



TI-Nspire™ CAS

Sovelluksen käsikirja

Lue lisää TI teknologiaan käytönaikaisessa ohjeessa education.ti.com/eguide.

Tärkeitä tietoja

Ellei muuten ilmoiteta ohjelman mukaan liitettyssä käyttöluvassa, Texas Instruments ei anna minkäänlaista suoraa tai välillistä takuuta mukaan lukien, mutta ei näihin rajoittuen, kaikki välilliset takuut, jotka koskevat kaikkien ohjelmien ja kirjojen myyntikelpoisuutta tai erityiseen tarkoitukseen sopivuutta, ja tarjoaa kyseisiä materiaaleja ainoastaan "sellaisina kuin ne ovat" -pohjalla. Texas Instruments ei ole missään tapauksessa vastuussa kenellekään mistään erityisistä, rinnakkaisista, tahattomista tai seurauksellisista vaurioista näiden materiaalien hankinnan tai käytön aiheuttamana, ja Texas Instruments:n yksinomainen ja eksklusiivinen vastuu toimintamuodosta riippumatta ei ylitä määrää, joka on asetettu käyttöluvassa ohjelmaa varten. Texas Instruments ei myöskään vastaa mistään vaateista, joita toinen osapuoli voi esittää aiheutuen näiden materiaalien käytöstä.

© 2006 - 2019 Texas Instruments Incorporated

Sisällys

Lausekemallit	1
Luettelo aakkosjärjestyksessä	8
A	8
B	17
C	21
D	47
E	60
F	71
G	81
I	91
L	100
M	116
N	125
O	134
P	137
Q	146
R	150
S	165
T	191
U	207
V	208
W	209
X	211
Z	212
Symbolit	221
Tyhjät elementit	249
Matemaattisten lausekkeiden syöttäminen pikavalintojen avulla	251
EOS-järjestelmän (yhtälökäyttöjärjestelmä) hierarkia	253
Vakiot ja arvot	255
Virhekoodit ja viestit	256
Varoituskoodit ja -viestit	263
Yleistä	265
Online-tuki	265

Ota yhteyttä TI-tukeen	265
Huolto- ja takuutiedot	265
Indeksi	266

Lausekemallit

Lausekemallien avulla voit syöttää matemaattisia lausekkeita normaalissa matemaattisessa muodossa. Lisätessäsi mallin se näkyy syöterivillä siten, että elementtien syöttökohdissa on pienet ruudut. Kohdistin on syötettävän elementin kohdalla.

Voit siirtää kohdistimen kunkin elementin kohdalle nuolipainikkeilla tai painikkeella **tab**, jonka jälkeen voit kirjoittaa elementin arvon tai lausekkeen. Lauseke sievennetään painamalla painikkeita **enter** tai **ctrl enter**.

Murtolukumalli

ctrl **÷** painikkeet



Huomaa: Katso myös / (jakolasku), sivu 223.

Esimerkki:

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

Eksponenttimalli

^ painike



Huomaa: Syötä ensimmäinen arvo, paina **^** ja syötä sen jälkeen eksponentti. Voit palauttaa kohdistimen perusviivalle painamalla oikealle osoittavaa nuolta (►).

Huomaa: Katso myös ^ (potenssi), sivu 224.

Esimerkki:

$$2^3 = 8$$

Neliöjuurimalli

ctrl **x²** painikkeet



Huomaa: Katso myös $\sqrt{}$ (neliöjuuri), sivu 235.

Esimerkki:

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

N:s juuri -malli

ctrl **^** **painikkeet**



Huomaa: Katso myös **root()**, sivu 161.

Esimerkki:

$$\sqrt[3]{8} \quad 2$$
$$\sqrt[3]{\{8, 27, b\}} \quad \left\{ 2, 3, b^{\frac{1}{3}} \right\}$$

e eksponenttimalli

e^x **painikkeet**



e-kantainen eksponenttifunktio korotettuna potenssiin

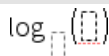
Huomaa: Katso myös **e^()**, sivu 60.

Esimerkki:

$$e^1 \quad e$$
$$e^1. \quad 2.71828182846$$

Logaritmimalli

ctrl **10^x** **painike**



Laskee määritetyn kantaan logaritmin. 10-kantaista logaritmia laskettaessa kantaluku jätetään pois.

Huomaa: Katso myös **log()**, sivu 112.

Esimerkki:

$$\log_{10}(2.) \quad 0.5$$

Paloittain määritellyn funktion malli (2-osainen)

Katalogi > **log_a^a**



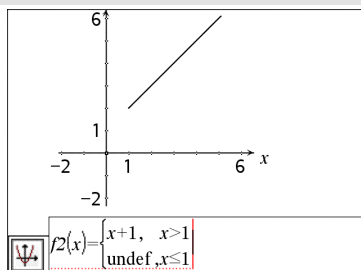
Esimerkki:

Voit luoda lausekkeita ja ehtoja 2-osaiselle paloittain määritellylle funktiolle. Lisää osa napsauttamalla mallia ja toista malli.

Huomaa: Katso myös **piecewise()**, sivu 139.

Paloittain määritellyn funktion malli (2-osainen)

Katalogi >

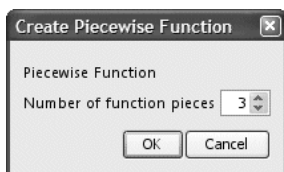


Paloittain määritellyn funktion malli (N-osainen)

Katalogi >



Voit luoda lausekkeita ja ehtoja N -osaiselle paloittain määritellylle funktiolle. Laskin pyytää N :n arvoa.



Esimerkki:

Katso paloittain määritellyn funktion (2-osaisen) mallin esimerkki.

Huomaa: Katso myös `piecewise()`, sivu 139.

Yhtälöparin malli

Katalogi >



Luo kahden yhtälön ryhmän. Voit lisätä rivin olemassa olevaan yhtälöön napsauttamalla mallia ja toistamalla mallin.

Huomaa: Katso myös `system()`, sivu 191.

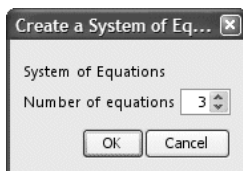
Esimerkki:

$$\begin{array}{l} \text{solve} \left(\begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y \right) \quad x=\frac{5}{2} \text{ and } y=-\frac{5}{2} \\ \text{solve} \left(\begin{cases} y=x^2-2 \\ x+2 \cdot y=-1 \end{cases}, x, y \right) \\ x=-\frac{3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and } y=-1 \end{array}$$

N-osaisen yhtälöryhmän malli

Katalogi > 

Voit luoda N yhtälöä sisältävän yhtälöryhmän. Laskin pyytää N :n arvoa.



Esimerkki:

Katso yhtälöparin (2 yhtälöä) mallin esimerkki.

Huomaa: Katso myös `system()`, sivu 191.

Itseisarvon malli

Katalogi > 



Huomaa: Katso myös `abs()`, sivu 8.

Esimerkki:

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

dd°mm'ss.ss"-malli

Katalogi > 

0°00'00"

Voit syöttää kulmia muodossa **dd°mm'ss.ss"**, jossa **dd** on desimaaliasteiden lukumäärä, **mm** on minuuttimäärä, ja **ss.ss** on sekuntimäärä.

Esimerkki:

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

Matriisimalli (2 x 2)

Katalogi > 



Esimerkki:

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

Luo 2 x 2 -matriisin.

Matriisimalli (1 x 2)

Katalogi > 



Esimerkki:

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

Matriisimalli (2 x 1)

Katalogi > 

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

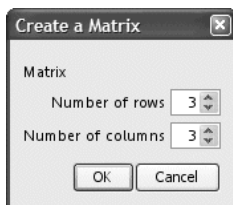
Esimerkki:

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

Matriisimalli (m x n)

Katalogi > 

Malli tulee näkyviin määritettyäsi riven ja sarakkeiden lukumäärän syöttöruutuun.



Esimerkki:

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

Huomaa: Jos luot paljon rivejä ja sarakkeita sisältävän matriisin, voi kestää jonkin aikaa, ennen kuin matriisi tulee näkyviin.

Summan malli (Σ)

Katalogi > 

$$\sum_{i=0}^0 (0)$$

Esimerkki:

$$\sum_{n=3}^7 (n) \quad 25$$

Huomaa: Katso myös $\Sigma()$ (**sumSeq**), sivu 236.

Tulon malli (II)

Katalogi > 

$$\frac{\boxed{}}{\boxed{} \boxed{}} (\boxed{})$$

$$\boxed{} = \boxed{}$$

Huomaa: Katso myös **II()** (**prodSeq**), sivu 235.

Esimerkki:

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

Ensimmäisen derivaatan malli

Katalogi > 

$$\frac{d}{d\boxed{}}(\boxed{})$$

Ensimmäisen derivaatan mallia voi käyttää myös laskettaessa ensimmäinen derivaatta pisteessä.

Huomaa: Katso myös **d()** (**derivaatta**), sivu 232.

Esimerkki:

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

$$\frac{d}{dx}(x^3)|_{x=3} \quad 27$$

Toisen derivaatan malli

Katalogi > 

$$\frac{d^2}{d\boxed{}^2}(\boxed{})$$

Toisen derivaatan mallia voi käyttää myös laskettaessa toinen derivaatta pisteessä.

Huomaa: Katso myös **d()** (**derivaatta**), sivu 232.

Esimerkki:

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

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

N:n:n derivaatan malli

Katalogi > 

$$\frac{d}{d\boxed{}}(\boxed{})$$

n :n:n derivaatan mallia voidaan käyttää laskettaessa n :s derivaatta.

Huomaa: Katso myös **d()** (**derivaatta**), sivu 232.

Esimerkki:

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

Määrätyn integraalin malli

Katalogi > 

$$\int_a^b f(x) dx$$

Huomaa: Katso myös `f()` `integraali()`, sivu 221.

Esimerkki:

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

määrittämättömän integraalin malli

Katalogi > 

$$\int f(x) dx$$

Huomaa: Katso myös `f()` `integral()`, sivu 221.

Esimerkki:

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

Raja-arvon malli

Katalogi > 

$$\lim_{x \rightarrow a} f(x)$$

Vasemman puolen raja-arvon saat painikkeella `-` tai `(-)`. Oikean puolen raja-arvon saat painikkeella `+`.

Huomaa: Katso myös `limit()`, sivu 102.

Esimerkki:

$$\lim_{x \rightarrow 5} (2 \cdot x + 3) = 13$$

Luettelo aakkosjärjestyksessä

Komennot, joiden nimiä ei voi järjestää aakkosjärjestykseen (esimerkiksi +, ! ja >), on esitetty tämän kappaleen lopussa alkaen sivulta (sivu 221). Ellei toisin ole mainittu, kaikki tämän kappaleen esimerkit on suoritettu laskimen oletustilassa, eikä mitään muuttujia ole määritetty.

A

abs()

Katalogi >

abs(*LausI*)⇒*lauseke*

abs(*Listal*)⇒*lista*

abs(*MatriisiI*)⇒*matriisi*

Laskee argumentin itseisarvon.

Huomaa: Katso myös **Itseisarvon malli**, sivu 4.

Jos argumentti on kompleksiluku, määrittää luvun moduulin.

Huomaa: Kaikkia määrittämättömiä muuttujia käsitellään reaali muuttujina.

$\left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\}$	$\left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\}$
$2-3 \cdot i$	$\sqrt{13}$
$ z $	$ z $
$ x+y \cdot i $	$\sqrt{x^2+y^2}$

amortTbl()

Katalogi >

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

Lainan lyhennysfunktio, joka laskee lyhennystaulukon tiettyjen TVM-argumenttien perusteella.

NPmt on taulukon maksuerien lukumäärä. Taulukko alkaa ensimmäisestä maksuerästä.

N, I, PV, Pmt, FV, PpY, CpY ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 205.

- Jos jätät argumentin *Pmt* pois, sen oletusarvoksi tulee *Pmt=tvmPmt(N,I,PV,FV,PpY,CpY,PmtAt)*.
- Jos jätät argumentin *FV* pois, sen

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

oletusarvoksi tulee $FV=0$.

- Argumenttien PpY , CpY ja $PmtAt$ oletusarvot ovat samat kuin TVM-funktioilla.

pyörArvo määrittää pyöristyksessä käytettävien desimaalien määrän.
Oletusarvo=2.

Tulosmatriisin sarakkeet ovat seuraavassa järjestyksessä: maksuerän numero, koron määrä, pääoman lyhennysmäärä ja velkasaldo.

Rivillä n näkyvä saldo on maksuerän n jälkeen jäljellä oleva velkasaldo.

Voit käyttää tulosmatriisia syötteenä muissa lyhennyslaskutoimituksissa $\Sigma\text{Int}()$ ja $\Sigma\text{Prn}()$, sivu 236, sekä **bal()**, sivu 17.

and

BooleanLaus1 and BooleanLaus2 $\Rightarrow$ *Boolean lausekeBooleanLista1 and BooleanLista2* $\Rightarrow$ *Boolean listaBooleanMatriisi1 and BooleanMatriisi2* $\Rightarrow$ *Boolean matriisi*

Määrittää totuusarvon tosi tai epätosi tai antaa vastauksena sievennetyn muodon alkuperäisestä syötteestä.

Kokonaisluku1

and*Kokonaisluku2* $\Rightarrow$ *kokonaisluku*

Vertaa kahta reaalikokonaislukua bitti bitiltä and-operaation avulla. Sisäisesti kumpikin kokonaisluku muunnetaan etumerkilliseksi, 64 bitin binaariluvuksi. Kun vastaavia bittejä verrataan, tulos on 1, jos kumpikin bitti on 1. Muussa tapauksessa tulos on 0. Laskettu arvo edustaa bittituloksia, ja se näkyy kantelukutilan mukaisesti.

$x \geq 3$ and $x \geq 4$	$x \geq 4$
$\{x \geq 3, x \leq 0\}$ and $\{x \geq 4, x \leq 2\}$	$\{x \geq 4, x \leq 2\}$

Heksadesimaalisessa kantelukutilassa:

0h7AC36 and 0h3D5F	0h2C16
--------------------	--------

Tärkeää: Nolla, ei O-kirjain.

Binaarisessa kantelukutilassa:

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

Desimaalisessa kantelukutilassa:

37 and 0b100	4
--------------	---

Kokonaisluvut voi syöttää minkä tahansa luvun kantalukuna. Binaarisen syötteen edelle tulee merkitä etumerkki 0b ja heksadesimaalisen syötteen edelle 0h. Jos etumerkkiä ei ole, kokonaislukuja käsitellään desimaalilukuina (kantaluku 10).

Jos syötät desimaalikokonaisluvun, joka on liian suuri etumerkilliselle, 64 bitin binaarimuodolle, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle.

Huomaa: Binaarisessa syötteessä voi olla korkeintaan 64 numeroa (etuliitettä 0b ei lasketa). Heksadesimaalisessa syötteessä voi olla korkeintaan 16 numeroa.

angle()

angle(LausI) ⇒ lauseke

Laskee argumentin kulman tulkiten argumentin kompleksiluvuksi.

Huomaa: Kaikkia määrittämättömiä muuttujia käsitellään reaali muuttujina.

Astekulmatilassa:

$$\text{angle}(0+2 \cdot i) \quad 90$$

Graadikulmatilassa:

$$\text{angle}(0+3 \cdot i) \quad 100$$

Radiaanikulmatilassa:

$$\text{angle}(1+i) \quad \frac{\pi}{4}$$

$$\text{angle}(z) \quad \frac{\pi \cdot (\text{sign}(z)-1)}{2}$$

$$\text{angle}(x+i \cdot y) \quad \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right)$$

$$\text{angle}\left(\left\{\left\{1+2 \cdot i, 3+0 \cdot i, 0-4 \cdot i\right\}\right\}\right) \quad \left\{\frac{\pi}{2}-\tan^{-1}\left(\frac{1}{2}\right), 0, \frac{\pi}{2}\right\}$$

angle(ListaI) ⇒ lista

angle(MatriisiI) ⇒ matriisi

Laskee listan tai matriisin *Listai*:n tai *MatriisiI*:n elementtien kulmista tulkiten jokaisen elementin kompleksiluvuksi, joka edustaa kaksiulotteista suorakulmakoordinaattipistettä.

ANOVA

ANOVA *Listai*, *Listai2*[*Listai3*,...,*Listai20*]
[,*Lippu*]

Suorittaa yksisuuntaisen varianssianalyysin 2-20 perusjoukon keskiarvon vertailua varten. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Lippu=0 datalle, *Lippu*=1 tilastoille

Tulosmuuttuja	Kuvaus
stat.F	F-tilaston arvo
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat.df	Ryhmiä vapausasteet
stat.SS	Ryhmiä neliöiden summa
stat.MS	Ryhmiä keskineliöt
stat.dfError	Virheiden vapausasteet
stat.SSError	Virheiden neliöiden summa
stat.MSError	Virheiden keskineliö
stat.sp	Poolattu keskihajonta
stat.xbarlist	Listojen syötteiden keskiarvo
stat.CLowerList	95 %:n luottamusvälit jokaisen syötelistan keskiarvolle
stat.CUpperList	95 %:n luottamusvälit jokaisen syötelistan keskiarvolle

ANOVA2way

ANOVA2way *List1,Lista2*
[,Lista3,...,Lista10][,TasoRivi]

Laskee kaksisuuntaisen varianssianalyysin 2-10 perusjoukon keskiarvojen vertaamiseksi. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

TasoRivi=0 lohkolle

TasoRivi=2,3,...,*Pit*-1, kahdelle tekijälle,
jossa *Pit*=pituus(*List1*)=pituus(*List2*) = ... =
pituus(*List10*) ja *Pit* / *TasoRivi* ∈ {2,3,...}

Tulokset: Lohkomuoto

Tulosmuuttuja	Kuvaus
stat.F	F-tilasto, saraketekijän F-tilasto
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat.df	Saraketekijän vapausasteet
stat.SS	Saraketekijän neliöiden summa
stat.MS	Saraketekijän keskineliöt
stat.FBlock	F-tilasto, tekijän F-tilasto
stat.PValBlock	Pienin todennäköisyys, jolla nollahypoteesi voidaan hylätä
stat.dfBlock	Tekijän vapausasteet
stat.SSBlock	Tekijän neliöiden summa
stat.MSBlock	Tekijän keskineliöt
stat.dfError	Virheiden vapausasteet
stat.SSError	Virheiden neliöiden summa
stat.MSError	Virheiden keskineliöt
stat.s	Virheen keskihajonta

SARAKETEKIJÄN tulokset

Tulosmuuttuja	Kuvaus
stat.Fcol	F-tilasto, saraketekijän F-tilasto
stat.PValCol	Saraketekijän todennäköisyysarvo
stat.dfCol	Saraketekijän vapausasteet
stat.SSCol	Saraketekijän neliöiden summa
stat.MSCol	Saraketekijän keskineliöt

RIVITEKIJÄN tulokset

Tulosmuuttuja	Kuvaus
stat.FRow	F-tilasto, rivitekijän F-tilasto
stat.PValRow	Rivitekijän todennäköisyysarvo
stat.dfRow	Rivitekijän vapausasteet
stat.SSRow	Rivitekijän neliöiden summa
stat.MSRow	Rivitekijän keskineliöt

VUOROVAIKUTUKSEN tulokset

Tulosmuuttuja	Kuvaus
stat.FInteract	F-tilasto, vuorovaikutuksen F-tilasto
stat.PValInteract	Vuorovaikutuksen todennäköisyysarvo
stat.dfiInteract	Vuorovaikutuksen vapausasteet
stat.SSInteract	Vuorovaikutuksen neliöiden summa
stat.MSInteract	Vuorovaikutuksen keskineliöt

VIRHEIDEN tulokset

Tulosmuuttuja	Kuvaus
stat.dfError	Virheiden vapausasteet
stat.SSError	Virheiden neliöiden summa
stat.MSError	Virheiden keskineliöt
s	Virheen keskihajonta

ans	ctrl (-) painikkeet	
ans⇒arvo	56	56
Näyttää viimeksi sievennetyn lausekkeen tuloksen.	56+4	60
	60+4	64

approx()	Katalogi > 	
approx(Laus1)⇒lauseke		
Määrittää argumentin sievennetyn arvon lausekkeena, joka sisältää desimaaliarvoja, mikäli mahdollista, riippumatta nykyisestä Automaattinen tai likimääräinen -tilasta.		
Tämä vastaa argumentin syöttämistä ja painikkeen ctrl enter painamista.		
approx(Lista1)⇒lista		
approx(Matriisi1)⇒matriisi		
	$\text{approx}\left(\frac{1}{3}\right)$	0.333333
	$\text{approx}\left(\left\{\frac{1}{3}, \frac{1}{9}\right\}\right)$	{0.333333,0.111111}
	$\text{approx}(\{\sin(\pi),\cos(\pi)\})$	{0,-1.}
	$\text{approx}(\left[\sqrt{2} \quad \sqrt{3}\right])$	[1.41421 1.73205]
	$\text{approx}\left(\left[\frac{1}{3} \quad \frac{1}{9}\right]\right)$	[0.333333 0.111111]
	$\text{approx}(\{\sin(\pi),\cos(\pi)\})$	{0,-1.}
	$\text{approx}(\left[\sqrt{2} \quad \sqrt{3}\right])$	[1.41421 1.73205]

approx()

Katalogi > 

Määrittää listan tai *matriisin*, jossa jokainen elementti on laskettu desimaaliarvoksi, mikäli mahdollista.

► approxFraction()

Katalogi > 

Laus ► **approxFraction**(*[Tol]*) ⇒ *lauseke*

$$\frac{1}{2} + \frac{1}{3} + \tan(\pi) \quad 0.833333$$

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

$$0.8333333333333333 \text{ ► } \text{approxFraction}(5.E-14)$$

Matriisi ► **approxFraction**(*[Tol]*) ⇒ *matriisi*

Laskee syötteen murtolukuna käyttäen toleranssia *Tol*. Jos operaattori *Tol* jätetään pois, laskin käyttää toleranssia 5.E-14.

$$\frac{5}{6}$$
$$\{\pi, 1.5\} \text{ ► } \text{approxFraction}(5.E-14)$$
$$\left\{ \frac{5419351}{1725033}, \frac{3}{2} \right\}$$

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla **@>approxFraction(...)**.

approxRational()

Katalogi > 

approxRational(*Laus*, *tol*) ⇒ *lauseke*

$$\text{approxRational}(0.333, 5 \cdot 10^{-5}) \quad \frac{333}{1000}$$

approxRational(*Lista*, *tol*) ⇒ *lista*

$$\text{approxRational}(\{0.2, 0.33, 4.125\}, 5.E-14)$$

approxRational(*Matriisi*, *tol*) ⇒ *matriisi*

$$\left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$$

Laskee argumentin murtolukuna käyttäen toleranssia *tol*. Jos operaattori *Tol* jätetään pois, laskin käyttää toleranssia 5.E-14.

arccos()

Katso $\cos^{-1}()$, sivu 33.

arccosh()

Katso $\cosh^{-1}()$, sivu 34.

arccot()

Katso $\cot^{-1}()$, sivu 35.

augment()

Katalogi > **augment(Lista1, Lista2)⇒lista**

Luo uuden listan, joka on *Lista2* liitettynä *Lista1*:n loppuun.

augment(Matriisi1, Matriisi2)⇒matriisi

Luo uuden matriisin, joka on *Matriisi2* liitettynä *Matriisi1*:een. Kun käytetään merkkiä “,” matriiseiden rivimäärien on oltava samat, ja *Matriisi2* liitetään *Matriisi1*:een uusina sarakkeina. Ei muuta *Matriisi1*:ä eikä *Matriisi2*:a.

$$\text{augment}(\{1, -3, 2\}, \{5, 4\}) \quad \{1, -3, 2, 5, 4\}$$

$$\begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} \rightarrow m1 \quad \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline 5 \\ \hline 6 \\ \hline \end{array} \rightarrow m2 \quad \begin{array}{|c|} \hline 5 \\ \hline 6 \\ \hline \end{array}$$

$$\text{augment}(m1, m2) \quad \begin{array}{|c|c|c|} \hline 1 & 2 & 5 \\ \hline 3 & 4 & 6 \\ \hline \end{array}$$

avgRC()

Katalogi > **avgRC(Laus1, Muutt [=Arvo] [, Askel])**
⇒lauseke

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

avgRC(Laus1, Muutt [=Arvo] [, Lista1])
⇒lista

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

avgRC(Lista1, Muutt [=Arvo] [, Askel])
⇒lista

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

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

avgRC(Matriisi1, Muutt [=Arvo] [, Askel])
⇒matriisi

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

Laskee erotusosamäärän eteenpäin (keskimääräisen muutosnopeuden).

Laus1 voi olla käyttäjän määrittämä funktionimi (katso **Func**).

Kun *Arvo* määritetään, se ohittaa mahdolliset aikaisemmat muuttujamäärytykset tai mahdolliset muuttujan nykyiset "I" -sijoitukset.

Askel on askeleen arvo. Jos *Askel* jätetään pois, sen oletusarvo on 0.001.

Huomaa, että samankaltaisessa funktiossa **centralDiff()** käytetään keskeiserotusomäärrää.

B

bal()

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

bal(*NPmt*,*amortTable*)⇒*arvo*

Lyhennysfunktio, joka laskee määritetyn maksuerän jälkeen jäljellä olevan velkasaldon.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 205.

NPmt määrittää sen maksuerän numeron, jonka jälkeen velkasaldo halutaan laskea.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 205.

- Jos jätät argumentin *Pmt* pois, sen oletusarvoksi tulee ***Pmt=tvmpmt(N,I,PV,FV,PpY,CpY,PmtAt)***.
- Jos jätät argumentin *FV* pois, sen oletusarvoksi tulee ***FV=0***.
- Argumenttien *PpY*, *CpY* ja *PmtAt* oletusarvot ovat samat kuin TVM-funktioilla.

pyörArvo määrittää pyöristyksessä käytettävien desimaalien määrän. Oletusarvo=2.

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

bal(*NPmt*,*amortTable*) laskee maksueränumeron *NPmt* jälkeen jäljellä olevan velkasaldon lyhennystaulukon *amortTable* perusteella. *amortTable*-argumentin on oltava matriisi, joka on kohdassa **amortTbl()** kuvatun muotoinen, katso sivu 8.

Huomaa: Katso myös $\Sigma\text{Int}()$ ja $\Sigma\text{Prn}()$, sivu 236.

► Base2 (►Kantaluku2)

Kokonaisluku1 ► **Base2** ⇒ *kokonaisluku*

256 ► Base2	0b100000000
0h1F ► Base2	0b11111

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>**Base2**.

Muuttaa *Kokonaisluku1*:n binaariluvuksi. Binaariluvuissa on aina etuliite 0b ja heksadesimaaliluvuissa etuliite 0h.

Ilman etuliitettä *Kokonaisluku1*:ä käsitellään desimaalilukuna (kantaluku 10). Vastaus näkyy binaarilukuna kantalukutilasta riippumatta.

Negatiiviset luvut näytetään kahden komplementteina. Esimerkki:

–1 näkyy muodossa
0hFFFFFFFFFFFFFFFF heksadesimaalisessa kantalukutilassa
0b111...111 (64 ykköstä) binaarisessa kantalukutilassa

–2⁶³ näkyy muodossa
0h8000000000000000 heksadesimaalisessa kantalukutilassa
0b100...000 (63 zeros) binaarisessa kantalukutilassa

Jos syötät desimaalikokonaisluvun, joka on etumerkillisen, 64 bitin binaarimuodon lukualueen ulkopuolella, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle. Tarkastele seuraavassa esitettyjä esimerkkejä lukualueen ulkopuolella olevista arvoista.

2^{63} muuttuu muotoon -2^{63} ja näkyy muodossa
 0h8000000000000000 heksadesimaalisessa kantelukutilassa
 0b100...000 (63 zeros) binaarisessa kantelukutilassa

2^{64} muuttuu muotoon 0 ja näkyy
 0h0 heksadesimaalisessa kantelukutilassa
 0b0 binaarisessa kantelukutilassa

$-2^{63} - 1$ muuttuu muotoon $2^{63} - 1$ ja näkyy muodossa
 0h7FFFFFFFFFFFFFFF heksadesimaalisessa kantelukutilassa
 0b111...111 (64 ykköstä) binaarisessa kantelukutilassa

► Base10 (► Kantaluku10)

Kokonaisluku1 ► Base10 ⇒ kokonaisluku

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>Base10.

Muuttaa *Kokonaisluku1*:n desimaaliluvuksi (kantaluku 10). Binaarisen syötteen edellä tulee aina olla etumerkki 0b ja heksadesimaalisen syötteen edellä 0h.

0b *binaariluku*
 0h *heksadesimaaliluku*

Nolla, ei O-kirjain, jonka perässä on b tai h.

Binaariluvussa voi olla enintään 64 numeroa. Heksadesimaaliluvussa voi olla enintään 16 numeroa.

0b10011►Base10	19
0h1F►Base10	31

Ilman etuliitettä *Kokonaisluku1*:ä käsitellään desimaalilukuna. Vastaus näkyy desimaalilukuna kantalukutilasta riippumatta.

► Base16 (►Kantaluku16)

Kokonaisluku1 ►Base16⇒*kokonaisluku*

256►Base16 0h100

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @►Base16.

