere	139
ordinamento	
ascendente, SortA	186
discendente, SortD	187
ottieni/restituisci	
denominatore, getDenom()	88
informazioni variabile, getVarInfo()	89, 92
numeratore, getNum()	91

P

P▶Rx(), coordinata x rettangolare	139
P▶Ry(), coordinata y rettangolare	140
parte immaginaria, imag()	96
parte intera, iPart()	99
passa errore, PassErr	140
PassErr, passa errore	140
Pdf()	83
per, For	81
percentuale, %	232
piecewise()	141
poissCdf()	141
poissPdf()	142
polare	
coordinata, R▶Pr()	152
coordinata, R▶Pθ()	152
visualizza vettore, ▶Polar	142
polinomi	
calcola, polyEval()	144
casuale, randPoly()	155
polinomio di Taylor, taylor()	198
polyCoef()	143

polyDegree()	143
polyEval(), calcola polinomio	144
polyGcd()	144-145
PolyRoots()	145
potenza di dieci, $10^()$	249
PowerReg, regressione su potenza	146
Prgm, definisci programma	147
primo,	247
probabilità di distribuzione normale, normCdf()	133
probabilità di distribuzione t di Student, tCdf()	199
prodotto (P)	
modello di	9
prodotto vettoriale, crossP()	43
prodotto, product()	148
prodotto, $\Pi()$	240
prodSeq()	147
product(), prodotto	148
programmazione	
definisci programma, Prgm	147
display data, Disp	62
passa errore, PassErr	140
visualizza dati, Vis	168
programmi	
definizione di libreria privata	56
definizione di libreria pubblica	56
programmi e programmazione	
cancella errore, ClrErr	31
fine tentativo, EndTry	205
tentativo, Try	205
visualizza schermata I/O, Vis	168
visualizza schermo I/O, Disp	62
propFrac, frazione propria	148
punto	
addizione, .+	230
divisione, .P	231
moltiplicazione, *	231
sottrazione, .N	230
su potenza, ^	231
Q	
QR, scomposizione QR	149
QuadReg, regressione quadratica	150
quando, when()	215

R

R, radianti	245
R>Pr(), coordinata polare	152
R>Pθ(), coordinata polare	152
radianti, R	245
radice ennesima	
modello di	5
radice quadrata	
modello di	5
radice quadrata, √()	188, 240
rand(), numero casuale	153
randBin, numero casuale	153
randInt(), intero casuale	153
randMat(), matrice casuale	154
randNorm(), normale casuale	154
randPoly(), polinomio casuale	155
randSamp()	155
RandSeed, seme numero casuale	155
real(), reale	155
reale, real()	155
reciproco, ^-1	249
ref(), forma a scalini per righe	157
regressione cubica, CubicReg	48
regressione esponenziale, ExpReg	75
regressione lineare, LinRegAx	105
regressione lineare, LinRegBx	104, 106
regressione logaritmica, LnReg	111
regressione logistica, Logistic	115
regressione logistica, LogisticD	116
regressione quadratica, QuadReg	150
Regressione quartica (QuartReg)	151
regressione sinusoidale, SinReg	181
regressione su potenza, PowerReg	145-146, 158, 160, 200
regressioni	
cubica, CubicReg	48
esponenziale, ExpReg	75
linea mediana-mediana, MedMed	120
logaritmica, LnReg	111
Logistic (Regressione logistica)	115
logistica, Logistic	116
MultReg (Regressione lineare multipla)	124