0b111100001111►Base16 0hF0F

Muuttaa *Kokonaisluku1*:n heksadesimaaliluvuksi. Binaariluvuissa on aina etuliite 0b ja heksadesimaaliluvuissa etuliite 0h.

0b *binaariluku*

0h *heksadesimaaliluku*

Nolla, ei O-kirjain, jonka perässä on b tai h.

Binaariluvussa voi olla enintään 64 numeroa. Heksadesimaaliluvussa voi olla enintään 16 numeroa.

Ilman etuliitettä *Kokonaisluku1*:ä käsitellään desimaalilukuna (kantaluku 10). Vastaus näkyy heksadesimaalilukuna kantalukutilasta riippumatta.

Jos syötät desimaalikokonaisluvun, joka on liian suuri etumerkilliselle, 64 bitin binaarimuodolle, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle.

Jos syötät desimaalikokonaisluvun, joka on etumerkillisen, 64 bitin binaarimuodon lukualueen ulkopuolella, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle. Lisätietoja, katso ►Base2, sivu 18.

binomCdf()

binomCdf(n, p)⇒*lista*

binomCdf()Katalogi > 

binomCdf($n, p, \text{alaraja}, \text{yläraja}$) $\Rightarrow$ luku, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

binomCdf($n, p, \text{yläraja}$) kun $P(0 \leq X \leq \text{yläraja})$
 $\Rightarrow$ luku, jos *yläraja* on luku, *lista*, jos *yläraja* on lista

Laskee kumulatiivisen todennäköisyyden diskreetille binomiselle jakaumalle, jossa toistojen määrä on n ja jokaisen toiston onnistumistodennäköisyys on p .

Kun $P(X \leq \text{yläraja})$, aseta *alaraja*=0

binomPdf()Katalogi > 

binomPdf(n, p) $\Rightarrow$ lista

binomPdf($n, p, XVal$) $\Rightarrow$ luku, jos *XVal* on luku, *lista*, jos *XVal* on lista

Laskee todennäköisyyden diskreetille binomiselle jakaumalle, jossa toistojen määrä on n ja jokaisen toiston onnistumistodennäköisyys on p .

C**ceiling()**Katalogi > 

ceiling(*LausI*) $\Rightarrow$ kokonaisluku

<code>ceiling(.456)</code>	1.
----------------------------	----

Laskee lähimmän kokonaisluvun, joka on $\geq$ argumentti.

Argumentti voi olla reaali- tai kompleksiluku.

Huomaa: Katso myös **floor()**.

ceiling(*ListaI*) $\Rightarrow$ lista

ceiling(*MatriisiI*) $\Rightarrow$ matriisi

Laskee listan tai matriisin jokaisen elementin ylärajasta.

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

centralDiff()

Katalogi > 

centralDiff(*LausI*, *Muutt* [=Arvo][, *Askel*])
⇒ *lauseke*

centralDiff(*LausI*, *Muutt* [, *Askel*])
| *Muutt* = Arvo ⇒ *lauseke*

centralDiff(*LausI*, *Muutt* [=Arvo][, *Listal*])
⇒ *lista*

centralDiff(*Listal*, *Muutt* [=Arvo][, *Askel*])
⇒ *lista*

centralDiff(*MatriisiI*, *Muutt* [=Arvo]
[, *Askel*]) ⇒ *matriisi*

Laskee numeerisen derivaatan käyttäen keskeiserotusosamäärän kaavaa.

Kun *Arvo* määritetään, se ohittaa mahdolliset aikaisemmat muuttujamäärittymät tai mahdolliset muuttujan nykyiset "|" -sijoitukset.

Askel on askeleen arvo. Jos *Askel* jätetään pois, sen oletusarvo on 0.001.

Listal:tä tai *MatriisiI*:tä käytettäessä operaatio mapataan listan arvojen tai matriisin elementtien suhteen.

Huomaa: Katso myös **avgRC()** ja **d()**.

$$\begin{aligned} \text{centralDiff}(\cos(x), x, h) &= \frac{-\cos(x-h) - \cos(x+h)}{2 \cdot h} \\ \lim_{h \rightarrow 0} (\text{centralDiff}(\cos(x), x, h)) &= -\sin(x) \\ \text{centralDiff}(x^3, x, 0.01) &= 3 \cdot (x^2 + 0.000033) \\ \text{centralDiff}(\cos(x), x) \Big|_{x=\frac{\pi}{2}} &= -1. \\ \text{centralDiff}(x^2, x, \{0.01, 0.1\}) &= \{2 \cdot x, 2 \cdot x\} \end{aligned}$$

cFactor()

Katalogi > 

cFactor(*LausI*, *Muutt*) ⇒ *lauseke*

cFactor(*Listal*, *Muutt*) ⇒ *lista*

cFactor(*MatriisiI*, *Muutt*) ⇒ *matriisi*

cFactor(*LausI*) jakaa *LausI*:n kaikki muuttujat supistaen ne yhteisellä nimittäjällä.

LausI:ä jaetaan tekijöihin mahdollisimman paljon kohti lineaarisia rationaalitekijöitä, vaikka tästä saataisiin uusia ei-reaalilukuja. Tämä vaihtoehto on sopiva, jos haluat jakaa lausekkeen tekijöihin useamman kuin yhden muuttujan suhteen.

$$\begin{aligned} \text{cFactor}(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x) &= a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i) \\ \text{cFactor}\left(x^2 + \frac{4}{9}\right) &= \frac{(3 \cdot x - 2 \cdot i) \cdot (3 \cdot x + 2 \cdot i)}{9} \\ \text{cFactor}(x^2 + 3) &= x^2 + 3 \\ \text{cFactor}(x^2 + a) &= x^2 + a \end{aligned}$$

cFactor()

Katalogi > 

cFactor(*Laus1*, *Muutt*) jakaa *Laus1*:n tekijöihin muuttujan *Muutt* suhteen.

Laus1:ä jaetaan tekijöihin mahdollisimman paljon kohti tekijöitä, jotka ovat lineaarisia muuttujassa *Muutt*, sisältäen mahdollisesti ei-reaalisia vakioita, vaikka tästä saataisiin irrationaalisia vakioita tai alalausekkeita, joissa on muita irrationaalisia muuttujia.

Tekijät ja niiden termit lajitellaan siten, että *Muutt* on päämuuttuja. Muuttujan *Muutt* samanlaiset potenssit kerätään jokaisessa tekijässä. Muuttujan *Muutt* tulee olla mukana, jos vain kyseistä muuttujaa halutaan jakaa tekijöihin ja jos irrationaalilausekkeet ovat hyväksyttäviä kaikissa muissa muuttujissa, jotta muuttujaa *Muutt* voitaisiin jakaa enemmän tekijöihin. Toimenpiteessä voi esiintyä jonkin verran satunnaista muiden muuttujien tekijöihin jakamista.

Auto or Approximate (Automaattinen tai likimääräinen) -tilan Auto (Automaattinen) -asetuksessa muuttujan *Muutt* mukanaolo sallii myös likiarvoistamisen liukulukuvakioilla, kun irrationaalisia kertoimia ei voida ilmaista täsmällisen tiiviisti sisäänrakennetuilla termeillä. Vaikka muuttujia olisi vain yksi, muuttujan *Muutt* mukanaolo voi tuottaa täydellisemmän tekijöihin jakamisen.

Huomaa: Katso myös **factor()**.

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

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

Jos haluat nähdä koko vastauksen, paina ► ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ►.

char()

Katalogi > 

char(*Kokonaisluku*) ⇒ *merkki*

Näyttää vastauksena merkkijonon, joka sisältää kämmenlaitteen merkkisarjasta olevan merkin, jonka tunnusnumero on *Kokonaisluku*. Kokonaisluvun *Kokonaisluku* sallittu alue on 0–65535.

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

charPoly(neliömatriisi,Muutt) $\Rightarrow$ polynomilauseke**charPoly(neliömatriisi,Laus)** $\Rightarrow$ polynomilauseke**charPoly(neliömatriisi1,Matriisi2)** $\Rightarrow$ polynomilauseke

Laskee *neliömatriisin* karakteristisen polynomin. Lausekkeen $n \times n$ matriisi A karakteristinen polynomi, merkitään $p_A(\lambda)$, on polynomi, joka on määritetty lausekkeella

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

jossa I tarkoittaa identtistä matriisia $n \times n$.

neliömatriisi1:n ja *neliömatriisi2*:n on oltava samankokoiset.

$m := \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$
charPoly(m, x)	$-x^3 + 5 \cdot x^2 + 7 \cdot x - 35$
charPoly($m, x^2 + 1$)	$-x^6 + 2 \cdot x^4 + 14 \cdot x^2 - 24$
charPoly(m, m)	0

 χ^2 way **χ^2 way ObsMatriisi****chi22way ObsMatriisi**

Laskee χ^2 -testin tarkasteltavan matriisin *ObsMatriisi* sisältämän kaksisuuntaisen lukemataulukon arvojen välisestä assosiaatiosta. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Lisätietoja matriisissa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat. χ^2	Khin neliö -tilasto: summa (tarkasteltava - odotettu) ² /odotettu
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat.df	Khin neliö -tilastojen vapausasteet
stat.ExpMat	Odotetun elementtilukemataulukon matriisi, oletuksena nollahypoteesi
stat.CompMat	Elementtien Khin neliö -tilastokontribuutioiden matriisi

$\chi^2\text{Cdf}(\text{alaraja}, \text{yläraja}, \text{df}) \Rightarrow \text{luku}$, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

$\text{chi2Cdf}(\text{alaraja}, \text{yläraja}, \text{df}) \Rightarrow \text{luku}$, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

Laskee χ^2 -jakauman todennäköisyyden *alarajan* ja *ylärajan* väliltä määritetyille vapausasteelle *df*.

Kun $P(X \leq \text{yläraja})$, aseta *alaraja* = 0.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

$\chi^2\text{GOF } \text{obsLista}, \text{expLista}, \text{df}$

$\text{chi2GOF } \text{obsLista}, \text{expLista}, \text{df}$

Suorittaa testin, jolla varmistetaan, että otoksen data on tiettyä jakaumaa vastaavasta perusjoukosta. *obsList* on lukemalista, ja sen tulee sisältää kokonaislukuja. Tulosten yhteenvedo tallentuu *stat.results*-muuttuun. (Katso sivu 186.)

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat. χ^2	Khin neliö -tilasto: $\text{sum}((\text{tarkasteltava} - \text{odotettu})^2 / \text{odotettu})$
stat.PVal	Alin merkitsevyydestaso, jolla nollahypoteesi voidaan hylätä
stat.df	Khin neliö -tilastojen vapausasteet
stat.Complist	Elementtien Khin neliö -tilastokontribuutiot

$\chi^2\text{Pdf}(XArvo,df)\Rightarrow$ luku, jos *XArvo* on luku,
lista, jos *XArvo* on lista

chi2Pdf(*XArvo*,*df*) $\Rightarrow$ luku, jos *XArvo* on
luku, *lista*, jos *XArvo* on lista

Laskee χ^2 -jakauman
todennäköisyysfunktio (pdf)
määritetyllä *XArvo* arvolla määritetylle
vapausasteelle *df*.

Lisätietoja listassa olevien tyhjien
elementtien vaikutuksesta, katso Tyhjät
elementitsivulla sivu 249.

clearAZKatalogi > **clearAZ**

$5 \rightarrow b$	5
<i>b</i>	5
ClearAZ	Done
<i>b</i>	<i>b</i>

Poistaa kaikki yksikirjaimiset muuttujat
nykyiseltä tehtäväalueelta.

Jos yksi tai useampia muuttujia on lukittu,
tämä komento aiheuttaa virheilmoituksen
ja poistaa vain lukitsemattomat muuttujat.
Katso **unLock**, sivu 208.

ClrErrKatalogi > **ClrErr**

Esimerkki **ClrErr**-komennosta, katso
esimerkki 2 **Try**-komennon kohdalla, sivu
201.

Poistaa virhetilan ja nollaa järjestelmän
muuttujan *errCode* .

Else-lauseessa lohossa **Try...Else...EndTry**
tulee käyttää komentoa **ClrErr** tai **PassErr**.
Jos virhe on tarkoitus käsitellä tai jättää
huomiotta, käytä komentoa **ClrErr**. Jos et
tiedä, mitä tehdä virheen suhteen, lähetä se
seuraavaan virheenkäsittelijään käyttämällä
komentoa **PassErr**. Jos odottavia
Try...Else...EndTry-virheenkäsittelijöitä ei ole
enää, virheen valintaikkuna tulee näkyviin
normaalisti.

Huomaa: Katso myös **PassErr**, sivu 138, ja
Try, sivu 201.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

colAugment()Katalogi > 

colAugment(*Matriisi1*, *Matriisi2*)
 $\Rightarrow$ *matriisi*

Luo uuden matriisin, joka on *Matriisi2* liitettyä *Matriisi1*:een. Matriiseiden sarakemäärän on oltava sama, ja *Matriisi2* liitetään *Matriisi1*:een uusina riveinä. Ei muuta *Matriisi1*:ä eikä *Matriisi2*:a.

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

colDim()Katalogi > 

colDim(*Matriisi*) $\Rightarrow$ *lauseke*

Laskee *Matriisin* sisältämien sarakkeiden lukumäärän.

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

Huomaa: Katso myös **rowDim()**.

colNorm()Katalogi > 

colNorm(*Matriisi*) $\Rightarrow$ *lauseke*

Laskee maksimiarvon *Matriisin* sarakkeissa olevien elementtien itseisarvojen summista.

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

Huomaa: Määrittämättömät matriiselementit eivät ole sallittuja. Katso myös **rowNorm()**.

comDenom(*Laus l* [, *Muutt*]) ⇒ *lauseke*

comDenom(*Lista l* [, *Muutt*]) ⇒ *lista*

comDenom(*Matriisi l* [, *Muutt*]) ⇒ *matriisi*

comDenom(*Laus l*) supistaa täydellisesti lavennetun osoittajan täydellisesti lavennetulla nimittäjällä.

comDenom(*Laus l*, *Muutt*) supistaa osoittajan ja nimittäjän, jotka on lavennettu muuttujalla *Muutt*. Termit ja niiden tekijät lajitellaan siten, että *Muutt* on päämuuttuja. Muuttujan *Muutt* samanlaiset potenssit kerätään. Toimenpiteessä voi esiintyä jonkin verran kerättyjen kertoimien satunnaista tekijöihin jakamista. Verrattuna siihen, että muuttuja *Muutt* jätettäisiin pois, tämä toiminto säästää usein aikaa, muistia ja näyttötilaa, ja samalla lausekkeesta tulee ymmärrettävämpi. Lisäksi tulokseen kohdistuvat seuraavat operaatiot ovat nopeampia eivätkä kuluta muistia yhtä todennäköisesti.

Jos muuttujaa *Muutt* ei ole *Laus l*:ssä, **comDenom**(*Laus l*, *Muutt*) supistaa laventamattoman osoittajan laventamattomalla nimittäjällä. Tällaiset tulokset säästävät yleensä vielä enemmän aikaa, muistia ja näyttötilaa. Tällaiset osittain tekijöihin jaetut tulokset nopeuttavat myös seuraavia tulokseen kohdistuvia operaatioita eivätkä kuluta muistia läheskään yhtä todennäköisesti.

Vaikka nimittäjää ei olisi, **comden**-funktio on usein nopea tapa suorittaa osittainen tekijöihin jako, mikäli **factor()** on liian hidas tai käyttää liikaa muistia.

Vinkki: Syötä tämä **comden()**-funktion määritys ja kokeile sitä rutiinomaisesti vaihtoehtona funktioille **comDenom()** ja **factor()**.

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

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

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

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

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

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

completeSquare ()

Katalogi > 

completeSquare(ExprOrEqn, Var)

⇒lauseke tai yhtälö

completeSquare(ExprOrEqn, Var^Power)

⇒lauseke tai yhtälö

completeSquare(ExprOrEqn, Var1, Var2

[...])⇒lauseke tai yhtälö

completeSquare(ExprOrEqn, {Var1, Var2

[...])⇒lauseke tai yhtälö

Muuntaa muotoa $a \cdot x^2 + b \cdot x + c$ olevan toisen asteen polynomilausekkeen muotoon $a \cdot (x-h)^2 + k$

- tai -

Muuntaa muotoa $a \cdot x^2 + b \cdot x + c$ olevan toisen asteen yhtälön muotoon $a \cdot (x-h)^2 + k$

Ensimmäisen argumentin on oltava toisen asteen lauseke tai yhtälö vakio muodossa toisen argumentin suhteen.

Toisen argumentin on oltava yhden muuttujan termi tai yhden muuttujan termi korotettuna rationaaliseen potenssiin x , y^2 tai $z^{(1/3)}$.

Kolmas ja neljäs syntaksi yrittävät neliöksi täydentämisen muuttujien Var1, Var2 [...]) suhteen.

$\text{completeSquare}(x^2+2 \cdot x+3, x)$	$(x+1)^2+2$
$\text{completeSquare}(x^2+2 \cdot x=3, x)$	$(x+1)^2=4$
$\text{completeSquare}(x^6+2 \cdot x^3+3, x^3)$	$(x^3+1)^2+2$
$\text{completeSquare}(x^2+4 \cdot x+y^2+6 \cdot y+3=0, x, y)$	$(x+2)^2+(y+3)^2=10$
$\text{completeSquare}(3 \cdot x^2+2 \cdot y+7 \cdot y^2+4 \cdot x=3, \{x, y\})$	$3 \cdot \left(x+\frac{2}{3}\right)^2+7 \cdot \left(y+\frac{1}{7}\right)^2=\frac{94}{21}$
$\text{completeSquare}(x^2+2 \cdot x \cdot y, x, y)$	$(x+y)^2-y^2$

conj()

Katalogi > 

conj(LausI)⇒lauseke

conj(ListaI)⇒lista

conj(MatriisiI)⇒matriisi

Laskee argumentin liittokompleksiluvun.

Huomaa: Kaikkia määrittämättömiä muuttujia käsitellään reaali muuttujina.

$\text{conj}(1+2 \cdot i)$	$1-2 \cdot i$
$\text{conj}\left(\begin{bmatrix} 2 & 1-3 \cdot i \\ -i & -7 \end{bmatrix}\right)$	$\begin{bmatrix} 2 & 1+3 \cdot i \\ i & -7 \end{bmatrix}$
$\text{conj}(z)$	z
$\text{conj}(x+i \cdot y)$	$x-y \cdot i$

constructMat()

Katalogi >

constructMat

(

Laus

,Muutt1,Muutt2,numRivit,numSarakkeet)

=>matriisi

constructMat

$\left(\frac{1}{i+j}, i, j, 3, 4\right)$

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

Laskee matriisin argumentteihin perustuen.

Laus on lauseke muuttujissa *Muutt1* ja *Muutt2*. Tuloksena olevan matriisin elementit muodostetaan sieventämällä *Laus* jokaisella *Muutt1*:n ja *Muutt2*:n lisätyllä arvolla.

Muutt1:ä lisätään automaattisesti välillä **1 - numRivit**. Kullakin rivillä *Muutt2*:a lisätään välillä **1 - numSarakkeet**.

CopyVar

Katalogi >

CopyVar Muutt1, Muutt2

Define $a(x)=\frac{1}{x}$

Done

CopyVar Muutt1., Muutt2.

Define $b(x)=x^2$

Done

CopyVar Muutt1, Muutt2 kopioi muuttujan Muutt1 arvon muuttujaan Muutt2 ja luo tarvittaessa Muutt2:n. Muuttujalla Muutt1 on oltava arvo.

CopyVar a,c: c(4)

$\frac{1}{4}$

CopyVar b,c: c(4)

16

Jos *Muutt1* on olemassa olevan käyttäjän määrittämän funktion nimi, kopioi kyseisen funktion määrittymisen funktioon *Muutt2*. Funktio *Muutt1* on määritettävä.

Muutt1:n on oltava muuttujien nimeämissääntöjen mukainen tai epäsuora lauseke, joka sieventyy näitä vaatimuksia vastaavaksi muuttujan nimeksi.

CopyVar Muutt1., Muutt2. kopioi kaikki Muutt1:n jäsenet. muuttujaryhmä Var2:een. ryhmä, Muutt2:n luominen. tarvittaessa.

aa.a:=45

45

aa.b:=6.78

6.78

CopyVar aa.,bb.

Done

getVarInfo()

aa.a

"NUM"

0

aa.b

"NUM"

0

bb.a

"NUM"

0

bb.b

"NUM"

0

Muutt1. tulee olla olemassa olevan muuttujaryhmän nimi, kuten tilastollinen *stat.mn* vastausta tai muuttujaa, jotka on luotu funktiolla **LibShortcut()**. Jos *Muutt2.* on jo olemassa, komento korvaa kaikki jäsenet, jotka ovat yhteisiä kummallekin ryhmälle, ja lisää jäsenet, joita ei vielä ole olemassa. Jos yksi tai useampia muuttujan *Muutt2.* jäseniä on lukittu, kaikki muuttujan *Var2.* jäsenet pysyvät muuttumattomina.

corrMat()

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

Laskee korrelaatiomatriisin laajennetulle matriisille [*Lista1*, *Lista2*, ..., *Lista20*].

► cos

Laus ► **cos**

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>**cos**.

Näyttää *Laus*:n kulman kosinin. Tämä on näytön muunnosoperaattori. Sitä voidaan käyttää vain syöterivin lopussa.

► **cos** alentaa kaikkia lausekkeen

$\sin(\dots)$ modulo $1 - \cos(\dots)^2$

potensseja, siten että jäljelle jäävien lausekkeen $\cos(\dots)$ potenssien eksponentit ovat alueella (0, 2). Tulos ei täten sisällä lauseketta $\sin(\dots)$, jos ja vain jos $\sin(\dots)$ esiintyy lausekkeessa korotettuna vain parillisiin potensseihin.

Huomaa: Tätä muunnosoperaattoria ei tueta aste- eikä graadikulmatilassa. Ennen kuin käytät sitä, varmista, että kulmatila on asetettu radiaaneiksi ja että *Laus* ei sisällä eksplisiittisiä viittauksia aste- tai graadikulmiin.

$\{\sin(x)\}^2 \blacktriangleright \cos$	$1 - \{\cos(x)\}^2$
--	---------------------

cos(Lausl)⇒*lauseke*

cos(Listal)⇒*lista*

cos(Lausl) määrittää argumentin kosinin lausekkeena.

cos(Listal) määrittää listan kaikkien *Listal*:n sisältämien elementtien kosineista.

Huomaa: Argumentti tulkitaan aste-, graadi- tai radiaanikulmaksi käytössä olevan kulmatila-asetuksen mukaisesti. Voit ohittaa kulmatilan väliaikaisesti painikkeilla °, ° tai °.

Astekulmatilassa:

$$\begin{array}{l} \cos\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2} \\ \cos(45) \quad \frac{\sqrt{2}}{2} \\ \cos(\{0,60,90\}) \quad \left\{1, \frac{1}{2}, 0\right\} \end{array}$$

Graadikulmatilassa:

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

Radiaanikulmatilassa:

$$\begin{array}{l} \cos\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2} \\ \cos(45^\circ) \quad \frac{\sqrt{2}}{2} \end{array}$$

cos(neliömatriisil)⇒*neliömatriisi*

Laskee *neliömatriisil*:n matriisikosinin. Tämä ei ole sama kuin kunkin elementin kosinin laskeminen.

Kun skaalarista funktiota *f(A)* käytetään *neliömatriisil*:een (A), tulos lasketaan algoritmilla:

Laske A:n ominaisarvot (λ_i) ja ominaisvektorit (V_i).

neliömatriisil:n on oltava diagonalisoitavissa. Lisäksi siinä ei voi olla symbolisia muuttujia, joille ei ole määritetty arvoa.

Radiaanikulmatilassa:

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

Matriiseista:

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

Tällöin $A = X B X^{-1}$ ja $f(A) = X f(B) X^{-1}$.

Esimerkiksi, $\cos(A) = X \cos(B) X^{-1}$, jossa:

$\cos(B) =$

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

Kaikki laskut suoritetaan liukulukuaritmetiikalla.

$\cos^{-1}(Laus\ I) \Rightarrow$ lauseke

Astekulmatilassa:

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

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

$\cos^{-1}(Laus\ I)$ määrittää lausekkeena kulman, jonka kosini on $Laus\ I$.

Graadikulmatilassa:

$\cos^{-1}(Listal\ I)$ laskee listan $Listal\ I$:n jokaisen elementin käänteiskosineista.

$$\cos^{-1}(0) \quad 100$$

Huomaa: Vastaus lasketaan aste-, graadi- tai radiaanikulmana käytössä olevan kulmatila-asetuksen mukaisesti.

Radiaanikulmatilassa:

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arccos** (...).

$\cos^{-1}(neliömatrissi\ I) \Rightarrow$ neliömatrissi

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

Laskee *neliömatrissi\ I*:n matriisin käänteiskosinin. Tämä ei ole sama kuin kunkin elementin käänteiskosinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

$\cos^{-1}()$

 **painike**

neliömatriisi *l*:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

$\cosh()$

Katalogi > 

$\cosh(\text{Laus } l) \Rightarrow \text{lauseke}$

Astekulmatilassa:

$\cosh(\text{Lista } l) \Rightarrow \text{lista}$

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

$\cosh(\text{Laus } l)$ määrittää argumentin hyperbolisen kosinin lausekkeena.

$\cosh(\text{Lista } l)$ määrittää listan *Lista* *l*:n kunkin elementin hyperbolisista kosineista.

$\cosh(\text{neliömatriisi } l) \Rightarrow \text{neliömatriisi}$

Radiaanikulmatilassa:

Laskee *neliömatriisi* *l*:n matriisin hyperbolisen kosinin. Tämä ei ole sama kuin kunkin elementin hyperbolisen kosinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

neliömatriisi *l*:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

$\cosh^{-1}()$

Katalogi > 

$\cosh^{-1}(\text{Laus } l) \Rightarrow \text{lauseke}$

$$\cosh^{-1}(1) \quad 0$$

$\cosh^{-1}(\text{Lista } l) \Rightarrow \text{lista}$

$$\cosh^{-1}\left(\left\{1, 2, 1, 3\right\}\right) \quad \left\{0, 1.37286, \cosh^{-1}(3)\right\}$$

$\cosh^{-1}(\text{Laus } l)$ määrittää argumentin käänteisen hyperbolisen kosinin lausekkeena.

$\cosh^{-1}(\text{Lista } l)$ määrittää listan *Lista* *l*:n kunkin elementin käänteisistä hyperbolisista kosineista.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arccosh** (...).

cosh⁻¹()Katalogi > **cosh⁻¹(*neliömatriisi*)** ⇒ *neliömatriisi*

Laskee *neliömatriisi* *l*:n matriisin käänteisen hyperbolisen kosinin. Tämä ei ole sama kuin kunkin elementin käänteisen hyperbolisen kosinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisi *l*:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

$$\cosh^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} 2.52503+1.73485 \cdot i & -0.009241-1.49086 \cdot i & 0.486969-0.725533 \cdot i \\ -0.322354-2.08316 \cdot i & 1.66262+0.623491 \cdot i & 1.26707+1.79018 \cdot i \end{bmatrix}$$

Jos haluat nähdä koko vastauksen, paina ► ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ►.

cot() **painike****cot(*Laus* *l*)** ⇒ *lauseke*

Astekulmatilassa:

$$\cot(45) = 1$$

cot(*Lista* *l*) ⇒ *lista*

Laskee *Laus* *l*:n kotangentin tai määrittää listan *Lista* *l*:n kaikkien elementtien kotangenteista.

Graadikulmatilassa:

$$\cot(50) = 1$$

Huomaa: Argumentti tulkitaan aste-, graadi- tai radiaanikulmaksi käytössä olevan kulmatila-asetuksen mukaisesti. Voit ohittaa kulmatilan väliaikaisesti painikkeilla °, G tai r.

Radiaanikulmatilassa:

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

cot⁻¹() **painike****cot⁻¹(*Laus* *l*)** ⇒ *lauseke*

Astekulmatilassa:

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

cot⁻¹(*Lista* *l*) ⇒ *lista*

Laskee kulman, jonka kotangenti on *Laus* *l*, tai määrittää listan, joka sisältää *Lista* *l*:n kunkin elementin käänteiskotangentit.

Graadikulmatilassa:

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

Huomaa: Vastaus lasketaan aste-, graadi- tai radiaanikulmana käytössä olevan kulmatila-asetuksen mukaisesti.

Radiaanikulmatilassa:

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arccot(...)**.

cot ⁻¹ (1)	$\frac{\pi}{4}$
-----------------------	-----------------

coth()**Katalogi** > **coth**(*Laus l*)⇒*lauseke*

coth(1.2)	1.19954
-----------	---------

coth(*Listal*)⇒*lista*

coth({ 1,3.2 })	$\left\{ \frac{1}{\tanh(1)}, 1.00333 \right\}$
-----------------	--

Laskee *Laus l*:n hyperbolisen kotangentin tai määrittää listan *Listal*:n kaikkien elementtien hyperbolisista kotangenteista.

coth⁻¹()**Katalogi** > **coth⁻¹**(*Laus l*)⇒*lauseke*

coth ⁻¹ (3.5)	0.293893
--------------------------	----------

coth⁻¹(*Listal*)⇒*lista*

coth ⁻¹ ({-2,2.1,6})	$\left\{ \frac{-\ln(3)}{2}, 0.518046, \frac{\ln\left(\frac{7}{5}\right)}{2} \right\}$
---------------------------------	---

Laskee *Laus l*:n käänteisen hyperbolisen kotangentin tai määrittää listan, joka sisältää *Listal*:n kaikkien elementtien käänteiset hyperboliset kotangentin.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arccoth** (...).

count()**Katalogi** > **count**(*Arvo1taiListal* [,*Arvo2taiListal2* [...]])⇒*arvo*

count(2,4,6)	3
--------------	---

count({ 2,4,6 })	3
------------------	---

Laskee elementtien kokonaismäärän argumenteille, jotka sievennyvät numeroarvoiksi.

count(2,{ 4,6 }, $\begin{bmatrix} 8 & 10 \\ 12 & 14 \end{bmatrix}$)	7
--	---

Argumentit voivat olla lausekkeita, arvoja, listoja tai matriiseja. Argumenttien datatyytit voivat olla erilaisia, ja argumentit voivat olla erikokoisia.

count($\frac{1}{2}$,3+4 <i>i</i> ,undef,"hello",x+5,sign(0))	2
--	---

Viimeisessä esimerkissä lukumäärään lasketaan mukaan vain 1/2 ja 3+4*i*. Muut argumenteista, olettaen että *x* on määrittämätön, eivät sievenny numeroarvoiksi.

Listan, matriisin tai solualueen jokainen elementti sievennetään, jotta voidaan määrittää, kuuluuko se laskettavaan lukumäärään.

Listat & Taulukot -sovelluksessa voit käyttää solualueita argumenttien tilalla.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementteistä, katso sivu 249.

countif()

countif(Lista,Kriteerit)⇒arvo

Laskee niiden *Listan* sisältämien elementtien kokonaismäärän, jotka vastaavat määritettyjä kriteereitä *Kriteerit*.

Kriteeri voi olla:

- Arvo, lauseke tai merkkijono. Jos kriteerinä käytetään esimerkiksi lukua **3**, laskee lukumäärään vain ne *Listan* elementit, jotka sieventyvät arvoksi 3.
- Boolean lauseke, joka sisältää symbolin ? kunkin elementin paikanpitäjänä. Esimerkiksi lauseke **?<5** laskee lukumäärään vain ne *Listan* elementit, jotka ovat alle 5.

Listat & Taulukot -sovelluksessa voit käyttää solualueita *Listan* tilalla.

Listassa olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

Huomaa: Katso myös **sumlf()**, sivu 190, ja **frequency()**, sivu 79.

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

Laskee niiden elementtien lukumäärän, jotka ovat yhtä kuin 3.

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

Laskee niiden elementtien lukumäärän, jotka ovat yhtä kuin "def".

countIf({x⁻²,x⁻¹,1,x,x²},x) 1

Laskee niiden elementtien lukumäärän, jotka ovat yhtä kuin x; tässä esimerkissä oletetaan, että muuttuja x on määrittämätön.

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

Laskee lukumäärään 1:n ja 3:n.

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

Laskee lukumäärään 3:n, 5:n ja 7:n.

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

Laskee lukumäärään 1:n, 3:n, 7:n ja 9:n.

cPolyRoots()Katalogi > **cPolyRoots**(*Poly*,*Muutt*) $\Rightarrow$ *lista*

$$\text{polyRoots}(y^3+1,y) \quad \{-1\}$$

cPolyRoots(*Kertoinlista*) $\Rightarrow$ *lista*

$$\text{cPolyRoots}(y^3+1,y) \\ \left\{-1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i\right\}$$

Ensimmäinen syntaksi, **cPolyRoots**(*Poly*,*Muutt*), laskee polynomin *Poly* kompleksisten juurten listan muuttujan *Muutt* suhteen.

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

Poly on oltava polynomi yhdessä muuttujassa.

$$\text{cPolyRoots}(\{1,2,1\}) \quad \{-1,-1\}$$

Toinen syntaksi, **cPolyRoots**(*Kertoinlista*), laskee kompleksisten juurten listan kertoimille, jotka sisältyvät *Kertoinlistaan*.

Huomaa: Katso myös **polyRoots()**, sivu 143.

crossP()Katalogi > **crossP**(*Lista1*, *Lista2*) $\Rightarrow$ *lista*

$$\text{crossP}(\{a1,b1\},\{a2,b2\}) \\ \{0,0,a1\cdot b2-a2\cdot b1\}$$

Määrittää listan *Lista1*:n ja *Lista2*:n ristitulosta.

$$\text{crossP}(\{0.1,2.2,-5\},\{1,-0.5,0\}) \\ \{-2.5,-5,-2.25\}$$

Lista1:n ja *Lista2*:n on oltava samankokoiset, ja koon on oltava joko 2 tai 3.

crossP(*Vektori1*, *Vektori2*) $\Rightarrow$ *vektori*

$$\text{crossP}([1 \ 2 \ 3],[4 \ 5 \ 6]) \quad [-3 \ 6 \ -3]$$

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

Laskee rivi- tai sarakvektorin (argumenteista riippuen), joka on *Vektori1*:n ja *Vektori2*:n ristitulo.

Sekä *Vektori1*:n että *Vektori2*:n on oltava rivivektoreita tai sarakvektoreita.

Vektoreiden on oltava samankokoiset, ja koon tulee olla joko 2 tai 3.

csc() **painike****csc**(*Laus1*) $\Rightarrow$ *lauseke*

Astekulmatilassa:

csc(*Lista1*) $\Rightarrow$ *lista*

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

Laskee *Laus1*:n kosekantin tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien kosekantit.

Graadikulmatilassa:

csc() **painike**

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

Radiaanikulmatilassa:

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

csc⁻¹() **painike****csc⁻¹(LausI)** ⇒ lauseke

Astekulmatilassa:

csc⁻¹(Listal) ⇒ lista

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

Laskee kulman, jonka kosekanti on *LausI*, tai määrittää listan, joka sisältää *Listal*:n kunkin elementin käänteiskosekantit.

Graadikulmatilassa:

Huomaa: Vastaus lasketaan aste-, graadi- tai radiaanikulmana käytössä olevan kulmatila-asetuksen mukaisesti.

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

Radiaanikulmatilassa:

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arccsc** (...).

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

csch()**Katalogi** > **csch(LausI)** ⇒ lauseke

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

csch(Listal) ⇒ lista

Laskee *LausI*:n hyperbolisen kosekantin tai määrittää listan, joka sisältää *Listal*:n kaikkien elementtien hyperboliset kosekantit.

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

csch⁻¹()**Katalogi** > **csch⁻¹(LausI)** ⇒ lauseke

$$\text{csch}^{-1}(1) \quad \sinh^{-1}(1)$$

csch⁻¹(Listal) ⇒ lista

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

Laskee *Laus1*:n käänteisen hyperbolisen kosekantin tai määrittää listan, joka sisältää *Listal*:n kaikkien elementtien käänteiset hyperboliset kosekantit.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arccsch** (...).

cSolve()

cSolve(*Yhtälö*, *Muutt*) ⇒ *Boolean lauseke*

cSolve(*Yhtälö*, *Muutt* = *Arvaus*) ⇒ *Boolean lauseke*

cSolve(*Epäyhtälö*, *Muutt*) ⇒ *Boolean lauseke*

$$\begin{array}{l} \text{cSolve}(x^3 = -1, x) \\ x = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ or } x = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } x = -1 \\ \text{solve}(x^3 = -1, x) \quad x = -1 \end{array}$$

Määrittää kompleksiyhtälön tai -epäyhtälön mahdollisia ratkaisuja muuttujalle *Muutt*. Tavoitteena on tuottaa kaikkien reaalisten ja ei-reaalisten ratkaisujen ehdotuksia. Vaikka *Yhtälö* olisi reaalinen, **cSolve()** sallii ei-reaaliset vastaukset reaalituloksen kompleksilukumuodossa.

cSolve() asettaa määritysjoukon väliaikaisesti kompleksilukumuotoon yhtälön ratkaisemisen ajaksi, vaikka nykyinen määritysjoukko olisi reaalinen. Kompleksilukujen määritysalueella murtolukueksponenteissa, joiden nimittäjä on pariton luku, käytetään perus- eikä reaaliilukualuetta. Tämän vuoksi **solve()**-funktion ratkaisut yhtälöille, joihin liittyy tällaisia murtopotensseja, eivät välttämättä ole **cSolve()**-funktion ratkaisujen alasarja.

cSolve()-funktion ratkaisu aloitetaan eksakteilla symbolisilla menetelmillä. **cSolve()** käyttää tarvittaessa myös iteratiivista likimääräistä kompleksipolynomin tekijöihin jakamista.

Huomaa: Katso myös **cZeros()**, **solve()** ja **zeros()**.

$$\begin{array}{l} \text{cSolve}\left(x^{\frac{1}{3}} = -1, x\right) \quad \text{false} \\ \text{solve}\left(x^{\frac{1}{3}} = -1, x\right) \quad x = -1 \end{array}$$

Desimaalien näyttötulossa Kiinteä 2:

$$\begin{array}{l} \text{exact}\left(\text{cSolve}\left(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3 = 0, x\right)\right) \\ x \cdot \left(x^4 + 4 \cdot x^3 + 5 \cdot x^2 - 6\right) = 3 \\ \text{cSolve}(\text{Ans}, x) \\ x = 1.11 + 1.07 \cdot i \text{ or } x = -1.11 - 1.07 \cdot i \text{ or } x = -2. \end{array}$$

Jos haluat nähdä koko vastauksen, paina ▲
ja siirrä sen jälkeen osoitinta painikkeilla
◀ ja ▶.

cSolve(*Yht1* and *Yht2* [and...],
MuuttTaiArvaus1, *MuuttTaiArvaus2* [, ...
) ⇒ *Boolean lauseke*

cSolve(*Yhtälöryhmä*, *MuuttTaiArvaus1*,
MuuttTaiArvaus2 [, ...]) ⇒ *Boolean lauseke*

Laskee mahdollisia kompleksiratkaisuja samanaikaisille algebrallisille yhtälöille, joissa jokainen *MuuttTaiArvaus* määrittää ratkaistavan muuttujan.

Voit halutessasi määrittää muuttujan ensimmäisen arvauksen. Jokaisen *muuttTaiArvaus*-komennon on oltava muodossa:

muuttuja

– tai –

muuttuja = *reaaliluku* tai *ei-reaaliluku*

Esimerkiksi x kelpaa ja samoin $x=3+i$.

Jos kaikki yhtälöt ovat polynomeja, ja jos ET määritä yhtään ensimmäistä arvausta, **cSolve()** käyttää leksikaalista Gröbner/Buchbergerin eliminaatiomenetelmää yrittäessään määrittää **kaikki** kompleksiratkaisut.

Kompleksiratkaisut voivat sisältää sekä reaali- että ei-reaaliratkaisuja kuten oikealla olevassa esimerkissä.

$$\text{cSolve}\left(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v\}\right) \\ u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } u = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i$$

Jos haluat nähdä koko vastauksen, paina ▲
ja siirrä sen jälkeen osoitinta painikkeilla
◀ ja ▶.

Samanaikaisissa polynomi yhtälöissä voi olla ylimääräisiä muuttujia, joilla ei ole arvoja, vaan ne edustavat tiettyjä numeerisia arvoja, jotka voidaan korvata myöhemmin.

$$\text{cSolve}\left(u \cdot v - u = c \cdot v \text{ and } v^2 = -u, \{u, v\}\right) \\ u = \frac{-(\sqrt{4 \cdot c - 1} \cdot i + 1)^2}{4} \text{ and } v = \frac{\sqrt{4 \cdot c - 1} \cdot i + 1}{2}$$

Voit ottaa mukaan myös ratkaisumuuttujia, jotka eivät esiinny yhtälöissä. Nämä ratkaisut osoittavat, miten ratkaisujen sarjat voivat sisältää mielivaltaisia vakioita, jotka ovat muotoa ck , jossa k on kokonaislukuliite väliltä 1-255.

Polynomisarjoissa laskun suoritus aika tai muistin käyttö voivat riippua merkittävästi ratkaisumuuttujien järjestyksestä. Jos ensimmäinen valintasi kuluttaa muistia, tai et jaksa odottaa vastausta, yritä järjestää muuttujat uudelleen yhtälöihin ja/tai *muuttTaiArvaus*-listaan.

Jos et ota mukaan arvauksia, ja jos jokin yhtälöistä on ei-polynominen minkä tahansa muuttujan suhteen, mutta kaikki yhtälöt ovat lineaarisia kaikissa ratkaisumuuttujissa, **cSolve()** käyttää Gaussin eliminointia yrittäessään määrittää kaikki ratkaisut.

Jos sarja ei ole polynominen kaikilta muuttujiltaan eikä lineaarinen muuttujiltaan, **cSolve()** määrittää korkeintaan yhden ratkaisun käyttäen likimääräistä iteratiivista menetelmää. Tässä ratkaisumuuttujien lukumäärän on oltava sama kuin yhtälöiden lukumäärä, ja kaikkien muiden yhtälöiden sisältämien muuttujien on sievennyttävä luvuiksi.

Ei-reaalinen arvaus on usein välttämätön ei-reaalisen ratkaisun määrittämiseksi. Suppenemista varten arvauksen on mahdollisesti oltava melko lähellä ratkaisua.

$$\text{cSolve}\left(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v, w\}\right) \\ u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } w = c43 \text{ or}$$

$$\text{cSolve}\left(u + v = e^w \text{ and } u - v = i, \{u, v\}\right) \\ u = \frac{e^w + i}{2} \text{ and } v = \frac{e^w - i}{2}$$

$$\left| \text{cSolve}\left(e^z = w \text{ and } w = z^2, \{w, z\}\right) \right. \\ \left. w = 0.494866 \text{ and } z = 0.703467 \right|$$

$$\left| \text{cSolve}\left(e^z = w \text{ and } w = z^2, \{w, z = 1 + i\}\right) \right. \\ \left. w = 0.149606 + 4.8919 \cdot i \text{ and } z = 1.58805 + 1.5402 \cdot i \right|$$

Jos haluat nähdä koko vastauksen, paina ▲
ja siirrä sen jälkeen osoitinta painikkeilla
◀ ja ▶.

CubicReg $X, Y, [Frekv] [, Luokka, Sisällyttä]$

Laskee 3. asteen polynomiregression $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$ listaista X ja Y frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja ≥ 0 .

Luokka on luokkakoodien lista vastaavalle X - ja Y -datalle.

Sisällytä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjet elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$.
stat.a, stat.b, stat.c, stat.d	Regressiokertoimet.
stat.R ²	Määrityskerroin.
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatus X Lista:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat</i> rajoitusten mukaisesti.
stat.YReg	Muokatus Y Lista:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat</i> rajoitusten mukaisesti.
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista.

cumulativeSum()

Katalogi > 

cumulativeSum(Lista1)⇒lista

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

Laskee listan *Listal*:n sisältämien elementtien kumulatiivisista summista alkaen elementistä 1.

cumulativeSum(Matriisi1)⇒matriisi

Laskee matriisin *Matriisi1*:n sisältämien elementtien kumulatiivisista summista. Jokainen elementti on ylhäältä alas ulottuvan sarakkeen kumulatiivinen summa.

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

Tyhjä elementti listassa *Listal* tai matriisissa *Matriisi1* tuottaa tyhjän elementin tuloksena olevaan listaan tai matriisiin. Lisätietoja tyhjästä elementeistä, katso sivu 249.

Cycle

Katalogi > 

Cycle

Siirtää ohjauksen välittömästi nykyisen silmukan (**For**, **While** tai **Loop**) seuraavaan iteraatioon.

Cycle ei ole sallittu näiden kolmen silmukkarakenteen (**For**, **While** tai **Loop**) ulkopuolella.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määrittysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Funktio, joka laskee yhteen kokonaisluvut väliltä 1-100 ohittaen luvun 50.

```
Define g() $\equiv$ Func                                     Done
    Local temp,i
    0→temp
    For i,1,100,1
    If i=50
    Cycle
    temp+i→temp
    EndFor
    Return temp
    EndFunc

g()
```

5000

►Cylind

Katalogi > 

Vektori ►**Cylind**

$\begin{bmatrix} 2 & 2 & 3 \end{bmatrix}$ ►Cylind $\begin{bmatrix} 2\cdot\sqrt{2} & \angle\frac{\pi}{4} & 3 \end{bmatrix}$

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>**Cylind**.

Näyttää rivi- tai sarakevektorin sylinterin muodossa $[r, \angle\theta, z]$.

Vektorissa on oltava täsmälleen kolme elementtiä. Se voi olla joko rivi tai sarake.

cZeros()

cZeros(Laus, Muutt)⇒lista

Määrittää listan muuttujan *Muutt* mahdollisista reaali- ja ei-reaaliarvoista, joiden tuloksena *Laus*=0. **cZeros()** suorittaa tämän seuraavasti:

explist(cSolve(Laus=0,Muutt),Muutt).

Muulta osin **cZeros()** on samanlainen kuin **zeros()**.

Huomaa: Katso myös **cSolve()**, **solve()** ja **zeros()**.

**cZeros({Laus1, Laus2 [, ...] },
{MuuttTaiArvaus1,MuuttTaiArvaus2 [, ...] })⇒matriisi**

Laskee mahdollisia kohtia, joissa lausekkeet ovat samanaikaisesti nolla. Jokainen *MuuttTaiArvaus* määrittää ratkaistavan tuntemattoman arvon.

Voit halutessasi määrittää muuttujan ensimmäisen arvauksen. Jokaisen *muuttTaiArvaus*-komennon on oltava muodossa:

muuttuja

– tai –

muuttuja = *reaaliluku* tai *ei-reaaliluku*

Esimerkiksi *x* kelpaa ja samoin *x*=3+*i*.

Jos kaikki lausekkeet ovat polynomeja, ja ET määritä ensimmäisiä arvauksia, **cZeros()** käyttää leksikaalista Gröbner/Buchbergerin eliminaatiomenetelmää yrittäessään määrittää **kaikki** kompleksiset nollakohdat.

Desimaalien näyttötilassa Kiinteä 3:

$$\text{cZeros}(x^5+4x^4+5x^3-6x-3,x)$$

$$\{-1.1138+1.07314 \cdot i, -1.1138-1.07314 \cdot i, -2. \}$$

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

Kompleksiset nollakohdat voivat sisältää sekä reaalisia että ei-reaalisia nollakohtia, kuten oikealla olevassa esimerkissä.

Jokainen tulomatriisin rivi edustaa vaihtoehtoista nollakohtaa, jossa komponentit on järjestetty samalla tavalla kuin *MuuttTaiArvaus*-listassa. Jos haluat määrittää rivin juuren, indeksoi matriisi [riveittäin].

Samanaikaisissa polynomeissa voi olla ylimääräisiä muuttujia, joilla ei ole arvoja, vaan ne edustavat tiettyjä numeerisia arvoja, jotka voidaan korvata myöhemmin.

Voit ottaa mukaan myös tuntemattomia muuttujia, jotka eivät esiinny lausekkeissa. Nämä nollakohdat osoittavat, miten nollakohtien sarjat voivat sisältää mielivaltaisia vakioita, jotka ovat muotoa ck , jossa k on kokonaisluku väliltä 1-255.

Polynomisarjoissa laskutoimituksen suoritus aika tai muistin käyttö voivat riippua merkittävästi tuntemattomien muuttujien järjestyksestä. Jos ensimmäinen valintasi kuluttaa muistia, tai et jaksaa odottaa vastausta, yritä järjestää muuttujat uudelleen lausekkeisiin ja/tai *MuuttTaiArvaus*-listaan.

Jos et ota mukaan arvauksia, ja jokin lausekkeista on ei-polynominen missä tahansa muuttujassa, mutta kaikki muuttujat ovat lineaarisia kaikissa tuntemattomissa muuttujissa, **cZeros()** käyttää Gaussin eliminointia yrittäessään määrittää kaikki nollakohdat.

$$cZeros\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \left\{u, v\right\}\right)$$

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

Määritä rivin 2 juuri:

$$Ans[2] \quad \left[\frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \quad \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \right]$$

$$cZeros\left(\left\{u \cdot v - u - c \cdot v^2, v^2 + u\right\}, \left\{u, v\right\}\right)$$

$$\begin{bmatrix} 0 & 0 \\ -(c-1)^2 & -(c-1) \end{bmatrix}$$

$$cZeros\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \left\{u, v, w\right\}\right)$$

$$cZero\left(\left\{u \cdot (v-1) - v, u + v^2\right\}, \left\{u, v, w\right\}\right)$$

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

$$cZeros\left(\left\{u + v - e^w, u - v - i\right\}, \left\{u, v\right\}\right)$$

$$\begin{bmatrix} e^w + i & e^w - i \\ 2 & 2 \end{bmatrix}$$

Jos sarja ei ole polynominen kaikilta muuttujiltaan eikä lineaarinen tuntemattomilta muuttujiltaan, **cZeros()** määrittää korkeintaan yhden nollakohdan käyttäen likimääräistä iteratiivista menetelmää. Tässä tuntemattomien muuttujien lukumäärän on oltava sama kuin lausekkeiden lukumäärä, ja kaikkien muiden lausekkeiden sisältämien muuttujien on sievennyttävä luvuiksi.

$$\left| \begin{array}{c} \text{cZeros}\left(\left\{e^z - w, w - z^2\right\}, \left\{w, z\right\}\right) \\ [0.494866 \quad -0.703467] \end{array} \right|$$

Ei-reaalinen arvaus on usein välttämätön ei-reaalisen nollakohdan määrittämiseksi. Suppenemista varten arvauksen on mahdollisesti oltava melko lähellä nollakohtaa.

$$\left| \begin{array}{c} \text{cZeros}\left(\left\{e^{-z} - w, w - z^2\right\}, \left\{w, z = 1 + i\right\}\right) \\ [0.149606 + 4.8919 \cdot i \quad 1.58805 + 1.54022 \cdot i] \end{array} \right|$$

D

dbd(pvm1,pvm2)⇒arvo

Laskee *pvm1*:n ja *pvm2*:n välissä olevien päivien lukumäärän käyttäen todellisten päivien laskentamenetelmää.

pvm1 ja *pvm2* voivat olla lukuja tai lukulistoja, jotka ovat vakiokalenterin päivämääräalueen sisällä. Jos sekä *pvm1* että *pvm2* ovat listoja, niiden on oltava samanpituiset.

pvm1:n ja *pvm2*:n on oltava vuosien 1950 ja 2049 välillä.

Voit syöttää päivämäärät kahdessa eri muodossa. Desimaalipisteen paikka on erilainen näissä päivämäärien esitystavoissa.

MM.DDYY (Yhdysvalloissa yleisesti käytetty esitystapa)
DDMM.YY (Euroopassa yleisesti käytetty esitystapa)

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

Lausl ►DD⇒*arvoListal*

►DD⇒*listaMatriisi1*

►DD⇒*matriisi*

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>DD.

Laskee vastaavan desimaaliluvun asteina ilmaistulle argumentille. Argumentti on luku, lista tai matriisi, jonka kulmatilasetus tulkitsee graadeina, radiaaneina tai asteina.

Astekulmatilassa:

$\{1.5^\circ\}$ ►DD	1.5°
$\{45^\circ 22' 14.3''\}$ ►DD	45.3706°
$\{\{45^\circ 22' 14.3'', 60^\circ 0' 0''\}\}$ ►DD	$\{45.3706^\circ, 60^\circ\}$

Graadikulmatilassa:

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

Radiaanikulmatilassa:

$\{1.5\}$ ►DD	85.9437°
---------------	----------

►Decimal

Lauseke1►Decimal⇒*lauseke*

Listal ►Decimal⇒*lauseke*

Matriisi1 ►Decimal⇒*lauseke*

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>Decimal.

Näyttää argumentin desimaalimuodossa. Tätä operaattoria voi käyttää ainoastaan syöterivin lopussa.

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

Define (Määritä)

Define *Muutt* = *Lauseke*

Define *Funktio*(*Param1*, *Param2*, ...) = *Lauseke*

Määrittää muuttujan *Muutt* tai käyttäjän määrittämän funktion *Funktio*.

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

Parametrit, kuten *Param1*, toimivat paikanpitäjinä argumenttien syöttämiseksi funktioon. Kun haet käyttäjän määrittämän funktion, sinun on annettava parametreja vastaavat argumentit (esimerkiksi arvoja tai muuttujia). Kun funktio haetaan, se sieventää *Lausekkeen* annettujen argumenttien perusteella.

Muutt ja *Funktio* eivät voi olla järjestelmän muuttujan tai sisäänrakennetun funktion tai komennon nimenä.

Huomaa: Seuraava **Define**-funktion muoto on vastaava kuin lausekkeen sieventäminen: *lauseke* $\rightarrow$ *Funktio*(*Param1*,*Param2*).

Define *Funktio*(*Param1*, *Param2*, ...) =
Func
Lohko
EndFunc

Define *Ohjelma*(*Param1*, *Param2*, ...) =
Prgm
Lohko
EndPrgm

Tässä muodossa käyttäjän määrittämä funktio tai ohjelma voi suorittaa useista lausekkeista koostuvan lohkon.

Lohko voi olla joko yksi lauseke tai eri riveillä olevien lausekkeiden sarja. *Lohko* voi sisältää myös lausekkeita ja ohjeita (kuten **If**, **Then**, **Else** ja **For**).

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Huomaa: Katso myös **Define LibPriv**, sivu 49, ja **Define LibPub**, sivu 50.

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

Define $g(x,y)$ =Prgm	
If $x > y$ Then	
Disp x , " greater than ", y	
Else	
Disp x , " not greater than ", y	
EndIf	
EndPrgm	
	Done
$g(3,-7)$	
	3 greater than -7
	Done

Define LibPriv (Määritä LibPriv)

Define LibPriv *Muutt* = *Lauseke*
Define LibPriv *Funktio*(*Param1*, *Param2*,

... = *Lauseke*

Define LibPriv *Funktio*(*Param1*, *Param2*,

...) = **Func**

Lohko

EndFunc

Define LibPriv *Ohjelma*(*Param1*, *Param2*,

...) = **Prgm**

Lohko

EndPrgm

Tämä komento toimii muuten samalla tavalla kuin **Define** paitsi, että se määrittää yksityisen kirjastomuuttujan, -funktion tai -ohjelman. Yksityiset funktiot ja ohjelmat eivät ole katalogissa.

Huomaa: Katso myös **Define**, sivu 48, ja

Define LibPub, sivu 50.

Define LibPub *Muutt* = *Lauseke*

Define LibPub *Funktio*(*Param1*, *Param2*, ...)

= *Lauseke*

Define LibPub *Funktio*(*Param1*, *Param2*, ...)

= **Func**

Lohko

EndFunc

Define LibPub *Ohjelma* (*Param1*, *Param2*,

...) = **Prgm**

Lohko

EndPrgm

Tämä komento toimii muuten samalla tavalla kuin **Define** paitsi, että se määrittää julkisen kirjastomuuttujan, -funktion tai -ohjelman. Julkiset funktiot ja ohjelmat näkyvät katalogissa sen jälkeen, kun kirjasto on tallennettu ja näyttö on päivitetty.

Huomaa: Katso myös **Define**, sivu 48, ja

Define LibPriv, sivu 49.

DelVar

Katalogi > 

DelVar *Muutt1* [, *Muutt2*] [, *Muutt3*] ...

2 → a

2

DelVar *Muutt.*

(a+2)²

16

Poistaa määritetyn muuttujan tai muuttujaryhmän muistista.

DelVar *a*

Done

(a+2)²

(a+2)²

Jos yksi tai useampia muuttujia on lukittu, tämä komento aiheuttaa virheilmoituksen ja poistaa vain lukitsemattomat muuttujat. Katso **unLock**, sivu 208.

DelVar *Muutt.* poistaa kaikki *Muutt:n* jäsenet. muuttujaryhmä (kuten tilastollinen *stat.nn* tulosta tai muuttujaa, jotka on luotu funktiolla **LibShortcut()**). Piste (.) tässä **DelVar**-komennon muodossa rajoittaa funktion muuttujaryhmän poistamiseen; komento ei vaikuta yksinkertaiseen muuttujaan *Muutt.*

aa.a:=45

45

aa.b:=5.67

5.67

aa.c:=78.9

78.9

getVarInfo()

aa.a "NUM" 

aa.b "NUM" 

aa.c "NUM" 

DelVar *aa.*

Done

getVarInfo()

"NONE"

delVoid()	Katalogi > 	
delVoid(<i>Listal</i>) $\Rightarrow$ <i>lista</i>	delVoid({1,void,3})	{1,3}
Antaa tuloksena listan, jossa on listan <i>Listal</i> sisältö, ja kaikki tyhjät elementit on poistettu.		
Lisätietoja tyhjistä elementeistä, katso sivu 249.		

deSolve(1.astTai2.astODE, Muutt, riippuvaMuutt) ⇒ yleinen ratkaisu

Ratkaisee yhtälön, joka määrittää eksplisiittisesti tai implisiittisesti yleisratkaisun 1. tai 2. asteen tavalliselle differentiaaliyhtälölle (ODE). ODE:ssa:

- Käytä jaottoman merkkiä (näppäin ) viittaamaan riippuvan muuttujan 1. derivaattaan riippumattomaan muuttujaan nähden.
- Käytä kahta jaottoman merkkiä viittaamaan vastaavaan toiseen derivaattaan.