quadratica, QuadReg	150
quartica, QuartReg	151
regressione lineare, LinRegAx	105
regressione lineare, LinRegBx	104, 106
regressione su potenza, PowerReg	146
sinusoidale, SinReg	181
regressions, regressioni	
regressione su potenza, PowerReg	145, 158, 160, 200
remain(), resto	158
Request	158
RequestStr	160
restituisci, Return	161
resto, remain()	158
retta normale, normalLine()	133
retta tangente, tangentLine()	197
Return, restituisci	161
riduzione trigonometrica, tCollect()	199
right(), destra	161
right, right()	34, 68, 98, 161, 214
rimuovi	
elementi vuoti dalla lista	57
ripeti, Cycle	50
risolvi, solve()	182
risultati a due variabili, TwoVar	210
risultati, statistica	189
risultato	
vis. rispetto ad e	72
visualizza rispetto al coseno	36
visualizza rispetto al seno	178
risultato (ultimo), Ans	17
rk23(), Runge Kutta function	161
rotate(), ruota	163
round(), arrotondamento	164
rowAdd(), somma di righe di matrice	165
rowDim(), dimensione righe matrice	165
rowNorm(), norma righe matrice	165
rowSwap(), inverti righe matrice	166
rref(), Forma a scalini ridotta per righe	166
ruota, rotate()	163
 S	
sblocco di variabili e gruppi di variabili	212
scelte rapide della tastiera	256

scelte rapide, tastiera	256
scomposizione QR, QR	149
se, If	94
$\sec^{-1}()$, secante inversa	167
$\sec()$, secante	166
$\operatorname{sech}^{-1}()$, secante iperbolica inversa	168
$\operatorname{sech}()$, secante iperbolica	167
segno negativo, introduzione di numeri negativi	259
segno, $\operatorname{sign}()$	176
seno, $\sin()$	178
seq(), sequenza	169
seqGen()	169
seqn()	170
sequence, seq()	169-170
sequenza, seq()	169
serie, series()	171
series(), serie	171
setMode(), imposta modo	172
shift(), sposta	174
$\operatorname{sign}()$, segno	176
simult(), equazioni simultanee	176
$\sin^{-1}()$, arcoseno	179
$\sin()$, seno	178
sine (Seno) visualizza lespessione rispetto a	178
$\sinh^{-1}()$, arcoseno iperbolico	180
$\sinh()$, seno iperbolico	180
sinistra, left()	102
SinReg, regressione sinusoidale	181
sistema di equazioni (2 equazioni) modello di	7
sistema di equazioni (N equazioni) modello di	7
soluzione di equazioni differenziali, deSolve()	58
solve(), risolvi	182
somma cumulativa, cumulativeSum()	49
somma dell'interesse pagato	241
somma di capitale versato	242
somma, sum()	193
somma, $\Sigma()$	241
sommatoria (G) modello di	9
SortA, ordinamento ascendente	186

SortD, ordinamento discendente	187
sostituzione con l'operatore " "	250
sottomatrice, subMat()	192, 194
sottrazione, -	226
sposta, shift()	174
sqrt(), radice quadrata	188
stat.results	189
stat.values	190
statistica	
combinazioni, nCr()	128
deviazione standard, stdDev()	190-191, 212
disposizioni semplici, nPr()	134
fattoriale, !	237
media, mean()	119
mediana, median()	120
normale casuale, randNorm()	154
risultati a due variabili, TwoVar	210
seme numero casuale, RandSeed	155
statistiche a una variabile, OneVar	136
varianza, variance()	213
statistiche a una variabile, OneVar	136
stdDevPop(), deviazione standard della popolazione	190
stdDevSamp(), deviazione standard del campione	191
string(), da espressione a stringa	192
stringa	
dimensione, dim()	61
lunghezza	61
stringa di caratteri, char()	28
stringa formato, format()	82
stringhe	
aggiunge, &	237
all'interno, InString	97
codice di carattere, ord()	139
conversione indiretta, #	243
da espressione a stringa, string()	192
da stringa a espressione, expr()	75, 114
destra, right()	161
dimensione, dim()	61
formato, format()	82
formattazione	82
in mezzo alla stringa, mid()	121
ruota, rotate()	163
sinistra, left()	102

sposta, shift()	174
stringa di caratteri, char()	28
utilizzo per creare nomi di variabile	259
strings	
right, right()	34, 68, 98, 161, 214
subMat(), sottomatrice	192, 194
sum(), somma	193
sumIf()	193
sumSeq()	194