Jaottoman merkkiä käytetään vain deSolve()-funktion derivaatoissa. Muissa tapauksissa käytetään merkintää **d()**.

1. asteen yhtälön yleisratkaisu sisältää mielivaltaisen vakion muotoa c_k , jossa k on kokonaislukuliite väliltä 1-255. 2. asteen yhtälön ratkaisu sisältää kaksi tällaista vakiota.

Käytä **solve()**-funktioita implisiittisessä ratkaisussa, jos haluat yrittää muuntaa sen yhdeksi tai useammaksi ekvivalenttiseksi eksplisiittiseksi ratkaisuksi.

Kun vertaat vastauksia oppikirjan tai käsikirjan ratkaisuihin, huomaa, että erilaiset menetelmät tuovat mielivaltaisia vakioita eri kohtiin laskutoimituksessa, mistä voi olla tuloksena erilaisia yleisratkaisuja.

deSolve(1.astODEandalkuehto, Muutt, riippuvaMuutt) ⇒ tietty ratkaisu

Laskee tietyn ratkaisun, joka täyttää 1.astODE:n ja alkuehdon vaatimukset. Tämä on yleensä helpompaa kuin yleisratkaisun määrittäminen, alkuarvojen korvaaminen, mielivaltaisen vakion ratkaiseminen ja sen jälkeen arvon korvaaminen yleisratkaisuun.

```
deSolve(y''+2*y'+y=x^2,x,y)
      y=(c3*x+c4)*e^-x+x^2-4*x+6
right(Ans)→temp (c3*x+c4)*e^-x+x^2-4*x+6
      0
      d^2(temp)+2*d/dx(temp)+temp-x^2
DelVar temp Done
```

```
deSolve(y'=(cos(y))^2,x,x,y) tan(y)=x^2/c4
solve(Ans,y)
      y=tan^-1(x^2+2*c4/2)+n3*pi
Ans|c4=c-1 and n3=0
      y=tan^-1(x^2+2*(c-1)/2)
```

```
sin(y)=(y*e^x+cos(y))*y'→ode
      sin(y)=(e^x*y+cos(y))*y'
deSolve(ode and y(0)=0,x,y)→soln
      -(2*sin(y)+y^2)/2=(e^x-1)*e^-x*sin(y)
soln|x=0 and y=0 true
ode|y'=impDif(soln,x,y) true
DelVar ode,soln Done
```

alkuehto on yhtälö, joka on muotoa:

riippuvaMuutt (*riippumatonAlkuarvo*) =
riippuvaAlkuarvo

riippumatonAlkuarvo ja *riippuvaAlkuarvo* voivat olla muuttujia, kuten x_0 ja y_0 , joilla ei ole tallennettuja arvoja. Implisiittinen derivointi voi helpottaa implisiittisten ratkaisujen tarkistamista.

deSolve

(*2.astODEandalkuehto1andalkuehto2*,
Muutt, *riippuvaMuutt*) $\Rightarrow$ tietty ratkaisu

Antaa tietyn ratkaisun, joka sopii *2. ast ODE*:hen ja jolla on määritetty riippuvan muuttujan arvo ja sen ensimmäinen derivaatta yhdessä pisteessä.

Käytä *alkuehto1*:lle muotoa:

riippuvaMuutt (*riippumatonAlkuarvo*) =
riippuvaAlkuarvo

Käytä *alkuehto2*:lle muotoa:

riippuvaMuutt (*riippumatonAlkuarvo*) =
1.derivaatanAlkuarvo

deSolve(*2.astODEandreunaehto1and*
reunaehto2, *Muutt*, *riippuvaMuutt*) $\Rightarrow$ tietty ratkaisu

Laskee tietyn ratkaisun, joka sopii *2.astODE*:lle ja jolla on määritetyt arvot kahdessa eri pisteessä.

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

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

$$\text{deSolve}\left(y'' = y^{\frac{-1}{2}} \text{ and } y(0) = 0 \text{ and } y'(0) = 0, t, y\right)$$

$$\frac{\frac{3}{2} \cdot y^{\frac{3}{4}}}{3} = t$$

$$\text{solve}\left(\frac{\frac{3}{2} \cdot y^{\frac{3}{4}}}{3} = t, y\right)$$

$$y = \frac{3 \cdot 3^{\frac{1}{3}} \cdot 2^{\frac{2}{3}} \cdot t^{\frac{4}{3}}}{4} \text{ and } t \geq 0$$

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

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

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

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

det()

Katalogi >

det(neliömatriisi[, Toleranssi])⇒*lauseke*Laskee *neliömatriisin* determinantin.

Valinnaisesti kaikkia matriisielementtejä käsitellään nollana, jos niiden itseisarvo on pienempi kuin *Toleranssi*. Tätä toleranssia käytetään vain, jos matriisissa on liukulukusyötteitä eikä se sisällä symbolisia muuttujia, joille ei ole määritetty arvoa. Muussa tapauksessa *Toleranssia* ei huomioida.

- Jos käytät painikkeita tai **Automaattinen tai likimääräinen** -tilan valintaa Approximate (Likimääräinen), laskut suoritetaan liukulukuaritmetiikalla.
- Jos *Toleranssi* jätetään pois tai sitä ei käytetä, oletusarvoinen toleranssi lasketaan seuraavasti:

$$5E-14 \cdot \max(\text{dim}(\text{neliömatriisi})) \cdot \text{rowNorm}(\text{neliömatriisi})$$

$\det\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	$a \cdot d - b \cdot c$
$\det\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$	-2
$\det\left(\text{identity}(3) - x \cdot \begin{pmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ 6 & -2 & 7 \end{pmatrix}\right)$	$-(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$
$\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \text{mat1}$	$\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$
$\det(\text{mat1})$	0
$\det(\text{mat1}, 1)$	1. E20

diag()

Katalogi >

diag(Lista)⇒*matriisi***diag(rivimatriisi)**⇒*matriisi***diag(sarakematriisi)**⇒*matriisi*

Laskee matriisin, joka sisältää arvot argumenttilistassa tai matriisin sen päälävistäjässä.

diag(neliömatriisi)⇒*rivimatriisi*Laskee rivimatriisin, joka sisältää elementit *neliömatriisin* päälävistäjästä.*neliömatriisi*:n on oltava neliö.

$\text{diag}\left(\begin{bmatrix} 2 & 4 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$
$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$	$\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$
$\text{diag}(\text{Ans})$	$\begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$

dim()

Katalogi >

dim(Lista)⇒*kokonaisluku*Laskee *Listan* mitat.

$\dim(\{0, 1, 2\})$	3
---------------------	---

dim()

Katalogi >

dim(Matriisi)⇒*lista*

Laskee matriisin mitat kahden elementin listana {rivit, sarakkeet}.

dim(Merkkijono)⇒*kokonaisluku*

Laskee merkkijonon *Merkkijono* sisältämien merkkien lukumäärän.

dim	$\begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{pmatrix}$	{ 3,2 }
dim	"Hello"	5
dim	"Hello "&"there"	11

Disp

Katalogi >

Disp lausTaiMerkkijono1 [, lausTaiMerkkijono2] ...

Näyttää *Laskin*-sovelluksen historiatietojen sisältämät argumentit. Argumentit näytetään peräkkäin, ja erotinmerkeinä käytetään ohuita välilyöntejä.

Käyttökelpoisia pääasiassa ohjelmissa ja funktioissa, jotta välilaskutoimitusten näyttäminen voidaan varmistaa.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määrittysten syöttämisestä löytyvät tuotteen ohjekirjan *Laskin*-osiosta.

Define	chars(start,end)=Prgm	
	For i,start,end	
	Disp i," ",char(i)	
	EndFor	
	EndPrgm	
		Done
chars	(240,243)	
		240 ö
		241 ñ
		242 ò
		243 ó
		Done

DispAt

Katalogi >

DispAt int,lauseke1 [,lauseke2 ...] ...

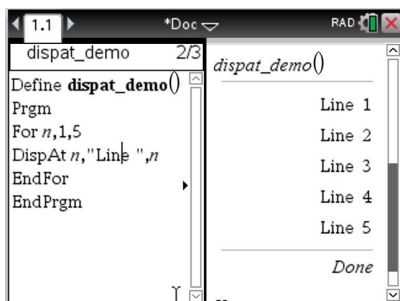
Komennolla DispAt voidaan määritellä rivi, jolla määrätty lauseke tai merkkijono näytetään ruudulla.

Rivinumero voidaan määritellä lausekkeeksi.

Huomaa, että rivin numero ei viittaa koko ruutuun vaan alueeseen, joka seuraa välittömästi komentoa/ohjelmaa.

DispAt

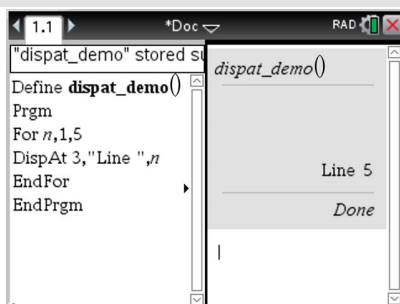
Esimerkki



Tämä komento mahdollistaa ohjauspaneelin kaltaisen tuotoksen ohjelmista, joissa lausekkeen arvo tai anturin lukema päivitetään samalle riville.

Komentoja **DispAt** ja **Disp** voidaan käyttää samassa ohjelmassa.

Huomaa: Suurin sallittu numero on asetettu luvuksi 8, koska se vastaa koko näyttöä täynnä rivejä kannettavassa laitteessa – kunhan riveillä ei ole kaksikulotteisia matemaattisia lausekkeitä. Rivien täsmällinen määrä riippuu näytetyn tiedon sisällöstä.



Havainnollistavia esimerkkejä:

Define z()=	Ulostulo
Prgm	z()
For n,1,50	Iteraatio 1:
DispAt 1,\"N: \",n	Rivi 1: N:1
Disp \"Hello\"	Rivi 2: Hello
EndFor	
EndPrgm	Iteraatio 2:
	Rivi 1: N:2
	Rivi 1: Hello
	Rivi 3: Hello
	Iteraatio 2:
	Rivi 1: N:3
	Rivi 2: Hello
	Rivi 3: Hello
	Rivi 4: Hello
Define z1()=	z1()
Prgm	Rivi 1: N:3
For n,1,50	Rivi 2: Hello
DispAt 1,\"N: \",n	Rivi 3: Hello
EndFor	Rivi 4: Hello
	Rivi 5: Hello
For n,1,50	
Disp \"Hello\"	
EndFor	
EndPrgm	

Virhetilat:

Virheviestit DisplAt-rivinumeron on oltava lukujen 1 ja 8 välillä	Kuvaus Lauseke arvioi rivinumeron välin 1–8 (mukaan lukien) ulkopuolella
Liian vähän argumentteja	Toiminnosta tai komennosta puuttuu yksi tai useampi argumentti
Ei argumentteja	Sama kuin nykyinen "syntaksivirhe" - dialogi
Liian monta argumenttia	Rajoita argumenttia. Sama virhe kuin Disp.
Virheellinen tietotyyppi	Ensimmäisen argumentin on oltava numero.
Mitätön: DispAt mitätön	"Hello World" Datatyyppivirhe on mitätöity (jos soittopyyntö on määritetty)
Muunnosoperaattori: DispAt 2_ft @> _m, "Hello World	CAS: Datatyyppi Virhe annetaan (jos takaisinkutsu on määritetty) Numeerinen Tulos arvioidaan ja jos tulos on kelpuutettu argumentti, DispAt tulostaa merkkijonon tulostuslinjalla.

►DMS

Katalogi >

Laus ►DMS

Astekulmatilassa:

Arvo ►DMS

$\{45.371\}$ ►DMS	45°22'15.6"
$\{\{45.371,60\}\}$ ►DMS	$\{45^{\circ}22'15.6",60^{\circ}\}$

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>DMS.

Tulkitsee argumentin kulmana ja näyttää vastaavan DMS-luvun (DDDDDD°MM'SS.ss"). DMS-muoto (asteet, minuutit, sekunnit) on kuvattu kohdissa °, ', " sivulla sivu 241 .

Huomaa: ►DMS muuntaa radiaanit asteiksi, kun sitä käytetään radiaanitulassa. Jos syötteen perässä on asteen merkki °, muunnosta ei suoriteta. Voit käyttää komentoa ►DMS ainoastaan syöterivin lopussa.

domain()Luettelo > **domain(Laus I, Muut)** ⇒ lauseke

Antaa vastauksena *Laus I* määrittelyjoukon verrattuna *Muut*.

domain() voi käyttää funktioiden arvoalueiden tarkasteluun. Se on rajattu todelliseen ja äärelliseen määrittelyjoukkoon.

Tällä toiminnolla on rajoituksia johtuen tietokonealgebran sieventämis- ja ratkaisualgoritmien puutteista.

Tiettyjä funktioita ei voi käyttää argumentteina **domain()**, -funktioille riippumatta siitä, ilmaantuvatko ne eksplisiitisti tai käyttäjän määrittämien muuttujien ja funktioiden puitteissa. Seuraavassa esimerkissä lauseketta ei voi sieventää, sillä $\int()$ on kielletty funktio.

$$\text{domain}\left(\int_1^x \frac{1}{t} dt, x\right) \rightarrow \text{domain}\left(\int_1^x \frac{1}{t} dt, x\right)$$

$$\text{domain}\left(\frac{1}{x+y}, y\right) \quad -\infty < y < -x \text{ or } -x < y < \infty$$

$$\text{domain}\left(\frac{x+1}{x^2+2 \cdot x}, x\right) \quad x \neq -2 \text{ and } x \neq 0$$

$$\text{domain}\left((\sqrt{x})^2, x\right) \quad 0 \leq x < \infty$$

$$\text{domain}\left(\frac{1}{x+y}, y\right) \quad -\infty < y < -x \text{ or } -x < y < \infty$$

dominantTerm()Katalogi > 

dominantTerm(Laus I, Muutt [, Piste])
⇒ lauseke

dominantTerm(Laus I, Muutt [, Piste]) | Muutt > Piste ⇒ lauseke

dominantTerm(Laus I, Muutt [, Piste]) | Muutt < Piste ⇒ lauseke

$$\text{dominantTerm}(\tan(\sin(x)) - \sin(\tan(x)), x)$$

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

$$\text{dominantTerm}\left(\frac{1 - \cos(x-1)}{(x-1)^3}, x, 1\right) \quad \frac{1}{2 \cdot (x-1)}$$

$$\text{dominantTerm}\left(x^{-2} \cdot \tan\left(\frac{1}{x^3}\right), x\right) \quad \frac{1}{x^3}$$

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

Laskee dominanttitermin *Lausl*:n potenssisarjaesityksestä, kun lauseke on lavennettu *Pisteellä*. Dominanttitermi on se, jonka suuruus kasvaa nopeimmin lähellä arvoa *Muutt* = *Piste*. Lausekkeen (*Muutt* - *Piste*) tuloksena olevalla potenssilla voi olla negatiivinen ja/tai murtolukueksponentti. Tämän potenssin kertoin voi sisältää lausekkeen (*Muutt* - *Piste*) logaritmeja ja muita *Muutt*:n funktioita, joita hallitsevat kaikki lausekkeen (*Muutt* - *Piste*) potenssit, joilla on sama eksponentin etumerkki.

Pisteen oletusarvo on 0. *Piste* voi olla ∞ tai $-\infty$, jolloin dominanttitermi on termi, jolla on suurin *Muutt*:n eksponentti eikä pienin *Muutt*:n eksponentti.

dominantTerm(...) antaa tuloksena "**dominantTerm(...)**", ellei se pysty määrittämään tällaista esitystä, kuten olennaisille erikoispisteille, esim. **sin(1/z)**, kun $z=0$, $e^{-1/z}$, kun $z=0$, tai e^z , kun $z = \infty$ tai $-\infty$.

Jos sarjassa tai yhdellä sen derivaatoista on hyppyepäjatkuvuus kohdassa *Piste*, tulos sisältää todennäköisesti alalausekkeita, jotka ovat muotoa **sign(...)** tai **abs(...)** reaaliselle kehittelmän muuttujalle tai **(-1) floor(...angle(...))** kompleksille kehittelmän muuttujalle, joka on merkkiin "**_**" päättyvä muuttuja. Jos tarkoituksesi on käyttää dominanttitermiä vain *Pisteen* toisella puolella oleville arvoille, siinä tapauksessa liitä funktioon **dominantTerm(...)** sopiva lauseke, "**| Muutt > Piste**", "**| Muutt < Piste**", "**| Muutt ≥ Piste**" tai "**Muutt ≤ Piste**", jotta saat yksinkertaisemman vastauksen.

dominantTerm() jakautuu 1. argumentin listoihin ja matriiseihin.

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

dominantTerm() on hyödyllinen funktio, kun haluat selvittää mahdollisimman yksinkertaisen lausekkeen, joka on asymptoottinen toisen lausekkeen suhteen, esim. *Muutt* $\rightarrow$ *Piste*. **dominantTerm()** on hyödyllinen myös silloin, kun sarjan ensimmäisen ei-nolla-termin astetta ei tiedetä, etkä halua arvailla iteratiivisesti tai interaktiivisesti ohjelmasilmukan avulla.

Huomaa: Katso myös **series()**, sivu 169.

dotP()

dotP(Lista1, Lista2) $\Rightarrow$ *lauseke*

Laskee kahden listan “pistetulon”.

$\text{dotP}(\{a,b,c\},\{d,e,f\})$	$a \cdot d + b \cdot e + c \cdot f$
$\text{dotP}(\{1,2\},\{5,6\})$	17

dotP(Vektori1, Vektori2) $\Rightarrow$ *lauseke*

Laskee kahden vektorin “pistetulon”.

$\text{dotP}([a \ b \ c],[d \ e \ f])$	$a \cdot d + b \cdot e + c \cdot f$
$\text{dotP}([1 \ 2 \ 3],[4 \ 5 \ 6])$	32

Kummankin on oltava rivivektoreita, tai kummankin on oltava sarakevektoreita.

E

e^()

e^(Lausl) $\Rightarrow$ *lauseke*

Laskee **e**:n arvon korotettuna *Lausl*:n potenssiin.

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

Huomaa: Katso myös **e eksponenttimalli**, sivu 2.

Huomaa: Painikkeen  painaminen, jotta näkyviin saadaan $e^()$, on eri asia kuin näppäimistön merkin  painaminen.

Voit syöttää kompleksiluvun $re^{i\theta}$ polaarisisä muodosssa. Käytä tätä muotoa kuitenkin vain radiaanikulmatilassa; aste- tai graadikulmatilassa se aiheuttaa määrittäjäjoukkovirheen (Domain).

e^()**e^x** painike**e^(Listal)⇒lista** $e^{\{1,1,0.5\}}$ $\{e,2.71828,1.64872\}$

Laskee **e:n** arvon korotettuna *Listal:n* jokaisen elementin potenssiin.

e^(neliömatrissi)⇒neliömatrissi

1	5	3	782.209	559.617	456.509
4	2	1	680.546	488.795	396.521
6	-2	1	524.929	371.222	307.879

Laskee *neliömatrissi:n* matriisieksponentin. Tämä ei ole sama kuin laskettaessa **e** korotettuna kunkin elementin mukaiseen potenssiin. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatrissi:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

eff()**Katalogi >** **eff(nimelliskorko,CpY)⇒arvo** $eff(5.75,12)$

5.90398

Talouseläintaloustoiminto, joka muuntaa nimelliskorkokannan *nimelliskorko* efektiiviseksi vuosikoroiksi, kun *CpY* määritetään korkojaksojen lukumääräksi vuodessa.

nimelliskoron on oltava reaalitylily, ja *CpY:n* on oltava reaalitylily > 0.

Huomaa: Katso myös **nom()**, sivu 129.

eigVc()**Katalogi >** **eigVc(neliömatrissi)⇒matrissi**

Suorakulmakompleksimuodossa:

Laskee matriisin, joka sisältää ominaisvektorit reaalitylily tai kompleksiselle *neliömatrissille*, jossa jokainen vastauksen sarake vastaa ominaisarvoa. Huomaa, että ominaisvektori ei ole yksilöllinen; sitä voidaan skaalata millä tahansa vakiokertoimella. Ominaisvektorit ovat normaalitylilytoisia, mikä tarkoittaa, että, jos $V = [x_1, x_2, \dots, x_n]$, tällöin:

-1	2	5	$\rightarrow m1$	-1	2	5
3	-6	9		3	-6	9
2	-5	7		2	-5	7

 $eigVc(m1)$

-0.800906	0.767947	(
0.484029	0.573804+0.052258 <i>i</i>	0.5738
0.352512	0.262687+0.096286 <i>i</i>	0.2626

eigVc()

Katalogi > 

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

neliömatrissiä tasapainotetaan ensin similaarimuunnoksilla, kunnes rivi- ja sarakenormit ovat mahdollisimman lähellä samaa arvoa. Sen jälkeen *neliömatrissiä* sievennetään Hessenbergin ylämatrissimuotoon ja ominaisvektorit lasketaan Schurin tekijöihin jaon menetelmällä.

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

eigVI()

Katalogi > 

eigVI(neliömatrissi)⇒lista

Suorakulmakompleksimuodossa:

Laskee listan reaalisen tai kompleksisen *neliömatrissin* ominaisarvoista.

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

neliömatrissiä tasapainotetaan ensin similaarimuunnoksilla, kunnes rivi- ja sarakenormit ovat mahdollisimman lähellä samaa arvoa. Sen jälkeen *neliömatrissiä* sievennetään Hessenbergin ylämatrissimuotoon ja ominaisarvot lasketaan Hessenbergin ylämatrissista.

$$\text{eigVI}(mI) \\ \{-4.40941, 2.20471 + 0.763006 \cdot i, 2.20471 - 0.763006 \cdot i\}$$

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

Else

Katso If, sivu 92.

ElseIf

Katalogi > 

If BooleanLaus1 Then

Lohko1

ElseIf BooleanLaus2 Then

Lohko2

⋮

ElseIf BooleanLausN Then

LohkoN

EndIf

⋮

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Define $g(x)$ =Func

If $x \leq -5$ Then

Return 5

ElseIf $x > -5$ and $x < 0$ Then

Return $-x$

ElseIf $x \geq 0$ and $x \neq 10$ Then

Return x

ElseIf $x = 10$ Then

Return 3

EndIf

EndFunc

Done

EndFor**Katso For, sivu 76.****EndFunc****Katso Func, sivu 80.****EndIf****Katso If, sivu 92.****EndLoop****Katso Loop, sivu 115.****EndPrgm****Katso Prgm, sivu 144.****EndTry****Katso Try, sivu 201.****EndWhile****Katso While, sivu 211.****euler ()****Katalogi > **

euler(*Expr*, *Var*, *depVar*, {*Var0*, *VarMax*},
depVar0, *VarStep*
[, *eulerStep*]) ⇒ *matriisi*

euler(*SystemOfExpr*, *Var*, *ListOfDepVars*,
{*Var0*, *VarMax*}, *ListOfDepVars0*,
VarStep [, *eulerStep*]) ⇒ *matriisi*

euler(*ListOfExpr*, *Var*, *ListOfDepVars*,

Differentiali yhtälö:

 $y' = 0.001 \cdot y \cdot (100 - y)$ ja $y(0) = 10$

$$\text{euler}(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1)$$

0.	1.	2.	3.	4.
10.	10.9	11.8712	12.9174	14.042

$\{Var0, VarMax\},$
 $ListOfDepVars0, VarStep [, eulerStep])$
 $\Rightarrow$ matriisi

Käyttää Eulerin menetelmää järjestelmän ratkaisuun

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

muuttujalla $\text{depVar}(Var0) = \text{depVar}0$ välillä $[Var0, VarMax]$. Laskee matriisin, jonka ensimmäinen rivi määrittelee Var tulosarvot ja jonka toinen rivi määrittelee ensimmäisen ratkaisukomponentin arvon vastaavilla Var -arvoilla jne.

Expr on oikea puoli, joka määrittelee tavallisen differentiaaliyhtälön (ODE).

SystemOfExpr on oikeiden puolten ryhmä, joka määrittelee ODE-yhtälöiden ryhmän (vastaa riippuvien muuttujien järjestystä kohdassa $ListOfDepVars$).

$ListOfExpr$ on oikeiden puolten luettelo, joka määrittelee ODE-yhtälöiden ryhmän (vastaa riippuvien muuttujien järjestystä kohdassa $ListOfDepVars$).

Var on riippumaton muuttuja.

$ListOfDepVars$ on riippuvien muuttujien luettelo.

$\{Var0, VarMax\}$ on kahden elementin lista, joka määrittää funktion integroinnin muuttujasta $Var0$ muuttujaan $VarMax$.

$ListOfDepVars0$ on riippuvien muuttujien aluehtojen luettelo.

$VarStep$ nollasta eroava numero niin, että $\text{sign}(VarStep) = \text{sign}(VarMax - Var0)$ ja ratkaisut lasketaan $Var0 + i \cdot VarStep$ kaikille $i = 0, 1, 2, \dots$ niin, että $Var0 + i \cdot VarStep$ on alueella $[Var0, VarMax]$ (muuttujalla $VarMax$ ei ehkä ole ratkaisuarvoa).

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

Vertaile yllä olevaa tulosta CAS:n tarkkaan tulokseen, joka on saatu käyttämällä $\text{deSolve}()$ - ja $\text{seqGEN}()$ -funktioita:

$$\text{deSolve}\{y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y\}$$

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

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

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

Yhtälöryhmä:

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

kun $y1(0) = 2$ ja $y2(0) = 5$

$$\text{euler}\left\{\begin{cases} y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right\}$$

0.	1.	2.	3.	4.	5.
2.	1.	1.	3.	27.	243.
5.	10.	30.	90.	90.	-2070.

eulerStep on positiivinen kokonaisluku (oletus 1), joka määrittelee Eulerin vaiheiden määrän tulosarvojen välillä. Eulerin menetelmän käyttämä varsinainen vaihemäärä on *VarStep/eulerStep*.

eval()**Laitevalikko**

eval(*Expr*) $\Rightarrow$ *string*

eval() on validi vain TI-Innovator™ Hub ohjelmointikomentojen komentoargumenteissa **Get**, **GetStr**, ja **Send**. Ohjelmisto käsittelee lausekkeen *Expr* ja korvaa **eval()**-ilmauksen lopputuloksella merkkijoukkona

Argumentin *Expr* on sievennyttävä reaaliarvo.

Aseta RGB-ledin sininen väri puolelle intensiteetille.

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

Palauta sininen väri OFF-tilaan.

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

eval()-argumentin on sievennyttävä reaaliarvo.

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

Ohjelmoi punainen väri voimistumaan

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

Suorita ohjelma.

fadein()	Done
-----------------	------

Vaikka **eval()** ei näytä tulostaan, voi tuloksena saatavaa laitekomentojonoa katsoa komennon suorittamisen jälkeen tarkastamalla jonkin seuraavista erikoismuuttujista.

iostr.SendAns
iostr.GetAns
iostr.GetStrAns

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

Huomio: Katso myös **Get** (sivu 82), **GetStr** (sivu 89), ja **Send** (sivu 167).

exact()

Katalogi >

exact(*Laus* [, *Toleranssi*]) $\Rightarrow$ *lauseke*

exact(*Listal* [, *Toleranssi*]) $\Rightarrow$ *lista*

exact(*Matriisi* [, *Toleranssi*]) $\Rightarrow$ *matriisi*

Laskee täsmällisen tilan aritmetiikalla argumentin vastaavan rationaaliluvun, mikäli mahdollista.

Toleranssi määrittää muunnoksen toleranssin; oletusarvo on 0 (nolla).

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

Exit

Katalogi >

Exit

Poistuu nykyisestä **For**-, **While**- tai **Loop**-lohkosta.

Exit-komento ei ole sallittu näiden kolmen silmukkarakenteen (**For**, **While** tai **Loop**) ulkopuolella.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Funktion listaus:

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

►exp

Katalogi >

Laus ►exp

Näyttää *Laus*:n *e*:n luonnollisen eksponentin arvolla. Tämä on näytön muunnosoperaattori. Sitä voidaan käyttää vain syöterivin lopussa.

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

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>exp.

exp()

 painike

exp(LausI)⇒lauseke

Laskee **e**:n arvon korotettuna *LausI*:n potenssiin.

Huomaa: Katso myös **e** eksponenttimalli, sivu 2.

Voit syöttää kompleksiluvun $re^{i\theta}$ polaarisisä muodossa. Käytä tätä muotoa kuitenkin vain radiaanikulmatilassa; aste- tai graadikulmatilassa se aiheuttaa määrittämissuorituksen (Domain).

exp(ListaI)⇒lista

Laskee **e**:n arvon korotettuna *Listal*:n jokaisen elementin potenssiin.

exp(neliömatriisiI)⇒neliömatriisi

Laskee *neliömatriisiI*:n matriisiekspontin. Tämä ei ole sama kuin laskettaessa **e** korotettuna kunkin elementin mukaiseen potenssiin. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisiI:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

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

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

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

exp►lista()

Katalogi > 

exp►list(Laus,Muutt)⇒lista

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

Tutkii, onko lausekkeessa *Laus* yhtälöitä, jotka on erotettu sanalla "or", ja laskee listan, joka sisältää yhtälöiden oikeat puolet, jotka ovat muotoa *Muutt*=*Laus*. Tällä tavoin voit saada helpolla tavalla joitakin ratkaisuarvoja, jotka on upotettu funktioiden *solve()*, *cSolve()*, *fMin()* ja *fMax()* vastauksiin.

Huomaa: *exp►list()* ei ole välttämätön funktioiden *zeros* ja *cZeros()* kanssa, koska ne laskevat suoraan ratkaisulistan.

Voit syöttää tämän funktion näppäimistöltä kirjoittamalla *exp@>list (...)*.

expand()

expand(Laus I [, Muutt]) ⇒ lauseke
expand(Lista I [, Muutt]) ⇒ lista
expand(Matriisi I [, Muutt]) ⇒ matriisi

expand(Laus I) laskee *Laus I*:n arvon lavennettuna kaikkien muuttujiensa suhteen. Lavennus on polynomilavennus polynomeille ja osamurtolukulavennus rationaalilukulausekkeille.

Funktion *expand()* tehtävä on muuntaa *Laus I* yksinkertaisten termien summaksi ja/tai erotukseksi. Funktion *factor()* tehtävä sen sijaan on muuntaa *Laus I* yksinkertaisten tekijöiden tuloksi ja/tai osamääräksi.

expand(Laus I, Muutt) laskee *Laus I*:n arvon lavennettuna muuttujan *Muutt* suhteen. Muuttujan *Muutt* samanlaiset potenssit kerätään. Termit ja niiden tekijät lajitellaan siten, että *Muutt* on päämuuttuja. Kerättyjen kertoimien satunnaista tekijöihin jakamista tai laventumista voi esiintyä jonkin verran. Verrattuna siihen, että muuttuja *Muutt* jätettäisiin pois, tämä toiminto säästää usein aikaa, muistia ja näyttötilaa, ja samalla lausekkeesta tulee ymmärrettävämpi.

$$\begin{aligned} &\text{expand}\left((x+y+1)^2\right) \\ &\quad x^2+2\cdot x\cdot y+2\cdot x\cdot y^2+2\cdot y+1 \\ &\text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}\right) \\ &\quad \frac{1}{x-1}-\frac{1}{x}+\frac{1}{y-1}-\frac{1}{y} \end{aligned}$$

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

expand()Katalogi > 

Vaikka muuttujia olisi vain yksi, muuttujan *Muutt* käytön ansiosta nimittäjää voidaan mahdollisesti jakaa täydellisemmin tekijöihinsä lavennettaessa murtolukua osittain.

Vinkki: Rationaalilausekkeissa **propFrac()** on nopeampi, mutta vähemmän äärimmäinen vaihtoehto kuin **expand()**.

Huomaa: Katso myös **comDenom()**, jossa käsitellään lavennetulla nimittäjällä lavennettua osoittajaa.

expand(LausI, [Muutt]) jakaa myös logaritmit ja murtopotenssit muuttujasta *Muutt* riippumatta. Jos logaritmeja ja murtopotensseja halutaan jakaa enemmän, tarvitaan mahdollisesta epäyhtälöehtoja, jotta voidaan varmistaa, että jotkin tekijät ovat ei-negatiivisia.

expand(LausI, [Muutt]) jakaa myös itseisarvot, **sign()**, ja eksponentit muuttujasta *Muutt* riippumatta.

Huomaa: Katso myös **tExpand()**, jossa käsitellään trigonometrista kulma-summa- ja monikulmalavennusta.

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

$$\begin{array}{l} \frac{\ln(2 \cdot x \cdot y) + \sqrt{2 \cdot x \cdot y}}{\text{expand}(Ans)} \quad \frac{\ln(2 \cdot x \cdot y) + \sqrt{2 \cdot x \cdot y}}{\ln(x \cdot y) + \sqrt{2 \cdot x \cdot y} + \ln(2)} \\ \text{expand}(Ans)_{y \geq 0} \quad \ln(x) + \sqrt{2 \cdot x \cdot y} + \ln(y) + \ln(2) \\ \text{sign}(x \cdot y) + |x \cdot y| + e^{2 \cdot x + y} \quad e^{2 \cdot x + y + \text{sign}(x \cdot y) + |x \cdot y|} \\ \text{expand}(Ans) \quad \text{sign}(x) \cdot \text{sign}(y) + |x| \cdot |y| + (e^x)^2 \cdot e^y \end{array}$$

expr()Katalogi > 

expr(Merkkijono) ⇒ lauseke

Määrittää *Merkkijonon* sisältämän merkkijonon lausekkeena ja suorittaa toimenpiteen välittömästi.

$$\begin{array}{l} \text{expr}("1+2+x^2+x") \quad x^2+x+3 \\ \text{expr}("expand((1+x)^2)") \quad x^2+2 \cdot x+1 \\ \text{"Define cube}(x)=x^3" \rightarrow \text{funcstr} \quad \text{"Define cube}(x)=x^3" \\ \text{expr}(\text{funcstr}) \quad \text{Done} \\ \text{cube}(2) \quad 8 \end{array}$$

ExpRegKatalogi > 

ExpReg *X*, *Y* [, [*Frekv*][, *Luokka*, *Sisällyttä*]]

Laskee eksponentiaalisen regressioyhtälön $y = a \cdot (b)^x$ listoista X ja Y frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja ≥ 0 .

Luokka on luokkakoodien lista vastaavalle X - ja Y -datalle.

Sisällytä on yhden tai useamman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementtisivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a \cdot (b)^x$
stat.a, stat.b	Regressiokertoimet
stat.r ²	Muunnettujen tietojen lineaarimäärittelyn kerroin
stat.r	Muunnettujen tietojen korrelaatiokerroin (x , $\ln(y)$)
stat.Resid	Eksponentiaalimalliin liittyvät jäännökset
stat.ResidTrans	Muunnettujen tietojen lineaariseen sovitukseen liittyvät jäännökset
stat.XReg	Muokatus X Lista:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat</i> rajoitusten mukaisesti
stat.YReg	Muokatus Y Lista:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat</i> rajoitusten mukaisesti
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

factor()

Katalogi > **factor(Laus I[, Muutt])** ⇒ lauseke**factor(Lista I[, Muutt])** ⇒ lista**factor(Matriisi I[, Muutt])** ⇒ matriisi

factor(Laus I) jakaa *Laus I*:n kaikki muuttujat tekijöihin yhteisen nimittäjän suhteen.

Laus I :ä jaetaan tekijöihin mahdollisimman paljon lineaaristen rationaalilukutekijöiden suuntaan ilman, että uusia ei-reaalisia alalausekkeita syntyy. Tämä vaihtoehto on sopiva, jos haluat jakaa lausekkeen tekijöihin useamman kuin yhden muuttujan suhteen.

factor(Laus I, Muutt) laskee *Laus I*:n jaettuna tekijöihin muuttujan *Muutt* suhteen.

Laus I :ä jaetaan mahdollisimman paljon kohti reaalisia tekijöitä, jotka ovat lineaarisia muuttujassa *Muutt*, vaikka tästä syntyisi irrationaalisia vakioita tai alalausekkeita, jotka ovat irrationaalisia muissa muuttujissa.

Tekijät ja niiden termit lajitellaan siten, että *Muutt* on päämuuttuja. Muuttujan *Muutt* samanlaiset potenssit kerätään jokaisessa tekijässä. Muuttujan *Muutt* tulee olla mukana, jos vain kyseistä muuttujaa halutaan jakaa tekijöihin ja jos irrationaalilausekkeet ovat hyväksyttäviä kaikissa muissa muuttujissa, jotta muuttujaa *Muutt* voitaisiin jakaa enemmän tekijöihin. Toimenpiteessä voi esiintyä jonkin verran satunnaista muiden muuttujien tekijöihin jakamista.

$$\text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a) \\ a \cdot (a-1) \cdot (a+1) \cdot (x-1) \cdot (x+1)$$

$$\text{factor}(x^2+1) \quad x^2+1$$

$$\text{factor}(x^2-4) \quad (x-2) \cdot (x+2)$$

$$\text{factor}(x^2-3) \quad x^2-3$$

$$\text{factor}(x^2-a) \quad x^2-a$$

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

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

$$\text{factor}(x^2-a, x) \quad (x+\sqrt{a}) \cdot (x-\sqrt{a})$$

Auto or Approximate (Automaattinen tai likimääräinen) -tilan Auto (Automaattinen) -asetuksessa muuttujan *Muutt* mukanaolo sallii likiarvoistamisen liukulukukertoimilla, kun irrationaalisia kertoimia ei voida ilmaista täsmällisen tiiviisti sisäänrakennetuilla termeillä. Vaikka muuttujia olisi vain yksi, muuttujan *Muutt* mukanaolo voi tuottaa täydellisemmän tekijöihin jakamisen.

Huomaa: Katso myös **comDenom()**, jossa on kuvattu nopea tapa suorittaa osittainen tekijöihin jako, kun **factor()** ei ole tarpeeksi nopea tai käyttää liikaa muistia.

Huomaa: Katso myös **cFactor()**, jossa on kuvattu täydellinen tekijöihin jako kompleksikertoimiksi etsittäessä lineaarisia tekijöitä.

factor(rationaaliluku) laskee rationaaliluvun, joka on jaettu jaottomiin tekijöihin. Sekalukujen kohdalla laskenta-aika pitenee eksponentiaalisesti toiseksi suurimman tekijän sisältämien numeroiden määrän suhteen. Esimerkiksi 30-numeroisen kokonaisluvun tekijöihin jakaminen voi kestää pitempään kuin vuorokauden ja 100-numeroisen luvun pitempään kuin vuosisadan.

Pysäytä laskenta käsin,

- **Kämmenlaite:** Pidä  -painiketta painettuna ja paina toistuvasti  -painiketta.
- **Windows®:** Pidä **F12**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **Macintosh®:** Pidä **F5**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **iPad®:** Sovellus näyttää kehotuksen. Voit jatkaa odottamista tai peruuttaa.

$$\text{factor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3)$$

$$x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3$$

$$\text{factor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3, x)$$

$$(x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x^2$$

$\text{factor}(152417172689)$	$123457 \cdot 1234577$
$\text{isPrime}(152417172689)$	false

factor()**Katalogi** > 

Jos haluat pelkästään määrittää, onko jokin luku jaoton, käytä sen sijaan komentoa **isPrime()**. Se on paljon nopeampi, erityisesti jos *rationaaliluku* ei ole jaoton, ja jos toiseksi suurimmassa tekijässä on enemmän kuin viisi numeroa.

Fcdf()**Katalogi** > 

Fcdf(alaraja,yläraja,dfOsoitt,dfNimitt)
⇒*luku*, jos *alaraja* ja *yläraja* ovat lukuja,
lista, jos *alaraja* ja *yläraja* ovat listoja

FCdf(alaraja,yläraja,dfOsoitt,dfNimitt)
⇒*luku*, jos *alaraja* ja *yläraja* ovat lukuja,
lista, jos *alaraja* ja *yläraja* ovat listoja

Laskee F-jakauman todennäköisyyden *alarajan* ja *ylärajan* välillä määritellylle *dfOsoittajalle* (vapausaste) ja *dfNimittäjälle*.

Aseta $P(X \leq \textit{yläraja})$:lle *alaraja* = 0.

Fill**Katalogi** > 

Fill Laus, matriisiMuutt⇒matriisi

Korvaa muuttujan *matriisiMuutt* jokaisen elementin lausekkeella *Laus*.

*matriisiMuutt*ujan on oltava valmiiksi olemassa.

Fill Laus, listaMuutt⇒lista

Korvaa muuttujan *listaMuutt* jokaisen elementin lausekkeella *Laus*.

*listaMuutt*ujan on oltava valmiiksi olemassa.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	→ <i>amatrix</i>	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Fill 1.01, <i>amatrix</i>	Done	
<i>amatrix</i>	$\begin{bmatrix} 1.01 & 1.01 \\ 1.01 & 1.01 \end{bmatrix}$	
$\{ 1,2,3,4,5 \}$	→ <i>alist</i>	$\{ 1,2,3,4,5 \}$
Fill 1.01, <i>alist</i>	Done	
<i>alist</i>	$\{ 1.01,1.01,1.01,1.01,1.01 \}$	

FiveNumSummary**Katalogi** > 

FiveNumSummary *X*[,*[Frekv]*
[,*Luokka,Sisällytä*]]

Antaa lyhennetyn version 1 muuttujan tilastoista listalle X . Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

X edustaa datan sisältävää listaa.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan X :n arvon esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on numeeristen luokkakoodien lista vastaaville X :n arvoille.

Sisällytyä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Tyhjä elementti jossakin listassa X , *Frekv* tai *Luokka* saa aikaan, että kaikkien listojen vastaava elementti on tyhjä. Lisätietoja tyhjästä elementeistä, katso sivu 249.

Tulosmuuttuja	Kuvaus
stat.MinX	$x:n$ arvojen minimi
stat.Q ₁ X	$x:n$ ensimmäinen neljännes
stat.MedianX	$x:n$ mediaani
stat.Q ₃ X	$x:n$ kolmas neljännes
stat.MaxX	$x:n$ arvojen maksimi

floor()

floor(*Laus I*) $\Rightarrow$ kokonaisluku

floor(-2.14)

-3.

Laskee suurimman kokonaisluvun, joka on $\leq$ argumentti. Tämä funktio on identtinen funktion **int()** kanssa.

Argumentti voi olla reaali- tai kompleksiluku.

floor()		Katalogi > 
floor (<i>Listal</i>)⇒ <i>lista</i>	$\text{floor}\left(\left\{\frac{3}{2}, 0, -5.3\right\}\right)$	$\{1, 0, -6\}$
floor (<i>Matriisi l</i>)⇒ <i>matriisi</i>	$\text{floor}\begin{pmatrix} 1.2 & 3.4 \\ 2.5 & 4.8 \end{pmatrix}$	$\begin{bmatrix} 1. & 3. \\ 2. & 4. \end{bmatrix}$
Määrittää listan tai matriisin jokaisen elementin alarajasta.		
Huomaa: Katso myös ceiling() ja int() .		

fMax()		Katalogi > 
fMax (<i>Laus</i> , <i>Muutt</i>)⇒ <i>Boolean lauseke</i>	$\text{fMax}\left(1-(x-a)^2-(x-b)^2, x\right)$	$x=\frac{a+b}{2}$
fMax (<i>Laus</i> , <i>Muutt</i> , <i>alaraja</i>)	$\text{fMax}\left(0.5 \cdot x^3-x-2, x\right)$	$x=\infty$
fMax (<i>Laus</i> , <i>Muutt</i> , <i>alaraja</i> , <i>yläraja</i>)		
fMax (<i>Laus</i> , <i>Muutt</i>) <i>alaraja</i> ≤ <i>Muutt</i> ≤ <i>yläraja</i>		
Laskee Boolean lausekkeen, joka määrittää mahdollisia arvoja muuttujalle <i>Muutt</i> , joilla saadaan suurin lausekkeen <i>Laus</i> arvo tai jotka määrittävät sen pienimmän ylärajan.		
Voit käyttää sijoitusoperaattoria (" ") rajoittaaksesi ratkaisuväliä ja/tai määrittääksesi muita ehtoja.		
	$\text{fMax}\left(0.5 \cdot x^3-x-2, x\right)_{ x \leq 1}$	$x=-0.816497$

Auto or Approximate (Automaattinen tai likimääräinen) -tilan Approximate (Likimääräinen) -asetuksessa **fMax()** etsii iteratiivisesti yhtä likimääräistä paikallista maksimiarvoa. Tämä on usein nopeampi menetelmä, erityisesti jos rajoitat haun operaattorilla "|" suhteellisen pienelle välille, joka sisältää täsmälleen yhden paikallisen maksimin.

Huomaa: Katso myös **fMin()** ja **max()**.

fMin()		Katalogi > 
fMin (<i>Laus</i> , <i>Muutt</i>)⇒ <i>Boolean lauseke</i>	$\text{fMin}\left(1-(x-a)^2-(x-b)^2, x\right)$	$x=-\infty \text{ or } x=\infty$
fMin (<i>Laus</i> , <i>Muutt</i> , <i>alaraja</i>)	$\text{fMin}\left(0.5 \cdot x^3-x-2, x\right)_{ x \geq 1}$	$x=1.$
fMin (<i>Laus</i> , <i>Muutt</i> , <i>alaraja</i> , <i>yläraja</i>)		
fMin (<i>Laus</i> , <i>Muutt</i>) <i>alaraja</i> ≤ <i>Muutt</i> ≤ <i>yläraja</i>		

Laskee Boolean lausekkeen, joka määrittää mahdollisia arvoja muuttujalle *Muutt*, joilla saadaan pienin lausekkeen *Laus* arvo tai jotka määrittävät sen suurimman alarajan.

Voit käyttää sijoitusoperaattoria (“|”) rajoittaaksesi ratkaisuväliä ja/tai määrittääksesi muita ehtoja.

Auto or Approximate (Automaattinen tai likimääräinen) -tilan Approximate (Likimääräinen) -asetuksessa **fMin()** etsii iteratiivisesti yhtä likimääräistä paikallista minimiarvoa. Tämä on usein nopeampi menetelmä, erityisesti jos rajoitat haun operaattorilla “|” suhteellisen pienelle välille, joka sisältää täsmälleen yhden paikallisen minimin.

Huomaa: Katso myös **fMax()** ja **min()**.

For

For *Muutt, Matala, Korkea* [, *Askel*]
Lohko

EndFor

Suorittaa *Lohkon* sisältämät lausekkeet iteratiivisesti jokaiselle muuttujan *Muutt* arvolle, *Matalasta Korkeaan* kohdassa *Askel* määritetyn portain.

Muutt ei saa olla järjestelmän muuttuja.

Askel voi olla positiivinen tai negatiivinen. Oletusarvo on 1.

Lohko voi olla joko yksi lauseke tai sarja lausekkeita, jotka on erotettu toisistaan kaksoispisteellä (:).

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Define $g()$ =Func	Done
Local <i>tempsum, step, i</i>	
$0 \rightarrow tempsum$	
$1 \rightarrow step$	
For $i, 1, 100, step$	
$tempsum + i \rightarrow tempsum$	
EndFor	
EndFunc	
$g()$	5050

format()Katalogi > 

format(Laus[, muotoMerkkijono])
 $\Rightarrow$ merkkijono

Määrittää lausekkeen *Laus* merkkijonona muotoilumallin perusteella.

Laus on voitava sieventää luvuksi.

muotoMerkkijono on merkkijono, ja sen tulee olla muodossa: "F[n]", "S[n]", "E[n]", "G[n][c]", jossa [] ilmaisevat valinnaisia osia.

F[n]: Kiinteä muoto. n on desimaalipisteen jälkeen näytettävien numeroiden lukumäärä.

S[n]: Kymmenpotenssimuoto. n on desimaalipisteen jälkeen näytettävien numeroiden lukumäärä.

E[n]: Tekninen esitystapa. n on ensimmäisen merkitsevän numeron jälkeen näytettävien numeroiden lukumäärä. Eksponentti säätyy kolmella kerroolliseksi, ja desimaalipiste siirtyy 0, 1 tai 2 numeroa oikealle.

G[n][c]: Muuten sama kuin kiinteä muoto, mutta erottaa myös juuren vasemmalla puolella olevat numerot kolmen ryhmiin. c määrittää ryhmän erotusmerkin, ja sen oletusarvo on pilkku. Jos c on piste, juuri näytetään pilkkuna.

[Rc]: Mihin tahansa edellä mainituista määrittäjistä voidaan liittää Rc-juurilippu, jossa c on yksi merkki, joka määrittää korvauksen kohteen juuripisteestä.

<code>format(1.234567,"f3")</code>	"1.235"
<code>format(1.234567,"s2")</code>	"1.23e0"
<code>format(1.234567,"e3")</code>	"1.235e0"
<code>format(1.234567,"g3")</code>	"1.235"
<code>format(1234.567,"g3")</code>	"1,234.567"
<code>format(1.234567,"g3,r:")</code>	"1:235"

fPart()Katalogi > 

fPart(LausI) $\Rightarrow$ lauseke

fPart(ListaI) $\Rightarrow$ lista

fPart(MatriisiI) $\Rightarrow$ matriisi

<code>fPart(-1.234)</code>	-0.234
<code>fPart({1,-2.3,7.003})</code>	{0,-0.3,0.003}

Laskee argumentin murtolukuosan.

Kun kyseessä on lista tai matriisi, laskee elementtien murtolukuosat.

Argumentti voi olla reaali- tai kompleksiluku.

$\text{FPdf}(XArvo, dfOsoitt, dfNimitt) \Rightarrow luku$, jos $XArvo$ on luku, $lista$, jos $XArvo$ on lista

Laskee F-jakauman todennäköisyyden $XArvo$ n kohdalle määritetyille $dfOsoittajalle$ (vapausasteet) ja $dfNimittäjälle$.

freqTable►list

$(Listal, frekvKokonaislukuLista) \Rightarrow lista$

Laskee listan, joka sisältää $Listal$:n elementit lavennettuina $frekvKokonaislukuListan$ määrittämien frekvenssien mukaisesti. Tätä funktiota voidaan käyttää laadittaessa frekvenssitaulukkoa Data & Tilastot -sovelluksessa.

$Listal$ voi olla mikä tahansa kelvollinen lista.

$frekvKokonaislukuListan$ on oltava samankokoinen kuin $Listal$ ja sen tulee sisältää ainoastaan ei-negatiivisia kokonaislukuelementtejä. Jokainen elementti määrittää kuinka monta kertaa $Listal$ -elementti toistetaan tuloslistassa. Nolla-arvo sulkee pois vastaavan $Listal$ -elementin.

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla **freqTable@>list(...)**.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjästä elementeistä, katso sivu 249.

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

frequency(*Listal*,*lokerotListal*) $\Rightarrow$ *lista*

Luo listan, joka sisältää *Listal*:n elementtien lukumäärät. Lukumäärät perustuvat alueisiin (lokeroihin), jotka määritetään kohtaan *lokerotListal*.

Jos *lokerotListal* on {b(1), b(2), ..., b(n)}, määritetyt alueet ovat { $? \leq b(1)$, $b(1) < ? \leq b(2)$, ..., $b(n-1) < ? \leq b(n)$, $b(n) > ?$ }. Tuloksena oleva lista on yhden elementin pitempi kuin *lokerotListal*.

Jokainen vastauksen elementti vastaa niiden *Listal*:n elementtien lukumäärää, jotka ovat kyseisen lokeron alueella. Funktion **countlf()** termeillä ilmaistuna vastaus on { countlf(list, $? \leq b(1)$), countlf(list, $b(1) < ? \leq b(2)$), ..., countlf(list, $b(n-1) < ? \leq b(n)$), countlf(list, $b(n) > ?$)}.

Niitä *Listal*:n elementtejä, joita ei voi "lokeroida", ei huomioida. Tyhjiä elementtejä ei myöskään huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

Listat & Taulukot -sovelluksessa voit käyttää solualueita kummankin argumentin tilalla.

Huomaa: Katso myös **countlf()**, sivu 37.

<i>datalist</i> := {1,2,e,3, π ,4,5,6,"hello",7}
{1,2,2.71828,3,3.14159,4,5,6,"hello",7}
frequency(<i>datalist</i> , {2.5,4.5})
{2,4,3}

Vastauksen selitys:

2 *Datalistan* elementtiä on ≤ 2.5

4 *Datalistan* elementtiä on > 2.5 ja ≤ 4.5

3 *Datalistan* elementtiä on > 4.5

Elementti "hei" on merkkijono, jota ei voi sijoittaa mihinkään määritetyistä lokeroista.

FTest_2Samp

FTest_2Samp *Listal*,*Listal2* [,*Frekv1* [,*Frekv2* [,*Hypot*]]]

FTest_2Samp *Listal*,*Listal2* [,*Frekv1* [,*Frekv2* [,*Hypot*]]]

(*Datalistan* syöte)

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

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

(Yhteenvetotilaston syöte)

Suorittaa kahden otoksen F -testin. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kun $H_0: \sigma_1 > \sigma_2$, aseta *Hypot*>0
 Kun $H_0^a: \sigma_1 \neq \sigma_2$ (oletus), aseta *Hypot* =0
 Kun $H_0^a: \sigma_1 < \sigma_2$, aseta *Hypot*<0

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.F	Laskettu F -tilasto datasekvensseille
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat.dfNumer	osoittajan vapausasteet = n1-1
stat.dfDenom	nimittäjän vapausasteet = n2-1
stat.sx1, stat.sx2	Otoksen keskihajonnat <i>Lista 1</i> :n ja <i>Lista 2</i> :n sisältämille datasekvensseille
stat.x1_bar stat.x2_bar	Otoksen keskiarvot <i>Lista 1</i> :n ja <i>Lista 2</i> :n sisältämille datasekvensseille
stat.n1, stat.n2	Otosten koko

Func

Func

Lohko

EndFunc

Malli käyttäjän määrittämän funktion luomista varten.

Lohko voi olla yksi lauseke tai sarja lausekkeita, jotka on erotettu toisistaan kaksoispisteellä (:), tai sarja eri riveillä olevia lausekkeita. Funktio voi käyttää **Return**-ohjetta tietyn vastauksen laskemiseen.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

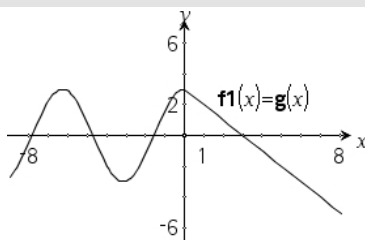
Määritä paloittain määritelty funktio:

```

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

```

Funktion g(x) kuvaajan piirtämisen tulos



G

gcd()

Katalogi > **gcd**(*Arvo1*, *Arvo2*) $\Rightarrow$ *lauseke* $\text{gcd}(18,33)$ 3

Laskee kahden argumentin suurimman yhteisen jakajan. Kahden murtoluvun **gcd** on niiden osoittajien **gcd** jaettuna nimittäjien **lcm**:llä.

Auto or Approximate (Automaattinen tai likimääräinen) -tilassa murtoluvun liukulukujen **gcd** on 1.0.

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

Laskee *Lista1*:n ja *Lista2*:n toisiaan vastaavien elementtien suurimmat yhteiset jakajat.

gcd(*Matriisi1*, *Matriisi2*) $\Rightarrow$ *matriisi* $\text{gcd}\left(\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}, \begin{bmatrix} 4 & 8 \\ 12 & 16 \end{bmatrix}\right)$ $\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$

Laskee *Matriisi1* :n ja *Matriisi2*:n toisiaan vastaavien elementtien suurimmat yhteiset jakajat.

geomCdf()

Katalogi > 

geomCdf(*p*,*alaraja*,*yläraja*) $\Rightarrow$ *luku*, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

geomCdf(*p*,*yläraja*)kun $P(1 \leq X \leq \text{yläraja})$
 $\Rightarrow$ *luku*, jos *yläraja* on luku, *lista*, jos *yläraja* on lista

Laskee kumulatiivisen geometrisen todennäköisyyden *alarajalta ylärajalle* määritetyllä onnistumistodennäköisyydellä *p*.

Kun $P(X \leq \textit{yläraja})$, aseta *alaraja* = 1.

geomPdf()

geomPdf(*p*, *XArvo*) ⇒ *luku*, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

Laskee diskreetin jakauman todennäköisyyden *XArvo*:n, eli ensimmäisen onnistuneen kokeen järjestysnumeron kohdalla, määritetyllä onnistumistodennäköisyydellä *p*.

Get

Laitevalikko

Get [*kehotemerkkijono*,] *var*[, *statusVar*]

Get [*kehotemerkkijono*,] *func*(*arg1*, ...*argn*)[, *statusVar*]

Ohjelmointikomento: Noutaa arvon liitetystä TI-Innovator™ Hub ja sijoittaa arvon muuttujaan *var*.

Arvo täytyy kysyä:

- Etukäteen **Send "READ ..."** - komennolla.
— tai —
- Upottamalla **"READ ..."** kysy valinnaisena *kehotemerkkijono* argumenttina. Tämä menetelmä antaa sinun käyttää yksittäistä komentoa kysyäksesi arvoa ja hakeaksesi sen.

Tapahtuu implisiittinen yksinkertaistus. Esimerkiksi vastaanotettu merkkijono "123" tulkitaan numeeriseksi arvoksi. Säilyttääksesi merkkijonon käytä toimintoa **GetStr** toiminnon **Get** sijaan.

Esimerkki: Kysy laitteen sisäänrakennetun valaistusanturin tämänhetkinen arvo. Käytä komentoa **Get** noutaaksesi arvon ja sijoittaaksesi se muuttujaan *lightval*.

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

Upota READ-kysely komentoon **Get**.

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

Jos sisällytät valinnaisen argumentin *statusVar*, sille määrätään arvo toimenpiteen onnistumisen perusteella. Arvo nolla merkitsee, ettei tietoa ole vastaanotettu.

Järjestyksessä toisessa syntaksissa *func()*-argumentti sallii ohjelman tallentaa vastaanotetun merkkijonon funktiomääritelmänä. Tämä syntaksi toimii ikään kuin ohjelma suorittaisi komennon:

Määrittele *func(arg1, ...argn) = vastaanotettu merkkijono*

Sen jälkeen ohjelma voi käyttää määriteltyä funktiota *func()*.