T

T, trasposizione	195
tabella di ammortamento, amortTbl()	12, 21
tan ⁻¹ (), arcotangente	196
tan(), tangente	195
tangente, tan()	195
tangentLine() (Retta tangente)	197
tanh ⁻¹ (), arcotangente iperbolica	197
tanh(), tangente iperbolica	197
tasso di variazione media, avgRC()	21
tasso effettivo, eff()	66
tasso nominale, nom()	131
taylor(), polinomio di Taylor	198
tCdf(), probabilità di distribuzione t di Student	199
tCollect(), riduzione trigonometrica	199
termine dominante, dominantTerm()	63
test per nullità, isVoid()	101
Test_2S, verifica F su due campioni	84
tExpand(), espansione trigonometrica	199
Text, comando	200
tInterval, intervallo di confidenza t	201
tInterval_2Samp, intervallo di confidenza t su due campioni	202
ΔtmpCnv() [tmpCnv]	203
tmpCnv()	203
tPpdf(), densità di probabilità t di Student	204
trace() (Traccia)	204
trasposizione, T	195
trattino basso, _	247
Try, comando gestione errore	205
tTest, verifica t	206
tTest_2Samp, verifica t su due campioni	207
tvmFV()	208

tvmI()	208
tvmN()	208
tvmPmt()	209
tvmPV()	209
TwoVar, risultati a due variabili	210

U

uguale, =	233
unità	
converti	248
unitV(), vettore unità	212
unLock, sblocca variabile o gruppo di variabili	212

V

vai a, Goto	93
valore assoluto	
modello di	7-8
valore del denaro rapportato al tempo, importo della rata	209
valore del denaro rapportato al tempo, interesse	208
valore del denaro rapportato al tempo, numero di rate	208
valore del denaro rapportato al tempo, valore futuro	208
valore del denaro rapportato al tempo, valore presente	209
valore presente netto, npv()	135
valori dei risultati, statistica	190
variabile	
creazione di un nome da una stringa di caratteri	259
variabile locale, Local	113
variabili	
cancella tutte con il nome di un solo carattere	31
elimina, DelVar	57
locale, Local	113
variabili e funzioni	
copia	35
variabili, blocco e sblocco	89, 113, 212
varianza, variance()	213
varPop()	212
varSamp(), varianza campione	213
verifica numero primo, isPrime()	100
Verifica t sulla regressione lineare multipla	126
verifica t, tTest	206
vettore unità, unitV()	212
vettori	
prodotto scalare, dotP()	65

prodotto vettoriale, crossP()	43
unità, unitV()	212
visualizza vettore in forma cilindrica, ►Cylind	50
Vis, visualizza dati	168
visualizza come	
angolo decimale, ►DD	54
binario, ►Base2	22
esadecimale, ►Base16	24
gradi/primi/secondi, ►DMS	62
numero decimale, ►Base10	24
vettore in forma cilindrica, 4Cylind	50
vettore polare, ►Polar	142
vettore rettangolare, ►Rect	156
vettore sferico, ►Sphere	187
visualizza come vettore rettangolare, ►Rect	156
visualizza dati, Disp	62
visualizza dati, Vis	168
visualizza gradi/primi/secondi, ►DMS	62
visualizza vettore in forma cilindrica, ►Cylind	50
visualizza vettore sferico, ►Sphere	187

W

warnCodes(), Warning codes	214
when(), quando	215
While, mentre	216

X

x^2 , quadrato	230
XNOR	236
xor, or esclusivo booleano	216

Z

zero, zeroes()	217
zeroes(), zero	217
zInterval, intervallo di confidenza z	219
zInterval_1Prop, intervallo di confidenza z per una proporzione	220
zInterval_2Prop, intervallo di confidenza z per due proporzioni	221
zInterval_2Samp, intervallo di confidenza z su due campioni	221
zTest	222
zTest_1Prop, verifica z per una proporzione	223
zTest_2Prop, verifica z per due proporzioni	224
zTest_2Samp, verifica z su due campioni	225

Δ

Δlist(), differenza in una lista 110

X

χ^2 Cdf() 29

χ^2 GOF 30

χ^2 Pdf() 31