Huomio: Komentoa **Get** voi käyttää käyttäjän määrittelemän ohjelman sisällä, mutta ei funktion sisällä.

Huomio: Katso myös **GetStr**, sivu 89 ja **Send** sivu 167.

getDenom()

Katalogi > 

getDenom(Laus l) ⇒ lauseke

Muuttaa argumentin lausekkeeksi, jolla on sievennetty yhteinen nimittäjä, ja laskee sen jälkeen lausekkeen nimittäjän.

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

getKey()

Katalogi > 

getKey([0|1]) ⇒ returnString

Kuvaus: **getKey()** – mahdollistaa TI-Basic -ohjelmalle näppäimistön – kannettava laite, pöytätietokone ja emulaattori pöytätietokoneella.

getKey()

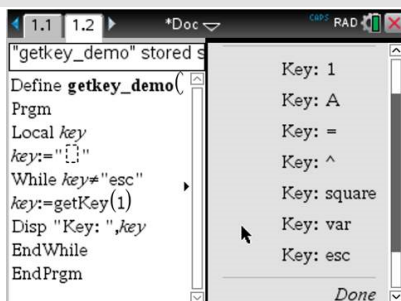
Esimerkki:

Esimerkki:

- alas painettu näppäin := **getKey()**
palauttaa näppäimen tai tyhjän

merkkijonon, jos mitään näppäintä ei ole painettu. Tämä komento palaa välittömästi.

- alas painettu näppäin := **getKey(1)** odottaa, kunnes jotakin näppäintä painetaan. Tämä komento pysäyttää ohjelman suorituksen, kunnes jotakin näppäintä on painettu.



Näppäimen pitäminen painettuna:

Kannettava laite/Emulaattorinäppäin	Pöytätietokone	Palauta arvo
Esc	Esc	"esc"
Kosketuslevy – Yläpainallus	N/A	"ylös"
Päällä	N/A	"päävalikko"
Scratchapps	N/A	muistilehtiö
Kosketuslevy – Vasen painallus	N/A	"vasen"
Kosketuslevy – Keskupainallus	N/A	"keskus"
Kosketuslevy – Oikea painallus	N/A	"oikea"
Doc	N/A	"doc"
Tabulaattori	Tabulaattori	"Tabulaattori"
Kosketuslevy – Alapainallus	Nuoli alaspäin	"alas"
Päävalikko	N/A	"päävalikko"
Ctrl	Ctrl	ei palautusta
Vaihto	Vaihto	ei palautusta
Var	N/A	"var"
Poista	N/A	"poista"
=	=	"="

Kannettava laite/Emulaattorinäppäin	Pöytätietokone	Palauta arvo
Trigonometria	N/A	Trigonometria
0:sta 9:ään	0-9	"0" ... 9
Sapluunat	N/A	"sapluuna"
Lista	N/A	"lista"
^	^	"^"
X^2	N/A	"neliö"
/ (jakonäppäin)	/	"/"
* (kertonäppäin)	*	"*"
e^x	N/A	"eksponentti"
10^x	N/A	"10voima"
+	+	"+"
-	-	"_"
(	(	"("
)	)	")"
.	.	"."
(-)	N/A	"_" (negatiivinen merkki)
Syötä	Syötä	"syötä"
ee	N/A	"E" (tieteellinen merkintä E)
a – z	a–z	alfa = kirjainmerkki painettu (pieni kirjain) ("a" – "z")
vaihto a–z	vaihto a–z	alfa = kirjainmerkki painettu "A" – "Z"
		Huomaa: Ctrl-vaihto lukitsee isot kirjaimet
?!	N/A	"?!"
pii	N/A	"pij"
Lippu	N/A	ei palautusta

Kannettava laite/Emulaattorinäppäin	Pöytätietokone	Palauta arvo
,	,	" , "
Palautus	N/A	Palautus
Välilyönti	Välilyönti	Välilyönti
Ei pääsyä	Näppäimet erikoismerkeille, kuten @,!,^, etc.	Kirjainmerkki on palautettu
N/A	Toimintinäppäimet	Kirjainmerkkejä ei ole palautettu
N/A	Eriyiset näytön kontrollinäppäimet	Kirjainmerkkejä ei ole palautettu
Ei pääsyä	Muita näytön näppäimiä, jotka eivät ole käytettävissä laskimessa, kun getKey() odottaa näppäimen painallusta. ({, },,, :, ...)	Sama kirjainmerkki, jonka saat Notesista (ei matemaattisessa kentässä)

Huomaa: On tärkeää huomata, että ohjelman **getKey ()** läsnäolo muuttaa ohjelmaa tiettyjen tapahtumien käsittelyä. Joitakin näistä kuvataan jäljempänä.

Lopeta ohjelma ja käsittele tapahtuma – aivan kuin jos käyttäjä poistuisi ohjelmasta painamalla **ON** -näppäintä

"**Tuki** " jäljempänä tarkoittaa – Järjestelmä toimii odotetusti – ohjelma jatkuu.

Tapahtuma	laite	Työpöytä – TI-Nspire™ Ohjelmisto opiskelijoille
Pikakysely	Lopeta ohjelma, käsittele tapahtuma	Sama kuin kannettava laite (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software – vain)
Etätiedoston hallinta (Sisältää lähetyksen "Exit Press 2 Test" -tiedoston toisesta kämmenlaitteesta tai työpöydän kannettavasta laitteesta)	Lopeta ohjelma, käsittele tapahtuma	Sama kuin kannettava laite. (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-vain)
Lopeta oppitunti	Lopeta ohjelma, käsittele tapahtuma	Tuki (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-vain)

Tapahtuma	Laite	Työpöytä – TI-Nspire™ Kaikki versiot
TI-Innovator™ Hub Yhdistä / katkaise	Tuki – onnistuu komennolla TI-Innovator™ Hub. Kun olet lopettanut ohjelman, TI-Innovator™ Hub työskentelee edelleen kämmenlaitteen kanssa.	Sama kuin kannettava laite

getLangInfo() Katalogi >

getLangInfo() ⇒ *merkkijono*

getLangInfo()

"en"

Antaa merkkijonon, joka vastaa parhaillaan käytössä olevan kielen lyhyttä nimeä. Voit käyttää sitä esimerkiksi ohjelmassa tai funktiossa nykyisen kielen määrittämiseen.

englanti = "en"

tanska = "da"

saksa = "de"

suomi = "fi"

ranska = "fr"

italia = "it"

hollanti = "nl"

flaami = "nl_BE"

norja = "no"

portugali = "pt"

espanja = "es"

ruotsi = "sv"

getLockInfo() Katalogi >

getLockInfo() (*Muutt*) ⇒ *arvo*

a:=65

65

Määrittää muuttujan *Muutt* nykyisen lukittu/lukitsematon-tilan.

Lock a

Done

getLockInfo(a)

1

arvo = 0: *Muutt* on lukitsematon tai sitä ei ole olemassa.

a:=75

"Error: Variable is locked."

DelVar a

"Error: Variable is locked."

arvo = 1: *Muutt* on lukittu eikä sitä voi muuttaa tai poistaa.

Unlock a

Done

a:=75

75

Katso **Lock**, sivu 111, ja **unLock**, sivu 208.

DelVar a

Done

getMode(*TilanNimiKokonaisluku*)⇒*arvo*

getMode(0)⇒*lista*

getMode(*TilanNimiKokonaisluku*) laskee arvon, joka vastaa *TilanNimiKokonaisluku*-tilan nykyistä asetusta.

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

getMode(0) laskee listan, joka sisältää lukupareja. Jokainen pari koostuu tilaa kuvaavasta kokonaisluvusta ja asetusta kuvaavasta kokonaisluvusta.

Tilat ja niiden asetukset on esitetty alla olevassa taulukossa.

Jos tallennat asetukset komennolla

getMode(0) → *muutt*, voit käyttää komentoa **setMode(*muutt*)** funktiossa tai ohjelmassa ja tallentaa asetukset näin väliaikaisesti pelkästään funktion tai ohjelman suorituksen ajaksi. Katso **setMode()**, sivu 171.

Tilan nimi	Tilaa vastaava kokonaisluku	Asetuksia vastaavat kokonaisluvut
Näytettävät numerot	1	1=Liukuva, 2=Liukuva1, 3=Liukuva2, 4=Liukuva3, 5=Liukuva4, 6=Liukuva5, 7=Liukuva6, 8=Liukuva7, 9=Liukuva8, 10=Liukuva9, 11=Liukuva10, 12=Liukuva11, 13=Liukuva12, 14=Kiinteä0, 15=Kiinteä1, 16=Kiinteä2, 17=Kiinteä3, 18=Kiinteä4, 19=Kiinteä5, 20=Kiinteä6, 21=Kiinteä7, 22=Kiinteä8, 23=Kiinteä9, 24=Kiinteä10, 25=Kiinteä11, 26=Kiinteä12
Kulma	2	1=Radiaani, 2=Aste, 3=Graadi
Eksponenttimuoto	3	1=Normaali, 2=Kymmenpotenssi, 3=Tekninen
Reaali- tai kompleksiluku	4	1=Reaali, 2=Suorakulma, 3=Polaarinen
Automaattinen tai likimääräinen.	5	1=Automaattinen, 2=Likimääräinen, 3=Täsmällinen
Vektorimuoto	6	1=Suorakulma, 2=Sylinteri, 3=Pallo

Tilan nimi	Tilaa vastaava kokonaisluku	Asetuksia vastaavat kokonaisluvut
Kantaluku	7	1=Desimaali, 2=Heksagonaalinen, 3=Binaarinen
Yksikköjärjestelmä	8	1=SI, 2=Eng/US

getNum()	Katalogi > 						
getNum (<i>Laus l</i>) $\Rightarrow$ <i>lauseke</i>							
Muuttaa argumentin lausekkeeksi, jolla on sievennetty yhteinen nimittäjä, ja laskee sen jälkeen lausekkeen osoittajan.							
	<table> <tr> <td>$\text{getNum}\left(\frac{x+2}{y-3}\right)$</td> <td>$x+2$</td> </tr> <tr> <td>$\text{getNum}\left(\frac{2}{7}\right)$</td> <td>2</td> </tr> <tr> <td>$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$</td> <td>$x+y$</td> </tr> </table>	$\text{getNum}\left(\frac{x+2}{y-3}\right)$	$x+2$	$\text{getNum}\left(\frac{2}{7}\right)$	2	$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$	$x+y$
$\text{getNum}\left(\frac{x+2}{y-3}\right)$	$x+2$						
$\text{getNum}\left(\frac{2}{7}\right)$	2						
$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$	$x+y$						

GetStr	Hub-valikko
GetStr [<i>kehotemerkkijono</i>], <i>var</i> [, <i>statusVar</i>]	Katso esimerkit kohdasta Get .
GetStr [<i>kehotemerkkijono</i>], <i>func</i> (<i>arg1</i> , ... <i>argn</i>)[, <i>statusVar</i>]	
Ohjelmointikomento: Toimii samalla tavalla kuin Get -komento, mutta vastaanotettu arvo tulkitaan aina merkkijonoksi. Get -komento kuitenkin tulkitsee vastauksen lausekkeeksi, jollei sitä merkitä lainausmerkkien ("") sisään.	
Huomio: Katso myös Get , sivu 82 ja Send sivu 167.	

getType()	Katalogi > 												
getType (<i>var</i>) $\Rightarrow$ <i>merkkijono</i>													
Antaa tulokseksi merkkijonon, joka ilmoittaa muuttujan <i>var</i> datatyyppin.													
Jos muuttujaa <i>var</i> ei ole määritelty, tulokseksi tulee merkkijono "EI MITÄÄN".													
	<table> <tr> <td>$\{1,2,3\} \rightarrow temp$</td> <td>$\{1,2,3\}$</td> </tr> <tr> <td>$\text{getType}(temp)$</td> <td>"LIST"</td> </tr> <tr> <td>$3 \cdot i \rightarrow temp$</td> <td>$3 \cdot i$</td> </tr> <tr> <td>$\text{getType}(temp)$</td> <td>"EXPR"</td> </tr> <tr> <td>$\text{DelVar } temp$</td> <td><i>Done</i></td> </tr> <tr> <td>$\text{getType}(temp)$</td> <td>"NONE"</td> </tr> </table>	$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$	$\text{getType}(temp)$	"LIST"	$3 \cdot i \rightarrow temp$	$3 \cdot i$	$\text{getType}(temp)$	"EXPR"	$\text{DelVar } temp$	<i>Done</i>	$\text{getType}(temp)$	"NONE"
$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$												
$\text{getType}(temp)$	"LIST"												
$3 \cdot i \rightarrow temp$	$3 \cdot i$												
$\text{getType}(temp)$	"EXPR"												
$\text{DelVar } temp$	<i>Done</i>												
$\text{getType}(temp)$	"NONE"												

getVarInfo()⇒*matriisi* tai *merkkijono*

getVarInfo(LibNameString)⇒*matriisi* tai *merkkijono*

getVarInfo() laskee tietomatriisin (muuttujan nimi, tyyppi, kirjaston käytettävyys ja lukittu/lukitsematon-tila) kaikille nykyisessä tehtävässä määritetyille muuttujille ja kirjasto-objekteille.

Jos yhtään muuttujaa ei ole määritetty, **getVarInfo()** antaa vastauksena merkkijonon "NONE".

getVarInfo(KirjNimiMerkkijono) antaa tuloksena tietomatriisin kaikista kirjastossa *KirjNimiMerkkijono* määritetyistä kirjasto-objekteista. *KirjNimiMerkkijono* on oltava merkkijono (lainausmerkkien sisällä oleva teksti) tai merkkijonomuuttuja.

Jos kirjastoa *KirjNimiMerkkijono* ei ole olemassa, esiintyy virhe.

Huomaa vasemmanpuoleinen esimerkki, jossa funktion **getVarInfo()** vastaus on määritetty muuttujaan *vs*. Jos muuttujan *vs* riviä 2 tai riviä 3 yritetään näyttää, tuloksena on "Kelpaamaton lista tai matriisi" -virhe, koska vähintään yksi näiden rivien elementeistä (esimerkiksi muuttuja *b*) sieventyy uudelleen matriisiksi.

Tämä virhe voi esiintyä myös käytettäessä *Ans*-muuttujaa funktion **getVarInfo()** tuloksen uudelleenlaskennassa.

Järjestelmä antaa edellä mainitun virheen, koska ohjelmiston nykyinen versio ei tue yleistettyä matriisirakennetta, jossa matriisin elementti voi olla joko matriisi tai lista.

getVarInfo()	"NONE"												
Define x=5	Done												
Lock x	Done												
Define LibPriv y={ 1,2,3 }	Done												
Define LibPub z(x)=3·x ² -x	Done												
getVarInfo()	<table><tr><td>x</td><td>"NUM"</td><td>"{"</td><td>1</td></tr><tr><td>y</td><td>"LIST"</td><td>"LibPriv "</td><td>0</td></tr><tr><td>z</td><td>"FUNC"</td><td>"LibPub "</td><td>0</td></tr></table>	x	"NUM"	"{"	1	y	"LIST"	"LibPriv "	0	z	"FUNC"	"LibPub "	0
x	"NUM"	"{"	1										
y	"LIST"	"LibPriv "	0										
z	"FUNC"	"LibPub "	0										
getVarInfo(tmp3)	"Error: Argument must be a string"												
getVarInfo("tmp3")	<table><tr><td>[volcyl2</td><td>"NONE"</td><td>"LibPub "</td><td>0]</td></tr></table>	[volcyl2	"NONE"	"LibPub "	0]								
[volcyl2	"NONE"	"LibPub "	0]										

$a:=1$	1												
$b:=[1\ 2]$	$[1\ 2]$												
$c:=[1\ 3\ 7]$	$[1\ 3\ 7]$												
$vs:=\text{getVarInfo}()$	<table><tr><td>a</td><td>"NUM"</td><td>"{"</td><td>0</td></tr><tr><td>b</td><td>"MAT"</td><td>"{"</td><td>0</td></tr><tr><td>c</td><td>"MAT"</td><td>"{"</td><td>0</td></tr></table>	a	"NUM"	"{"	0	b	"MAT"	"{"	0	c	"MAT"	"{"	0
a	"NUM"	"{"	0										
b	"MAT"	"{"	0										
c	"MAT"	"{"	0										
$vs[1]$	$[1\ \text{"NUM"}\ \text{"{"}\ 0]$												
$vs[1,1]$	1												
$vs[2]$	"Error: Invalid list or matrix"												
$vs[2,1]$	$[1\ 2]$												

Goto *tunnusnimi*

Siirtää ohjauksen tunnuksen *tunnusnimi*.

tunnusnimi on määritettävä samassa funktiossa käyttäen **Lbl**-ohjetta.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Define $g()$ =Func	Done
Local <i>temp,i</i>	
$0 \rightarrow temp$	
$1 \rightarrow i$	
Lbl <i>top</i>	
$temp+i \rightarrow temp$	
If $i < 10$ Then	
$i+1 \rightarrow i$	
Goto <i>top</i>	
EndIf	
Return <i>temp</i>	
EndFunc	
$g()$	55

►Grad

Laus1 ► **Grad** ⇒ *lauseke*

Muuttaa *Laus1*:n graadikulmaan.

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla **@>Grad**.

Astekulmatilassa:

$(1.5) \blacktriangleright \text{Grad}$ $(1.66667)^9$

Radiaanikulmatilassa:

$(1.5) \blacktriangleright \text{Grad}$ $(95.493)^9$

/

identity()

identity(kokonaisluku) ⇒ *matriisi*

Laskee identiteettimatriisin, jonka koko on *kokonaisluku*.

Kokonaisluvun on oltava positiivinen kokonaisluku.

identity(4)	<table><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
1	0	0	0														
0	1	0	0														
0	0	1	0														
0	0	0	1														

Jos *BooleanLaus*
Ilmaisut

Define $g(x)$ =Func	Done
If $x < 0$ Then	
Return x^2	
EndIf	
EndFunc	

Jos *BooleanLaus Niin*
Lohko

EndIf

$g(-2)$	4
---------	---

Jos BooleanLaus on tosi, suorittaa yhden lausekkeen *Lauseke* tai lausekkeiden lohkon *Lohko* ennen suorituksen jatkamista.

Jos BooleanLaus on epätosi, jatkaa suoritusta suorittamatta lauseketta tai lausekkeiden lohkoa.

Lohko voi olla joko yksi lauseke tai sarja lausekkeita, jotka on erotettu toisistaan kaksoispisteellä (:) merkki.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Jos *BooleanLaus, niin*
Lohko1

Tai

Lohko2

EndIf

Define $g(x)$ =Func	Done
If $x < 0$ Then	
Return $-x$	
Else	
Return x	
EndIf	
EndFunc	

Jos BooleanLaus on tosi, suorittaa *Lohko1*:n ja sen jälkeen jättää väliin *Lohko2*:n.

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

Jos BooleanLaus on epätosi, ohittaa *Lohko1*:n, mutta suorittaa *Lohko2*:n.

Lohko1 ja *Lohko2* voivat olla yksi lauseke.

Jos *BooleanLaus1*, **niin**

Lohko1

Jos taas *BooleanLaus2*, **niin**

Lohko2

:

Jos taas *BooleanLausN*, **niin**

LohkoN

EndIf

Sallii haarautumisen Jos *BooleanLaus1* on tosi, suorittaa *Lohko1*:n Jos *BooleanLaus1* on epätosi, laskee *BooleanLaus2*:n jne.

Define $g(x)$ = Func

If $x < 5$ 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

$g(-4)$	4
$g(10)$	3

ifFn()

Luettelo >

ifFn(*BooleanLaus*, *Arvo_Jos_tosi* [, *Arvo_Jos_epätosi* [, *Arvo_Jos_tuntematon*]]) $\Rightarrow$ lauseke, lista tai matriisi

Laskee *BooleanLaus* (jokaiselle *BooleanLaus*) elementille) ja antaa tuloksen noudattaen seuraavia sääntöjä:

- BooleanLaus* voi testata yksittäisen arvon, listan tai matriisin.
- Jos jokin *BooleanLaus* elementti on tosi, laskee vastaavan elementin lausekkeesta *Arvo_Jos_tosi*.
- Jos jokin *BooleanLaus* elementti on epätosi, laskee vastaavan elementin lausekkeesta *Arvo_Jos_epätosi*. Jos jätät pois lausekkeen *Arvo_Jos_epätosi*, laskee määrittelemättömäksi.
- Jos *BooleanLaus* elementti ei ole tosi eikä epätosi, laskee vastaavan elementin *Arvo_If_unknown*. Jos jätät pois *Arvo_If_unknown*, laskee määrittelemättömäksi.
- Jos funktion **josFn()** toinen, kolmas tai neljäs argumentti on yksi lauseke, Boolean testiä sovelletaan jokaiseen sijaintiin Boolean lausekkeessa *BooleanLaus*.

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

1:n testiarvo on alle 2.5, joten sen vastaava

Arvo_Jos_tosi -elementti arvolle **5**
kopioidaan vastausten listaan.

2:n testiarvo on alle 2.5, joten sitä vastaava

Arvo_Jos_tosi -elementti arvolle **6**
kopioidaan vastausten listaan.

3:n testiarvo ei ole alle 2.5, joten sitä vastaava *Arvo_Jos_epätosi*-elementti **10**
kopioidaan vastausten listaan.

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

Arvo_Jos_tosi on yksittäinen arvo ja vastaa mitä tahansa valittua sijaintia.

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

Arvoa *Arvo_Jos_epätosi* ei ole määriteltä. Käytetään merkintää EIMää

ifFn()Luettelo > 

Huomaa: Jos sievennetty *BooleanLaus* ilmaisu sisältää listan tai matriisin, kaikkien muiden lista- tai matriisiargumenttien on oltava samansuuruisia, ja myös tuloksen on oltava samansuuruinen.

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

$$\{6, "err"\}$$

Yksi elementti valittu lausekkeesta *Arvo_Jos_tosi*. Yksi elementti valittu lausekkeesta *Arvo_Jos_epätosi*.

imag()Luettelo > 

imag(*ExprI*) $\Rightarrow$ *lauseke*

Laskee argumentin imaginaarisen osan.

Huomaa: Kaikkia määrittelemättömiä muuttujia käsitellään reaali muuttujina. Katso myös *real()*, page 153

imag(*ListaI*) $\Rightarrow$ *lista*

Laskee listan alkutekijöiden imaginaarisista osista.

imag(*MatriisiI*) $\Rightarrow$ *matriisi*

Laskee matriisin alkutekijöiden imaginaarisista osista.

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

$$2$$

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

$$0$$

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

$$y$$

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

$$\{0, -1, 1\}$$

$$\text{imag}\left(\begin{bmatrix} a & b \\ i \cdot c & i \cdot d \end{bmatrix}\right)$$

$$\begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$$

impDif()Luettelo > 

impDif(*Yhtälö*, *Muutt*, *riippuvaMuutt* [, *Aste*]) $\Rightarrow$ *lauseke*

jossa luokan *Aste* oletusarvo on 1

Laskee implisiittisen derivaatan yhtälöille, joissa yksi muuttuja määritellään implisiittisesti toisen suhteen

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

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

Epäsuora operaattori

Katso #(), sivu 238.

inString()Luettelo > 

inString(*srcMerkkijono*, *alaMerkkijono* [, *Alku*]) $\Rightarrow$ *kokonaisluku*

$$\text{inString}("Hello there", "the")$$

$$7$$

$$\text{inString}("ABCEFG", "D")$$

$$0$$

Laskee merkin paikan merkkijonossa *srcMerkkijono*, jossa merkkijonon *alaMerkkijono* ensimmäinen esiintyminen alkaa.

Alku, jos se sisältyy, määrää merkin paikan siinä merkkijonossa *srcMerkkijono*, josta haku alkaa. Oletusarvo = 1 (*srcMerkkijonon* ensimmäinen merkki).

Jos *srcMerkkijono* ei sisällä *alaMerkkijonoa* tai *Alku* on *srcMerkkijonon* pituus, vastaus on nolla.

int()

int(Laus) $\Rightarrow$ kokonaisluku

$\text{int}\{-2.5\}$	-3.
----------------------	-----

int(Lista1) $\Rightarrow$ lista

$\text{int}[\{-1.234 \ 0 \ 0.37\}]$	$[-2. \ 0 \ 0.]$
-------------------------------------	------------------

int(Matriisi1) $\Rightarrow$ matriisi

Laskee suurimman kokonaisluvun, joka on pienempi tai yhtä suuri kuin argumentti. Tämä funktio on identtinen funktion pohjan **floor()** kanssa.

Argumentti voi olla reaali- tai kompleksiluku.

Kun kyseessä on lista tai matriisi, laskee kunkin elementin suurimman kokonaisluvun.

intDiv()

intDiv(Luku1, Luku2) $\Rightarrow$ kokonaisluku

$\text{intDiv}\{-7,2\}$	-3
-------------------------	----

intDiv(Lista1, Lista2) $\Rightarrow$ lista

$\text{intDiv}\{4,5\}$	0
------------------------	---

intDiv(Matriisi1, Matriisi2) $\Rightarrow$ matriisi

$\text{intDiv}\{\{12, -14, -16\}, \{5, 4, -3\}\}$	$\{2, -3, 5\}$
---	----------------

Laskee lausekkeen $\text{Luku1} \div \text{Luku2}$ etumerkillisen kokonaislukuosan.

Laskee sarjoille ja matriiseille lausekkeen (argumentti 1 $\div$ argumentti 2) etumerkillisen kokonaislukuosan kullekin elementtiparille.

interpoloi()

Luettelo >

Interpoloi(*xArvo*, *xList*, *yList*, *yPrimeList*)
 $\Rightarrow$ lista

Tällä toiminnolla suoritetaan seuraavaa:

Kun ilmoitetaan *xList*, *yList*=**f**(*xList*) ja *yPrimeList*=**f'**(*xList*) jollekin tuntemattomalle funktiolle **f**, käytetään kuutiointerpolanttia funktion **f** määrittelemiseksi arvolla *xArvo*. Oletetaan, että *xList* on monotonisesti kasvavien tai laskevien numeroiden lista, mutta tämän toiminnon tuloksena saattaa olla arvo, vaikka se ei olisikaan sitä. Tämä toiminto käy läpi listan *xList* etsien väliä [*xList*[*i*], *xList*[*i*+1]], joka sisältää arvon *xValue*. Jos se löytää tällaisen välin, se palauttaa interpoloidun arvon funktiolle **f** (*xValue*); muuten se antaa tuloksen **määrittelemätön**.

Sarjojen *xList*, *yList* ja *yPrimeList* on oltava samansuuruiset ≥ 2 ja sisällettävä lausekkeita, jotka sieventyvät luvuiksi.

xValue voi olla määrittelemätön muuttuja, luku tai lukuluettelo.

Differentiaaliyhtälö:

$$y' = -3 \cdot y + 6 \cdot t + 5 \text{ and } y(0) = 5$$

$rk := rk23(-3 \cdot y + 6 \cdot t + 5, t, y, \{0, 10\}, 5, 1)$					
0.	1.	2.	3.	4.	
5.	3.19499	5.00394	6.99957	9.00593	10

Jos haluat nähdä koko vastauksen, paina $\blacktriangle$ ja siirrä sen jälkeen osoitinta painikkeilla $\blacktriangleleft$ ja $\blacktriangleright$.

Käytä interpolate()-funktioita laskeaksesi funktion arvot *x*-arvolistalle:

$xvalueList := seq(i, 0, 10, 0.5)$	
{0, 0.5, 1., 1.5, 2., 2.5, 3., 3.5, 4., 4.5, 5., 5.5, 6., 6.5, 7., 7.5, 8., 9., 10.}	
$xList := mat \blacktriangleright list(rk[1])$	
{0., 1., 2., 3., 4., 5., 6., 7., 8., 9., 10.}	
$yList := mat \blacktriangleright list(rk[2])$	
{5., 3.19499, 5.00394, 6.99957, 9.00593, 10.9978}	
$yPrimeList := -3 \cdot y + 6 \cdot t + 5 y = yList \text{ and } t = xList$	
{-10., 1.41503, 1.98819, 2.00129, 1.98221, 2.0067}	
$interpolate(xvalueList, xList, yList, yPrimeList)$	
{5., 2.67062, 3.19499, 4.02782, 5.00394, 6.00011}	

inv χ^2 ()

Luettelo >

inv χ^2 (*Ala*, *df*)

invChi2(*Ala*, *df*)

Laskee käänteisen kumulatiivisen χ^2 (chi-neliö) -todennäköisyysfunktion, joka on määriteltä vapauden asteella, *df* annettulle käyrän alapuoliselle alalle *Ala*.

invF()

Luettelo >

invF(*Ala*, *dfOsoitt*, *dfNimitt*)

vakioF(*Area*, *dfOsoitt*, *dfNimitt*)

Laskee käänteisen kumulatiivisen F-jakaumafunktion, jolle on määritetty $dfOsoitt$ ja $dfNimitt$, annetulle käyrän alapuoliselle alueelle Ala .

invBinom()

Luettelo >

invBinom

$(CumulativeProb, NumTrials, Prob, OutputForm) \Rightarrow$ asteikkomuoto tai matriisi

Käänteinen binomi. Johtuen kokeiden ($NumTrials$) ja kunkin kokeen todennäköisyydestä onnistua ($Prob$), tämä toiminto laskee onnistumisten minimimäärän, ksiten, että arvo k on suurempi tai yhtä suuri kuin kumuloituva todennäköisyys ($CumulativeProb$).

$OutputForm=0$, näyttää tuloksen asteikkomuodossa (oletus).

$OutputForm=1$, näyttää tuloksen matriisina.

Esimerkki: Mary ja Kevin pelaavat noppapeliä. Maryn on arvattava, miten monta kertaa numero 6 enintään esiintyy 30 heittoa kohti. Jos numero 6 esiintyy yhtä monta kertaa tai vähemmän, Mary voittaa. Lisäksi, mitä pienempi on hänen arvaamansa määrä, sitä suuremmat ovat hänen voittonsa. Mikä on pienin määrä, jonka Mary voi arvata, jos hän haluaa voittamisen todennäköisyyden olevan suurempi kuin 77 %?

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

invBinomN()

Luettelo >

$\text{invBinomN}(CumulativeProb, Prob, NumSuccess, OutputForm) \Rightarrow$ asteikkomuoto tai matriisi

$N:n$ suhteen käänteisesti binominen. Johtuen menestyksen todennäköisyydestä kussakin kokeessa ($Prob$), onnistumisten määrä ($NumSuccess$), tämä funktio laskee kokeiden vähimmäismäärän N siten, että arvo, N , on vähemmän tai yhtä suuri kuin annettu kumulatiivinen todennäköisyys ($CumulativeProb$).

$OutputForm=0$, näyttää tuloksen asteikkomuodossa (oletus).

$OutputForm=1$, näyttää tuloksen matriisina.

Esimerkki: Monique harjoittelee koriin heittoa koripallossa. Hän tietää kokemuksesta, että hänen mahdollisuutensa tehdä kori on 70 %. Hän suunnittelee harjoittelevansa, kunnes hän on tehnyt 50 koria. Kuinka montaa koria hänen on yritettävä varmistaakseen, että todennäköisyys tehdä ainakin 50 koria on enemmän kuin 0,99?

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

invNorm()Luettelo > **invNorm(Ala[,μ[,σ]])**

Laskee käänteisen kumulatiivisen normaalijakaumafunktion annetulle alalle *Ala*, joka on normaalijakaumakäyrän alapuolella ja jonka määräävät μ ja σ .

invt()Luettelo > **invt(Ala,df)**

Laskee käänteisen kumulatiivisen student t -todennäköisyysfunktion, jonka määräävät vapausaste, *df* ja annettu alue *Ala* käyrän alapuolella.

iPart()Luettelo > **iPart(Luku) $\Rightarrow$ kokonaisluku****iPart(Lista1) $\Rightarrow$ lista****iPart(Matriisi1) $\Rightarrow$ matriisi**

$\text{iPart}(-1.234)$	-1.
$\text{iPart}\left(\left\{\frac{3}{2}, -2.3, 7.003\right\}\right)$	$\{1, -2, 7\}$

Laskee argumentin kokonaisosan.

Laskee sarjoille ja matriiseille kunkin elementin kokonaisosan.

Argumentti voi olla reaali- tai kompleksiluku.

irr()Luettelo > **irr(CF0,CFLista[,CFFrekv]) $\Rightarrow$ arvo**

Talouseläämätoiminto, joka laskee investoinnin sisäisen korkokannan.

CF0 on kassavirta alussa aikana 0; arvon on oltava kokonaisluku.

CFLista on lista kassavirtamääristä alun kassavirran *CF0* jälkeen.

$\text{list1} := \{6000, -8000, 2000, -3000\}$	$\{6000, -8000, 2000, -3000\}$
$\text{list2} := \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$\text{irr}(5000, \text{list1}, \text{list2})$	-4.64484

CFFrekv on valinnainen lista, jossa kukin elementti määrää esiintymisfrekvenssin ryhmitetylle (peräkkäiselle) kassavirtamäärälle, joka on *CFFrekv*:n vastaava alkutekijä. Oletusarvo on 1; jos syötät arvoja, niiden on oltava positiivisia kokonaislukuja ja $< 10\,000$.

Huomaa: Katso myös `mirr()`, sivu 121.

isPrime()

isPrime(*Luku*) $\Rightarrow$ *Boolean vakiolauseke*

Laskee totuusarvon tosi tai epätosi ilmaistakseen, onko *luku* kokonaisluku ≥ 2 , joka on tasan jaollinen vain itsellään ja ykkösellä (1)

Jos *Luku* on pitempi kuin 306 numeroa, eikä siinä ole tekijöitä ≤ 1021 , kaava **isPrime** (*Luku*) näyttää virheilmoituksen.

Jos haluat pelkästään määrätä, onko *Luku* jaoton, käytä komentoa **isPrime()** funktion **factor()**sijaan. Se on paljon nopeampi, erityisesti jos *Lukuei* ole jaoton ja sen toiseksi suurimmassa tekijässä on enemmän kuin viisi numeroa.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

isPrime(5)	true
isPrime(6)	false

Funktio, jolla etsitään seuraava jaoton luku määrätyn luvun jälkeen_

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

isVoid()

isVoid(*Muutt*) $\Rightarrow$ *Boolean vakiolauseke*
isVoid(*Laus*) $\Rightarrow$ *Boolean vakiolauseke*
isVoid(*Lista*) $\Rightarrow$ *lista Boolean vakiolausekkeista*

Laskee totuusarvon tosi tai epätosi ilmaisten, onko argumentti tyhjä datatyyppi.

Lisätietoja tyhjiä elementeistä, katso sivu 249.

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

Lbl		Katalogi > 
Lbl <i>tunnusnimi</i>	Define $g^{(f)} = \text{Func}$	Done
Määrittää funktion sisällä tunnuksen, jonka nimi on <i>tunnusnimi</i> .	Local <i>temp, i</i>	
	$0 \rightarrow \text{temp}$	
	$1 \rightarrow i$	
Ohjeella Siirry <i>tunnusnimi</i> voit siirtää ohjauksen kyseistä tunnusta välittömästi seuraavaan ohjaukseen.	Lbl <i>top</i>	
	$\text{temp} + i \rightarrow \text{temp}$	
<i>tunnusnimellä</i> on samat nimeämissäännöt kuin muuttujan nimellä.	If $i < 10$ Then	
	$i + 1 \rightarrow i$	
	Goto <i>top</i>	
	EndIf	
	Return <i>temp</i>	
	EndFunc	
Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.	$g^{(f)}$	55

lcm()		Katalogi > 
lcm (<i>Luku1, Luku2</i>) $\Rightarrow$ <i>lauseke</i>	$\text{lcm}(6,9)$	18
lcm (<i>Lista1, Lista2</i>) $\Rightarrow$ <i>lista</i>		
lcm (<i>Matriisi1, Matriisi2</i>) $\Rightarrow$ <i>matriisi</i>	$\text{lcm}\left(\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}\right)$	$\left\{\frac{2}{3}, 14, 80\right\}$
Laskee kahden argumentin pienimmän yhteisen jaettavan. Kahden murtoluvun lcm on niiden osoittajien lcm jaettuna niiden nimittäjien gcd :llä. Murtoluvun liukulukujen lcm on niiden tulo.		
Kun kyseessä on kaksi listaa tai matriisia, laskee vastaavien elementtien pienimmät yhteiset jakajat.		

left()		Katalogi > 
left (<i>lähdemerkkijono</i> [, <i>Num</i>]) $\Rightarrow$ <i>merkkijono</i>	$\text{left}(\text{"Hello"}, 2)$	"He"
Määrittää vasemmanpuoleisimmat <i>Num</i> -merkit, jotka sisältyvät merkkijonoon <i>lähdemerkkijono</i> .		
Jos jätät pois komennon <i>Num</i> , määrittää kaiken merkkijonosta <i>lähdemerkkijono</i> .		

left()

Katalogi > 

left(*Listal* [, *Num*]) ⇒ *lista*

$\text{left}(\{1, 3, -2, 4\}, 3)$	$\{1, 3, -2\}$
-----------------------------------	----------------

Määrittää vasemmanpuoleisimmat *Num*-elementit, jotka sisältyvät listaan *Listal*.

Jos jätät pois komennon *Num*, määrittää kaiken listasta *Listal*.

left(*Vertailu*) ⇒ *lauseke*

$\text{left}(x < 3)$	x
----------------------	-----

Laskee yhtälön tai epäyhtälön vasemman puolen.

libShortcut()

Katalogi > 

libShortcut(*KirjNimiMerkkijono*,
PikavalNimiMerkkijono
[, *KirjYksitLippu*]) ⇒ *muuttujalista*

Luo muuttujaryhmän nykyiseen ongelmaan, joka sisältää viittauksia kaikkiin määritetyn kirjastoasiakirjan *kirjNimiMerkkijono* sisältämiin objekteihin. Lisää myös ryhmän jäsenet Muuttujat-valikkoon. Tällöin voit viitata kuhunkin objektiin käyttäen sen komentoa *PikavalNimimerkkijono*.

Aseta *KirjYksitLippu*=0, kun haluat sulkea pois yksityiset kirjasto-objektit (oletusarvo)
Aseta *KirjYksitLippu*=1, kun haluat sisällyttää yksityiset kirjasto-objektit

Muuttujaryhmän kopioiminen, katso

CopyVar sivulla sivu 30.

Muuttujaryhmän poistaminen, katso **DelVar** sivulla sivu 51.

Tässä esimerkissä edellytetään asianmukaisesti tallennettua ja päivitettyä kirjastoasiakirjaa nimeltä **linalg2**, joka sisältää objektit *clearmat*, *gauss1* ja *gauss2*.

$\text{getVarInfo}(\text{"linalg2"})$			
$\left[\begin{array}{llll} \text{clearmat} & \text{"FUNC"} & \text{"LibPub"} & \text{" "} \end{array} \right.$			
$\left. \begin{array}{llll} \text{gauss1} & \text{"PRGM"} & \text{"LibPriv"} & \text{" "} \end{array} \right]$			
$\left[\begin{array}{llll} \text{gauss2} & \text{"FUNC"} & \text{"LibPub"} & \text{" "} \end{array} \right.$			
$\left. \begin{array}{llll} \text{gauss2} & \text{"FUNC"} & \text{"LibPub"} & \text{" "} \end{array} \right]$			
$\text{libShortcut}(\text{"linalg2"}, \text{"la"})$			
$\{ \text{la.clearmat}, \text{la.gauss2} \}$			
$\text{libShortcut}(\text{"linalg2"}, \text{"la"}, 1)$			
$\{ \text{la.clearmat}, \text{la.gauss1}, \text{la.gauss2} \}$			

limit(Lausl, Muutt, Piste[, Suunta]) $\Rightarrow$ lauseke**raja-arvo(Listal, Muutt, Piste [, Suunta])** $\Rightarrow$ lista**raja-arvo(Matriisil, Muutt, Piste [, Suunta])** $\Rightarrow$ matriisi

Laskee pyydetyn raja-arvon.

Huomaa: Katso myös **Raja-arvomalli**, sivu 7.

Suunta: negatiivinen=vasemmalta, positiivinen=oikealta, muu=molemmat. (Jos ohje jätetään pois, *Suunta* on oletusarvoisesti 'molemmat'.)

Positiivisen ∞ :n ja negatiivisen ∞ :n kohdalla olevat raja-arvot muunnetaan aina äärellisen puolen yksipuolisiksi raja-arvoiksi.

Tilanteesta riippuen **limit()** antaa vastauksena itsensä tai 'undef' silloin, kun se ei pysty määrittämään yksilöllistä raja-arvoa. Tämä ei välttämättä tarkoita, että yksilöllistä raja-arvoa ei ole olemassa. undef tarkoittaa, että vastaus on joko tuntematon luku, jonka suuruus on äärellinen tai ääretön, tai se on tällaisten lukujen koko sarja.

limit() käyttää menetelmiä, kuten L'Hopitalin sääntöä, joten on olemassa yksilöllisiä raja-arvoja, joita se ei pysty määrittämään. Jos *Lausl* sisältää muita määrittämättömiä muuttujia kuin *Muutt*, joudut mahdollisesti rajoittamaan niitä saadaksesi suppeamman tuloksen.

Raja-arvot voivat olla hyvin herkkiä pyöristysvirheille. Mikäli mahdollista, vältä **Auto or Approximate (Automaattinen tai Likimääräinen)** -tilan Approximate (Likimääräinen) -asetusta ja likiarvoista luvut laskeessasi raja-arvoja. Muussa tapauksessa ne raja-arvot, joiden pitäisi olla nolla tai jotka ovat äärettömiä, eivät todennäköisesti ole tätä, ja raja-arvot, joiden pitäisi olla äärellisiä ei-nolla-arvoja, eivät välttämättä ole sitä.

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

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

LinRegBx $X, Y, [Frekv], [Luokka, Sisällyttä]$

Laskee lineaarisen regressioyhtälön $a + b \cdot x$ listaista X ja Y frekvenssillä $Frekv$. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällyttä*-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

$Frekv$ on valinnainen frekvenssiarvojen lista. Jokainen $Frekv$:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

$Luokka$ on luokkakoodien lista vastaavalle X - ja Y -datalle..

Sisällyttä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a + b \cdot x$
stat.a, stat.b	Regressiokertoimet
stat.r ²	Määrittäyskerroin
stat.r	Korrelaatiokerroin
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun Y Listan sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällyttä</i> luokat rajoitusten mukaisesti
stat.YReg	Muokatun Y Listan sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällyttä</i> luokat rajoitusten mukaisesti
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

LinRegMx $X, Y, [Frekv], [Luokka, Sisällytä]$

Laskee lineaarisen regression $y = m \cdot x + b$ listoista X ja Y frekvenssillä *Frekv*. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle X - ja Y -datalle.

Sisällytä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $m \cdot x + b$
stat.m, stat.b	Regressiokertoimet
stat.r ²	Määrittäyskerroin
stat.r	Korrelaatiokerroin
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä</i> luokat rajoitusten mukaisesti
stat.YReg	Muokatun <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä</i> luokat rajoitusten mukaisesti
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

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

Kulmakerroin. Laskee tason C luottamusvälin kulmakertoimelle.

LinRegtIntervals $X, Y, F[,1, Xarvo[, CTaso]]]$

Vaste. Laskee ennustetun y :n arvon, tason C ennustevälin yhdelle havainnolle ja tason C luottamusvälin keskiarvovasteelle.

Tulosten yhteenveto tallentuu stat.results-muuttujaan. (Katso sivu sivu 186.)

Kaikkien listojen on oltava samankokoisia.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

F on valinnainen frekvenssiarvojen lista. Jokainen F :n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja ≥ 0 .

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a+b \cdot x$
stat.a, stat.b	Regressiokertoimet
stat.df	Vapausasteet
stat.r ²	Määrittyskerroin
stat.r	Korrelaatiokerroin
stat.Resid	Regressioyhtälön jäännökset

Vain Kulmakerroin-tyyppi

Tulosmuuttuja	Kuvaus
[stat.CLower, stat.CUpper]	Kulmakertoimen luottamusväli
stat.ME	Luottamusvälin virhemarginaali

Tulosmuuttuja	Kuvaus
stat.SESlope	Kulmakertoimen keskivirhe
stat.s	Keskivirhe suoran ympärillä

Vain Vaste-tyyppi

Tulosmuuttuja	Kuvaus
[stat.CLower, stat.CUpper]	Keskiarvovasteen luottamusväli
stat.ME	Luottamusvälin virhemarginaali
stat.SE	Keskiarvovasteen keskivirhe
[stat.LowerPred, stat.UpperPred]	Yhden havainnon ennusteväli
stat.MEPred	Ennustevälin virhemarginaali
stat.SEPred	Ennusteen keskivirhe
stat. $\hat{y}$	$a + b \cdot X$ Arvo

LinRegtTest

Katalogi > 

LinRegtTest $X, Y[, Frekv[, Hypot]]$

Laskee lineaarisen regression X - ja Y -listoista ja suorittaa t -testin kulmakertoimen β ja korrelaatiokertoimen ρ arvosta yhtälölle $y = \alpha + \beta x$. Testaa nollahypoteesia $H_0: \beta = 0$ (vastaavasti, $\rho = 0$) johonkin kolmesta vaihtoehtoisesta hypoteesista.

Kaikkien listojen on oltava samankokoisia.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

$Frekv$ on valinnainen frekvenssiarvojen lista. Jokainen $Frekv$:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

$Hypot$ on valinnainen arvo, joka määrittää yhden kolmesta hypoteesista, johon nollahypoteesia ($H_0: \beta = \rho = 0$) testataan.

Kun $H_0: \beta \neq 0$ ja $p \neq 0$ (oletus), aseta $Hypot=0$

Kun $H_a: \beta < 0$ ja $p < 0$, aseta $Hypot < 0$

Kun $H_a: \beta > 0$ ja $p > 0$, aseta $Hypot > 0$

Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu sivu 186.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementtisivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a + b \cdot x$
stat.t	<i>t</i> -tilasto merkitsevyydestille
stat.PVal	Alin merkitsevyytaso, jolla nollahypoteesi voidaan hylätä
stat.df	Vapausasteet
stat.a, stat.b	Regressiokertoimet
stat.s	Keskivirhe suoran ympärillä
stat.SESlope	Kulmakertoimen keskivirhe
stat.r ²	Määrittyskerroin
stat.r	Korrelaatiokerroin
stat.Resid	Regressioyhtälön jäännökset

linSolve()

linSolve(*Lineaariyhtälöryhmä*, *Muutt1*, *Muutt2*, ...)⇒*lista*

linSolve(*Lineaariyht1* ja *Lineaariyht2* ja ..., *Muutt1*, *Muutt2*, ...)⇒*lista*

linSolve({*Lineaariyht1*, *Lineaariyht2*, ...}, *Muutt1*, *Muutt2*, ...)⇒*lista*

linSolve(*Lineaariyhtälöryhmä*, {*Muutt1*, *Muutt2*, ...})⇒*lista*

linSolve(*Lineaariyht1* ja *Lineaariyht2* ja

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

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

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

$$\left\{ \frac{13}{3}, \frac{14}{3} \right\}$$

$$\text{linSolve}\left(\begin{cases} \text{apple} \cdot 4 + \frac{\text{pear}}{3} = 14 \\ -\text{apple} + \text{pear} = 6 \end{cases}, \{\text{apple}, \text{pear}\}\right)$$

$$\left\{ \frac{36}{13}, \frac{114}{13} \right\}$$

...,
 $\{Muutt1, Muutt2, \dots\} \Rightarrow lista$

linSolve($\{Lineaariyht1, Lineaariyht2, \dots\},$
 $\{Muutt1, Muutt2, \dots\}$)
 $\Rightarrow lista$

Laskee ratkaisulistan muuttujille *Muutt1*,
Muutt2, ...

Ensimmäisen argumentin sievennyksen tuloksena on oltava lineaariyhtälöryhmä tai yksi lineaariyhtälö. Muussa tapauksessa esiintyy argumenttivirhe.

Esimerkiksi yhtälön **linSolve**($x=1$ and $x=2,x$) sieventäminen antaa tuloksena virheilmoituksen Argumenttivirhe.

ΔList()

ΔList(*Listal*) $\Rightarrow lista$

$\Delta List(\{20,30,45,70\}) \quad \{10,15,25\}$

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **deltaList** (...).

Määrittää listan, joka sisältää *Listal*:n peräkkäisten elementtien väliset erotukset. Jokainen *Listal*:n elementti vähennetään *Listal*:n seuraavasta elementistä. Tuloksena oleva lista on aina yhden elementin lyhyempi kuin alkuperäinen *Listal*.

list▶mat()

list▶mat(*Listal* [, *elementtiäRivillä*])
 $\Rightarrow matriisi$

$list▶mat(\{1,2,3\}) \quad \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$

$list▶mat(\{1,2,3,4,5\},2) \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$

Laskee matriisin, joka on täytetty rivi riviltä *Listan* elementeillä.

elementtiäRivillä, mikäli sisällytetään, määrittää elementtien määrän rivillä. Oletusarvo on *Listan* elementtien määrä (yksi rivi).

Jos *Lista* ei täytä tulosmatriisia, siihen lisätään nolliä.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla `list@>mat` (...).

Laus ►ln⇒*lauseke*

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

Aiheuttaa sen, että lausekkeen *Laus* syöte muunnetaan luonnollisia logaritmeja (ln) sisältäväksi lausekkeeksi.

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla `@>ln`.

ln(*Laus*)⇒*lauseke*

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

ln(*Lista*)⇒*lista*

Määrittää argumentin luonnollisen logaritmin.

Jos kompleksilukumuodon tila on Reaali:

$$\ln(\{-3, 1.2, 5\})$$

"Error: Non-real calculation"

Jos kyseessä on lista, määrittää elementtien luonnolliset logaritmit.

Jos kompleksilukumuodon tila on Suorakulma:

$$\ln(\{-3, 1.2, 5\}) \quad \{\ln(3) + \pi \cdot i, 0.182322, \ln(5)\}$$

ln(*neliömatriisi*)⇒*neliömatriisi*

Määrittää *neliömatriisi* *l*:n matriisin luonnollisen logaritmin. Tämä ei ole sama kuin kunkin elementin luonnollisen logaritmin laskeminen. Laskentamenetelmä on kuvattu kohdassa `cos()`.

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

neliömatriisi *l*:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Jos haluat nähdä koko vastauksen, paina ▲
ja siirrä sen jälkeen osoitinta painikkeilla
◀ ja ▶.

LnReg

Katalogi > 

LnReg X , Y , [*Frekv*] [, *Luokka*, *Sisällytä*]

Laskee logaritmisin regressioon $y = a + b \cdot \ln(x)$ listoista X ja Y frekvenssillä *Frekv*. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia
Sisällytä-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle X - ja Y -datalle.

Sisällytä on yhden tai useamman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a + b \cdot \ln(x)$
stat.a, stat.b	Regressiokertoimet
stat.r ²	Muunnettujen tietojen lineaarimäärittelyn kerroin
stat.r	Muunnettujen tietojen korrelaatiokerroin ($\ln(x)$, y)
stat.Resid	Logaritmimalliin liittyvät jäännökset

Tulosmuuttuja	Kuvaus
stat.ResidTrans	Muunnettujen tietojen lineaariseen sovitukseen liittyvät jäännökset
stat.XReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

Local

Katalogi >

Local *Muutt1* [, *Muutt2*] [, *Muutt3*] ...

Määrittää määritetyt *muuttujat* paikallisina muuttujina. Nämä muuttujat ovat olemassa vain funktion sievennyksen aikana, ja ne poistetaan, kun funktion suoritus päättyy.

Huomaa: Paikalliset muuttujat säästävät muistia, koska ne ovat olemassa vain väliaikaisesti. Lisäksi ne eivät häiritse mitään olemassa olevia globaalien muuttujien arvoja. Paikallisia muuttujia on käytettävä **For**-silmukoissa sekä tallennettaessa arvoja väliaikaisesti monirivisessä funktiossa, sillä globaalien muuttujien modifiointeja ei sallita funktiossa.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Define <i>rollcount()</i> =Func	
Local <i>i</i>	
1 → <i>i</i>	
Loop	
If randInt(1,6)=randInt(1,6)	
Goto <i>end</i>	
<i>i</i> +1 → <i>i</i>	
EndLoop	
Lbl <i>end</i>	
Return <i>i</i>	
EndFunc	
	Done
<i>rollcount()</i>	16
<i>rollcount()</i>	3

Lock

Katalogi >

Lock*Muutt1* [, *Muutt2*] [, *Muutt3*] ...
Lock*Muutt*.

Lukitsee määritetyt muuttujat tai muuttujaryhmän. Lukittuja muuttujia ei voi muokata eikä poistaa.

Et voi lukita tai vapauttaa järjestelmän muuttujaa *Ans*, etkä voi lukita järjestelmän muuttujaryhmiä *stat.* tai *tvm.*

<i>a</i> :=65	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a</i> :=75	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a</i> :=75	75
DelVar <i>a</i>	Done

Huomaa: Lukitse-komento (**Lock**) tyhjentää toimintojen Kumoa/Tee uudelleen historian, kun sitä käytetään lukitsemattomiin muuttujiin.

Katso **unLock**, sivu 208, ja **getLockInfo()**, sivu 87.

log()  **painikkeet**

$\log(Laus1[,Laus2]) \Rightarrow \text{lauseke} \log(Lista1[,Laus2]) \Rightarrow lista$

$$\log_{10} (2.) \quad 0.30103$$

$$\log_4 (2.) \quad 0.5$$

$$\log_3 (10) - \log_3 (5) \quad \log_3 (2)$$

Laskee ensimmäisen argumentin kantaluku-*-Laus2*:n logaritmin.

Huomaa: Katso myös **Logaritmimalli**, sivu 2.

Kun kyseessä on lista, laskee elementtien kantaluku-*-Laus2*:n logaritmin.

Jos toinen argumentti jätetään pois, kantelukuna käytetään lukua 10.

Jos kompleksilukumuodon tila on Reaali:

$$\log_{10} (\{-3, 1.2, 5\}) \quad \text{Error: Non-real result}$$

Jos kompleksilukumuodon tila on Suorakulma:

$$\log_{10} (\{-3, 1.2, 5\}) \\ \left\{ \log_{10} (3) + 1.36438 \cdot i, 0.079181, \log_{10} (5) \right\}$$

$\log(neliömatrissi1[,Laus]) \Rightarrow \text{neliömatrissi}$

Laskee matriisin kantaluku-*-Laus*:n logaritmin *neliömatrissi1*:lle. Tämä ei ole sama kuin kunkin elementin kantaluku-*-Laus*-logaritmin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatrissi1:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Jos kantalukuargumentti jätetään pois, kantelukuna käytetään lukua 10.

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

Jos haluat nähdä koko vastauksen, paina ► ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ►.

Laus ▶ **logbase**(*Laus*) ⇒ *lauseke*

Aiheuttaa sen, että syötteenä oleva Lauseke sievennetään lausekkeeksi, joka käyttää kantaluku-*Laus* l:ä.

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>**logbase** (...).

$$\log_3(10) - \log_5(5) \Rightarrow \logbase(5)$$

$$\frac{\log_5\left(\frac{10}{3}\right)}{\log_5(3)}$$

Logistic *X*, *Y*, [*Frekv*] [, *Luokka*, *Sisällyttä*]

Laskee logistisen regressioy = (c/(1+a · e^{-bx})) listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällyttä*-listaa lukuunottamatta.

X ja *Y* ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen *X* ja *Y* esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle *X*- ja *Y*-datalle.

Sisällyttä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: c/(1+a · e ^{-bx})
stat.a, stat.b, stat.c	Regressiokertoimet

Tulosmuuttuja	Kuvaus
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatun <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

LogisticD

Katalogi > 

LogisticD *X*, *Y* [, [*Iteraatiot*] , [*Frekv*] [, *Luokka*, *Sisällytä*]]

Laskee logistisen regressioon $y = (c/(1+a \cdot e^{bx})+d)$ listoista *X* ja *Y* frekvenssillä *Frekv* käyttäen tiettyä *Iteraatioiden* määrää. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-listaa lukuunottamatta.

X ja *Y* ovat riippumattomien ja riippuvien muuttujien listoja.

Iteraatiot on valinnainen arvo, joka määrittää ratkaisun yrityskertojen enimmäismäärän. Mikäli se jätetään pois, käytetään arvoa 64. Suuremmilla arvoilla saadaan tyypillisesti parempi tarkkuus, mutta suoritus aika on pitempi ja päin vastoin.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen *X* ja *Y* esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle *X*- ja *Y* -datalle.

Sisällytä on yhden tai useamman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $c/(1+a \cdot e^{-bx})+d$
stat.a, stat.b, stat.c, stat.d	Regressiokertoimet
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatus <i>X Listan</i> sisältämä datapisteiden lista, jota käytetään todellisesti regressiossa komentojen <i>Frekv</i> , <i>Luokka Lista</i> ja <i>Sisällytä Luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatus <i>Y Listan</i> sisältämä datapisteiden lista, jota käytetään todellisesti regressiossa komentojen <i>Frekv</i> , <i>Luokka Lista</i> ja <i>Sisällytä Luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

Loop (Silmukka)

Loop

Lohko

EndLoop

Suorittaa toistuvasti *Lohkon* sisältämät lausekkeet. Huomaa, että silmukkaa suoritetaan loputtomasti, ellei **Goto**- tai **Exit**-ohjetta suoriteta *Lohkon* sisällä.

Lohko on lausekkeiden sarja, jotka on erotettu toisistaan kaksoispisteellä (:).

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määrittysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Define *rollcount()*=Func

Local *i*

1 → *i*

Loop

If randInt(1,6)=randInt(1,6)

Goto *end*

i+1 → *i*

EndLoop

Lbl *end*

Return *i*

EndFunc

Done

<i>rollcount()</i>	16
<i>rollcount()</i>	3

LU *Matriisi*, *lMatriisi*, *uMatriisi*,
pMatriisi[*Tol*]

Laskee Doolittlen LU (ala-ylä)-
dekomponoinnin reaali- tai
kompleksimatriisista. Alakolmiomatriisi
tallentuu muuttujaan *lMatriisi*,
yläkolmiomatriisi muuttujaan *uMatriisi* ja
permutaatiomatriisi (joka kuvaa laskennan
aikana tehdyt rivien vaihdot) muuttujaan
pMatriisi.

$lMatriisi \cdot uMatriisi = pMatriisi \cdot$
matriisi

Valinnaisesti kaikkia matriisielementtejä
käsitellään nollana, jos niiden itseisarvo on
pienempi kuin *Tol*. Tätä toleranssia
käytetään vain, jos matriisissa on
liukulukusyötteitä eikä se sisällä symbolisia
muuttujia, joille ei ole määritetty arvoa.
Muussa tapauksessa *Tol*-komentoa ei
huomioida.

- Jos käytät painikkeita   tai
Automaattinen tai likimääräinen -tilan
valintaa Approximate (Likimääräinen),
laskut suoritetaan liukulukuaritmetiikalla.
- Jos *Tol* jätetään pois tai sitä ei käytetä,
oletusarvoinen toleranssi lasketaan
seuraavasti:
 $5E-14 \cdot \max(\dim(Matriisi)) \cdot \text{riviNorm}$
(*Matriisi*)

LU-dekomponointialgoritmi käyttää
osittaista rivien vaihtoa (pivoting).

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

LU *m1*, *lower*, *upper*, *perm* Done

<i>lower</i>	$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
--------------	---

<i>upper</i>	$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
--------------	--

<i>perm</i>	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
-------------	---

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
---	--

LU *m1*, *lower*, *upper*, *perm* Done

<i>lower</i>	$\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \\ o & \end{bmatrix}$
--------------	---

<i>upper</i>	$\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$
--------------	--

<i>perm</i>	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
-------------	--

M

mat►list()

mat►list(*Matriisi*) $\Rightarrow$ *lista*

Luo listan, joka on täytetty *Matriisin*
elementeillä. Elementit kopioidaan
Matriisista rivi riviltä.

mat►list($\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$) $\Rightarrow$ {1,2,3}

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
---	--

mat►list(*m1*) $\Rightarrow$ {1,2,3,4,5,6}

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla `mat@>list(...)`.

max()

max(*Laus1*, *Laus2*) $\Rightarrow$ *lauseke*

$\max(2.3, 1.4)$ 2.3

max(*Lista1*, *Lista2*) $\Rightarrow$ *lista*

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

max(*Matriisi1*, *Matriisi2*) $\Rightarrow$ *matriisi*

Laskee kahden argumentin maksimiarvon. Jos argumentteina on kaksi listaa tai matriisia, laskee listan tai matriisin, joka sisältää vastaavien elementtien kunkin parin maksimiarvon.

max(*Lista*) $\Rightarrow$ *lauseke*

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

Laskee *lista*:n maksimielementin.

max(*Matriisi1*) $\Rightarrow$ *matriisi*

$\max\left(\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}\right)$ $\begin{bmatrix} 1 & 0 & 7 \end{bmatrix}$

Luo rivivektorin, joka sisältää *Matriisi1*:n jokaisen sarakkeen maksimielementin.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjästä elementeistä, katso sivu 249.

Huomaa: Katso myös **fMax()** ja **min()**.

mean()

mean(*Lista*[, *frekvLista*]) $\Rightarrow$ *lauseke*

$\text{mean}(\{0.2, 0.1, -0.3, 0.4\})$ 0.26

Laskee *Listan* sisältämien elementtien keskiarvon.

$\text{mean}(\{1, 2, 3\}, \{3, 2, 1\})$ $\frac{5}{3}$

Jokainen *frekvListan* elementti näyttää *Listan* vastaavien elementtien peräkkäisten esiintymien lukumäärän.

mean(*Matriisi1*[, *frekvMatriisi*])
 $\Rightarrow$ *matriisi*

Suorakulmavektorimuodossa:

Luo rivivektorin kaikkien *Matriisi1*:n sarakkeiden keskiarvoista.

mean()Katalogi > 

Jokainen *frekvMatriisin* elementti näyttää *Matriisi I*:n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

mean	$\begin{pmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{pmatrix}$	$\begin{bmatrix} -0.133333 & 0.833333 \end{bmatrix}$
mean	$\begin{pmatrix} \frac{1}{5} & 0 \\ -1 & 3 \\ \frac{2}{5} & \frac{-1}{2} \end{pmatrix}$	$\begin{bmatrix} \frac{-2}{15} & \frac{5}{6} \end{bmatrix}$
mean	$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \cdot \begin{pmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{pmatrix}$	$\begin{bmatrix} \frac{47}{15} & \frac{11}{3} \end{bmatrix}$

median()Katalogi > 

median(Lista[, frekvLista]) ⇒ lauseke

Laskee *Listan* elementtien mediaanin.

Jokainen *frekvListan* elementti näyttää *Listan* vastaavien elementtien peräkkäisten esiintymien lukumäärän.

median(Matriisi I[, frekvMatriisi])
⇒ matriisi

Luo rivivektorin, joka sisältää *Matriisi I*:n sarakkeiden mediaanit.

Jokainen *frekvMatriisin* elementti näyttää *Matriisi I*:n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa:

- Kaikkien listan tai matriisin syötteiden tulee sieventyä luvuiksi.
- Listassa tai matriisissa olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

median	$\begin{pmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{pmatrix}$	$\begin{bmatrix} 0.4 & -0.3 \end{bmatrix}$
--------	---	--

MedMedKatalogi > 

MedMed X,Y[, Frekv] [, Luokka, Sisällyttä]

Laskee mediaani-mediaani-suorany = $(m \cdot x + b)$ listoista X ja Y frekvenssillä *Frekv*. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle X - ja Y -dataalle.

Sisällytä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Mediaani-mediaani-suoran yhtälö: $m \cdot x + b$
stat.m, stat.b	Mallin kertoimet
stat.Resid	Mediaani-mediaani-suoran jäännökset
stat.XReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava <i>frekvenssilista</i>

mid()**Katalogi >**

mid(lähdemerkkijono, Alku[, Count])
 $\Rightarrow$ merkkijono

Laskee *Count*:n merkit merkkijonosta *lähdemerkkijono* aloittaen merkistä numero *Alku*.

Jos *Count* jätetään pois, tai jos se on suurempi kuin *lähdemerkkijono*, laskee kaikki merkit *lähdemerkkijonosta* aloittaen merkistä numero *Alku*.

Count:n on oltava ≥ 0 . Jos *Count* = 0, antaa vastauksena tyhjän merkkijonon.

mid(lähdelista, Alku [, Count]) $\Rightarrow$ lista

Laskee *Count*:n elementit *lähdelistasta* aloittaen elementistä numero *Alku*.

Jos *Count* jätetään pois, tai jos se on suurempi kuin *lähdelista*, laskee kaikki elementit *lähdelistasta* aloittaen elementistä numero *Alku*.

Count:n on oltava ≥ 0 . Jos *Count* = 0, antaa vastauksena tyhjän listan.

mid(lähdeMerkkijonoLista, Alku[, Count])
 $\Rightarrow$ lista

Laskee *Count*:n merkkijonot merkkijonolistasta *lähdeMerkkijonoLista* aloittaen elementistä numero *Alku*.

mid("Hello there",2)	"ello there"
mid("Hello there",7,3)	"the"
mid("Hello there",1,5)	"Hello"
mid("Hello there",1,0)	"{}"

mid({9,8,7,6},3)	{7,6}
mid({9,8,7,6},2,2)	{8,7}
mid({9,8,7,6},1,2)	{9,8}
mid({9,8,7,6},1,0)	{{} }

mid({"A","B","C","D"},2,2)	{"B","C"}
----------------------------	-----------

min()**Katalogi >**

min(Laus1, Laus2) $\Rightarrow$ lauseke

min(Lista1, Lista2) $\Rightarrow$ lista

min(Matriisi1, Matriisi2) $\Rightarrow$ matriisi

Laskee kahden argumentin minimiarvon. Jos argumentteina on kaksi listaa tai matriisia, laskee listan tai matriisin, joka sisältää vastaavien elementtien kunkin parin minimiarvon.

min(Lista) $\Rightarrow$ lauseke

Laskee *Listan* minimielementin.

min(2.3,1.4)	1.4
min({1,2},{-4,3})	{-4,2}

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

min()	Katalogi > 
min (<i>Matriisi I</i>)⇒ <i>matriisi</i>	$\min \begin{pmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{pmatrix} \quad \begin{bmatrix} -4 & -3 & 0.3 \end{bmatrix}$
Luo rivivektorin, joka sisältää <i>Matriisi I</i> :n jokaisen sarakkeen minimielementin.	
Huomaa: Katso myös fMin() ja max() .	

mirr()	Katalogi > 
mirr (<i>tal.arvo,uud.invest.arvo,CF0,CFLista</i> [<i>,CFFrekv</i>])	$\begin{aligned} list1 &:= \{6000, -8000, 2000, -3000\} \\ &\quad \{6000, -8000, 2000, -3000\} \\ list2 &:= \{2, 2, 2, 1\} \quad \{2, 2, 2, 1\} \\ \text{mirr}(4.65, 12, 5000, list1, list2) &\quad 13.41608607 \end{aligned}$
Talouslaskentatoiminto, joka laskee investoinnin modifioidun sisäisen korkokannan.	
<i>tal.arvo</i> on kassavirtamääristä maksettava korkoprosentti.	
<i>uud.invest.arvo</i> on korkokanta, jolla kassavirrat investoidaan uudelleen.	
<i>CF0</i> on alkuperäinen kassavirta aikana 0; arvon on oltava kokonaisluku.	
<i>CFLista</i> on lista kassavirtamääristä ensimmäisen kassavirran CF0 jälkeen.	
<i>CFFrekv</i> on valinnainen lista, jossa jokainen elementti määrittää esiintymisfrekvenssin ryhmitetylle (peräkkäiselle) kassavirtamäärälle, joka on <i>CFListan</i> vastaava elementti. Oletusarvo on 1; jos syötät arvoja, niiden on oltava positiivisia kokonaislukuja < 10,000.	
Huomaa: Katso myös irr() , sivu 98.	

mod()	Katalogi > 
mod (<i>Laus1, Laus2</i>)⇒ <i>lauseke</i>	$\begin{aligned} \text{mod}(7, 0) &\quad 7 \\ \text{mod}(7, 3) &\quad 1 \\ \text{mod}(-7, 3) &\quad 2 \\ \text{mod}(7, -3) &\quad -2 \\ \text{mod}(-7, -3) &\quad -1 \\ \text{mod}(\{12, -14, 16\}, \{9, 7, -5\}) &\quad \{3, 0, -4\} \end{aligned}$
mod (<i>Listal1, Lista2</i>)⇒ <i>lista</i>	
mod (<i>Matriisi1, Matriisi2</i>)⇒ <i>matriisi</i>	
Laskee ensimmäisen argumentin modulo toinen argumentti identtisten yhtälöiden määrittelyn mukaisesti:	

mod()Katalogi >  $\text{mod}(x,0) = x$ $\text{mod}(x,y) = x - y \text{ floor}(x/y)$

Kun toinen argumentti on ei-nolla, vastaus on periodinen tässä argumentissa. Vastaus on joko nolla tai samanmerkkinen kuin toinen argumentti.

Jos argumentteina on kaksi listaa tai matriisia, laskee listan tai matriisin, joka sisältää vastaavien elementtien kunkin parin modulon (jakojäännöksen).

Huomaa: Katso myös **remain()**, sivu 157

mRow()Katalogi > **mRow(Laus, Matriisi1, Indeksi) ⇒ matriisi**

Luo kopion *Matriisi1*:stä, jossa jokainen rivin *Indeksi* elementti *Matriisi1*:ssä on kerrottu lausekkeella *Laus*.

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

mRowAdd()Katalogi > **mRowAdd(Laus, Matriisi1, Indeksi1, Indeksi2) ⇒ matriisi**

Luo kopion *Matriisi1*:stä, jossa jokainen rivin *Indeksi2* elementti *Matriisi1*:ssä on korvattu seuraavasti:

$$\begin{array}{l} \text{mRowAdd}\left(-3, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix} \\ \text{mRowAdd}\left(n, \begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} a & b \\ a \cdot n + c & b \cdot n + d \end{bmatrix} \end{array}$$

Laus · rivi *Indeksi1* + rivi *Indeksi2*

MultRegKatalogi > **MultReg Y, X1[,X2[,X3[,...[,X10]]]**

Laskee listan *Y* moninkertaisen lineaarisen regression listojen *X1*, *X2*, ..., *X10* suhteen. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.b0, stat.b1, ...	Regressiokertoimet
stat.R ²	Moninkertaisen määrityksen kerroin
stat.ŷ Lista	$\hat{y}$ Lista = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Regressioyhtälön jäännökset

MultRegIntervals

MultRegIntervals *Y, X1[,X2[,X3,...*
[,X10]]], XArvoLista[,CTaso]

Laskee ennustetun y:n arvon, tason C ennustevälin yhdelle havainnolle ja tason C luottamusvälin keskiarvovasteelle.

Tulosten yhteenveto tallentuu *stat.results-*muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.ŷ	Pisteen A arvio: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ <i>XArvoListalle</i>
stat.dfError	Virheen vapausasteet
stat.CLower, stat.CUpper	Keskiarvovasteen luottamusväli
stat.ME	Luottamusvälin virhemarginaali
stat.SE	Keskiarvovasteen keskivirhe
stat.LowerPred, stat.UpperrPred	Yhden havainnon ennusteväli

Tulosmuuttuja	Kuvaus
stat.MEPred	Ennustevälin virhemarginaali
stat.SEPred	Ennusteen keskivirhe
stat.bList	Regressiokertoimien lista, {b0,b1,b3,...}
stat.Resid	Regressioyhtälön jäännökset

MultRegTests

Katalogi > 

MultRegTests $Y, X1[,X2[,X3,...[,X10]]]$

Moninkertaisen lineaarisen regression testi laskee lineaarisen regression tietystä datasta ja määrittää kertoimille globaalin F -testin tilastot sekä t -testin tilastot.

Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu sivu 186.)

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjat elementitsivulla sivu 249.

Tulokset

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.F	Globaalin F -testin tilasto
stat.PVal	Globaaliin F -tilastoon liittyvä P-arvo
stat.R ²	Moninkertaisen määrittelyn kerroin
stat.AdjR ²	Moninkertaisen määrittelyn säädetty kerroin
stat.s	Virheen keskihajonta
stat.DW	Durbin-Watsonin tilasto; käytetään määrittäessä, esiintyykö mallissa ensimmäisen asteen automaattista korrelaatiota
stat.dfReg	Regression vapausasteet
stat.SSReg	Regression neliöiden summa
stat.MSReg	Regression keskineliö
stat.dfError	Virheen vapausasteet
stat.SSError	Virheen neliöiden summa

Tulosmuuttuja	Kuvaus
stat.MSError	Virheen keskineliö
stat.bList	{b0,b1,...} Kertoimien lista
stat.tList	t-tilastojen lista, yksi kullekin bListan sisältämälle kertoimelle
stat.PList	P-arvojen lista kullekin t-tilastolle
stat.SEList	Keskivirheiden lista bListan sisältämille kertoimille
stat. $\hat{y}$ Lista	$\hat{y}$ Lista = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Regressioyhtälön jäännökset
stat.sResid	Standardoidut jäännökset; saadaan jakamalla jäännös keskihajonnalla
stat.CookDist	Cookin etäisyys; jäännökseen ja tuottosuhteeseen perustuvan havainnon vaikutus
stat.Leverage	Miten kaukana riippumattoman muuttujan arvot ovat niiden keskiarvoista

N

nand (ei-ja)	ctrl	=	näppäimet
<i>BooleanLaus1</i> nand <i>BooleanLaus2</i> antaa vastauksena <i>Boolean</i> lausekkeen <i>BooleanList1</i>	$x \geq 3$ and $x \geq 4$		$x \geq 4$
nand <i>BooleanList2</i> antaa vastauksena <i>Boolean</i> listan <i>BooleanMatriisi1</i>	$x \geq 3$ nand $x \geq 4$		$x < 4$
nand <i>BooleanMatriisi2</i> antaa vastauksena <i>Boolean</i> matriisin			
Antaa vastauksena loogisen and operaation negaation kahdesta argumentista. Antaa vastauksena totuusarvon tosi, epätosi tai yhtälön sievennetyn muodon.			
Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.			
<i>Kokonaisluku1</i> nand	3 and 4		0
<i>Kokonaisluku2</i> $\Rightarrow$ <i>kokonaisluku</i>	3 nand 4		-1
	{1,2,3} and {3,2,1}		{1,2,1}
	{1,2,3} nand {3,2,1}		{-2,-3,-2}

Vertaa kahta reaalikokonaislukua bitti bitiltä **nand**-operaation avulla. Sisäisesti kumpikin kokonaisluku muunnetaan etumerkilliseksi, 64 bitin binaariluvuksi. Kun vastaavia bittejä verrataan, tulos on 0, jos kumpikin bitti on 1. Muussa tapauksessa tulos on 1. Laskettu arvo edustaa bittituloksia ja se näkyy kantalukutilan mukaisesti.

Kokonaisluvut voi syöttää minkä tahansa luvun kantalukuna. Binaarisen syötteen edelle tulee merkitä etumerkki 0b ja heksadesimaalisen syötteen edelle 0h. Jos etumerkkiä ei ole, kokonaislukuja käsitellään desimaalilukuina (kantaluku 10).

nCr()

Katalogi > 

nCr(Laus1, Laus2)⇒lauseke

Kokonaisluvulle *Laus1* ja *Laus2* sekä *Laus1* $\geq$ *Laus2* $\geq$ 0, **nCr()** on *Laus1*:n asioiden kombinaatioiden lukumäärä, joita otetaan *Laus2*:n verran kerrallaan. (Tästä käytetään myös nimitystä binomikerroin.) Kumpikin argumentti voi olla kokonaisluku tai symbolinen lauseke.

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

nCr(Laus, 0)⇒1

nCr(Laus, negKokonaisluku)⇒0

nCr(Laus, posKokonaisluku)⇒ Laus · (Laus-1) ... (Laus-posKokonaisluku+1)/ posKokonaisluku!

nCr(Laus, eiKokonaisluku)⇒lauseke!/((Laus-eiKokonaisluku)! · eiKokonaisluku!)

nCr(Lista1, Lista2)⇒lista

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

Laskee listan kombinaatioista kahden listan sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia listoja.

nCr()Katalogi > **nCr**(*Matriisi1*, *Matriisi2*) $\Rightarrow$ *matriisi*

$nCr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$
--	--

Laskee matriisin kombinaatioista kahden matriisin sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia matriiseja.

nDerivative()Katalogi > **nDerivative**(*Laus1*, *Muutt*=*Arvo*[*Aste*])
 $\Rightarrow$ *arvo*

$nDerivative(x , x=1)$	1
-------------------------	---

$nDerivative(x , x) _{x=0}$	undef
------------------------------	-------

nDerivative(*Laus1*, *Muutt*[*Aste*]) |
Muutt=*Arvo* $\Rightarrow$ *arvo*

$nDerivative(\sqrt{x-1}, x) _{x=1}$	undef
-------------------------------------	-------

Laskee numeerisen derivaatan käyttäen automaattisia derivointimenetelmiä.

Kun *Arvo* määritetään, se ohittaa mahdolliset aikaisemmat muuttujamäärytykset tai mahdolliset muuttujan nykyiset "|" -sijoitukset.

Derivaatan Asteen on oltava **1** tai **2**.

newList()Katalogi > **newList**(*numElementit*) $\Rightarrow$ *lista*

$newList(4)$	$\{0,0,0,0\}$
--------------	---------------

Antaa tuloksena listan, jonka koko on *numElementit*. Jokainen elementti on nolla.

newMat()Katalogi > **newMat**(*numRivit*, *numSarakkeet*)
 $\Rightarrow$ *matriisi*

$newMat(2,3)$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
---------------	--

Antaa tuloksena nollamatriisin, jonka koko on *numRivit* ja *numSarakkeet*.

nfMax()Katalogi > 

nfMax(*Laus*, *Muutt*) $\Rightarrow$ *arvo*
nfMax(*Laus*, *Muutt*, *alaraja*) $\Rightarrow$ *arvo*
nfMax(*Laus*, *Muutt*, *alaraja*, *yläraja*)
 $\Rightarrow$ *arvo*
nfMax(*Laus*, *Muutt*) *alaraja* $\leq$ *Muutt*
 $\leq$ *yläraja* $\Rightarrow$ *arvo*

$\text{nfMax}\left(x^2 - 2 \cdot x - 1, x\right)$	-1.
$\text{nfMax}\left(0.5 \cdot x^3 - x - 2, x, -5, 5\right)$	5.

Laskee muuttujan *Muutt* numeerisen ehdotusarvon, jossa lausekkeen *Laus* paikallinen maksimi esiintyy.

Jos ilmoitat *alarajan* ja *ylärajan*, funktio etsii suljetulta väliltä [*alaraja*,*yläraja*] paikallista maksimia.

Huomaa: Katso myös **fMax()** ja **d()**.

nfMin()Katalogi > 

nfMin(*Laus*, *Muutt*) $\Rightarrow$ *arvo*
nfMin(*Laus*, *Muutt*, *alaraja*) $\Rightarrow$ *arvo*
nfMin(*Laus*, *Muutt*, *alaraja*, *yläraja*)
 $\Rightarrow$ *arvo*
nfMin(*Laus*, *Muutt*) *alaraja* $\leq$ *Muutt*
 $\leq$ *yläraja* $\Rightarrow$ *arvo*

$\text{nfMin}\left(x^2 + 2 \cdot x + 5, x\right)$	-1.
$\text{nfMin}\left(0.5 \cdot x^3 - x - 2, x, -5, 5\right)$	-5.

Laskee muuttujan *Muutt* numeerisen ehdotusarvon, jossa lausekkeen *Laus* paikallinen minimi esiintyy.

Jos ilmoitat *alarajan* ja *ylärajan*, funktio etsii suljetulta väliltä [*alaraja*,*yläraja*] paikallista minimia.

Huomaa: Katso myös **fMin()** ja **d()**.

nInt()Katalogi > 

nInt(*LausI*, *Muutt*, *Ala*, *Ylä*) $\Rightarrow$ *lauseke*

$\text{nInt}\left(e^{-x^2}, x, -1, 1\right)$	1.49365
--	---------

Jos integroitava funktio *LausI* ei sisällä muita muuttujia kuin *Muutt*, ja jos *Ala* ja *Ylä* ovat vakioita, positiivinen ∞ tai negatiivinen ∞ , tällöin **nInt()** laskee likiarvon lausekkeesta $\int (LausI, Var, Ala, Ylä)$. Tämä likiarvo on integrandin välillä *Ala* $<$ *Muutt* $<$ *Ylä* olevien joidenkin otosarvojen painotettu keskiarvo.

nInt()	Katalogi >
Tavoitteena on kuusi merkitsevää numeroa. Adaptiivinen algoritmi päättyy, kun näyttää todennäköiseltä, että tavoite on saavutettu, tai kun näyttää epätodennäköiseltä, että lisäotokset tuottaisivat merkittävää parannusta.	$\text{nInt}\left(\cos(x), x, -\pi, \pi+1.\text{E}-12\right) \quad -1.04144\text{E}-12$ $\int_{-\pi}^{\pi+10^{-12}} \cos(x) dx \quad -\sin\left(\frac{1}{1000000000000}\right)$
Näkyviin tulee viesti (Questionable accuracy (Kyseenalainen tarkkuus)), kun näyttää siltä, että tavoitetta ei ole saavutettu.	
Sijoita nInt() -komentoa sisäkkäin, jos haluat suorittaa moninkertaisen numeerisen integroinnin. Integroinnin raja-arvot voivat riippua niiden ulkopuolella olevista integrointimuuttujista.	$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x\right), x, 0, 1\right) \quad 3.30423$
Huomaa: Katso myös f() , sivu 221.	

nom()	Katalogi >
nom (<i>efektiivinenKorko</i> , <i>CpY</i>) $\Rightarrow$ <i>arvo</i>	$\text{nom}(5.90398, 12) \quad 5.75$
Talouselääntätoiminto, joka muuntaa efektiivisen vuosikoron <i>efektiivinenKorko</i> nimelliskoroksi, kun <i>CpY</i> määritetään korkojaksojen lukumääräksi vuodessa.	
<i>efektiivinenKorko</i> on oltava reaali-luku, ja <i>CpY</i> :n on oltava reaali-luku > 0.	
Huomaa: Katso myös eff() , sivu 61.	

nor (eikä)	näppäimet
<i>BooleanLaus1</i> nor <i>BooleanLaus2</i> antaa vastauksena <i>Boolean</i>	$x \geq 3 \text{ or } x \geq 4 \quad x \geq 3$
<i>lausekkeenBooleanList1</i>	$x \geq 3 \text{ nor } x \geq 4 \quad x < 3$
norBooleanList2 antaa vastauksena <i>Boolean</i>	
<i>listanBooleanMatriisi1</i>	
norBooleanMatriisi2 antaa vastauksena <i>Boolean matriisin</i>	
Antaa vastauksena loogisen or operaation negaation kahdesta argumentista. Antaa vastauksena totuusarvon tosi, epätosi tai yhtälön sievennetyn muodon.	

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Kokonaisluku1

nor*Kokonaisluku2*⇒*kokonaisluku*

Vertaa kahta reaalikokonaislukua bitti bitiltä **nor**-operaation avulla. Sisäisesti kumpikin kokonaisluku muunnetaan etumerkilliseksi, 64 bitin binaariluvuksi. Kun vastaavia bittejä verrataan, tulos on 1, jos kumpikin bitti on 1. Muussa tapauksessa tulos on 0. Laskettu arvo edustaa bittituloksia ja se näkyy kantelukutilan mukaisesti.

Kokonaisluvut voi syöttää minkä tahansa luvun kantelukuna. Binaarisen syötteen edelle tulee merkitä etumerkki 0b ja heksadesimaalisen syötteen edelle 0h. Jos etumerkkiä ei ole, kokonaislukuja käsitellään desimaalilukuina (kantaluksi 10).

3 or 4	7
3 nor 4	-8
$\{1,2,3\}$ or $\{3,2,1\}$	$\{3,2,3\}$
$\{1,2,3\}$ nor $\{3,2,1\}$	$\{-4,-3,-4\}$

norm()

Katalogi >

norm(*Matriisi*)⇒*lauseke*

norm(*Vektori*)⇒*lauseke*

Laskee Frobeniusin normin.

$\text{norm}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$\sqrt{a^2+b^2+c^2+d^2}$
$\text{norm}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	$\sqrt{30}$
$\text{norm}\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right)$	$\sqrt{5}$
$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	$\sqrt{5}$

normalLine()

Katalogi >

normalLine(*Lausl*,*Muutt*,*Piste*)⇒*lauseke*

normalLine(*Lausl*,*Muutt*=*Piste*)⇒*lauseke*

Määrittää normaalisuoran *Lausl*:n kuvaamasta käyrästä pisteessä, joka on määritetty kohtaan *Muutt*=*Piste*.

$\text{normalLine}(x^2,x,1)$	$\frac{3}{2} \cdot \frac{x}{2}$
$\text{normalLine}((x-3)^2-4,x,3)$	$x=3$
$\text{normalLine}\left(\frac{1}{x^3},x=0\right)$	0
$\text{normalLine}(\sqrt{ x },x=0)$	undef

normalLine()

Katalogi > 

Varmista, että riippumatonta muuttujaa ei ole määritetty. Esimerkiksi, jos $f1(x):=5$ ja $x:=3$, tällöin **normalLine**($f1(x),x,2$) antaa vastauksen "epätosi".

normCdf()

Katalogi > 

normCdf(*alaraja*,*yläraja*[, μ],[σ]]) $\Rightarrow$ *luku*, jos *yläraja* ja *alaraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

Laskee normaalijakauman todennäköisyyden *alarajan* ja *ylärajan* välillä määritetyille μ :lle (oletus=0) ja σ :lle (oletus=1).

Kun $P(X \leq \textit{yläraja})$, aseta *alaraja* = $-\infty$.

normPdf()

Katalogi > 

normPdf(*XArvo*[, μ],[σ]]) $\Rightarrow$ *luku*, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

Laskee normaalijakauman pistetodennäköisyysfunktion määritetyssä *XArvossa* määritetyille μ :lle ja σ :lle.

not (ei)

Katalogi > 

not *BooleanLaus* $\Rightarrow$ *Boolean lauseke*

Määrittää totuusarvon tosi, epätosi tai argumentin sievennetyn muodon.

not *Kokonaisluku* $\Rightarrow$ *kokonaisluku*

Laskee reaalikokonaisluvun ykkösen komplementin. Sisäisesti *Kokonaisluku*1 muunnetaan etumerkilliseksi, 64-bittiseksi binaariluvuksi. Jokaisen bitin arvo vaihtuu (0:sta tulee 1 ja päin vastoin) ykkösen komplementille. Tulokset näytetään kantalukutilan mukaisesti.

$\text{not}(2 \geq 3)$	true
$\text{not}(x < 2)$	$x \geq 2$
not not innocent	<i>innocent</i>

Heksadesimaalisessa kantalukutilassa:

not 0h7AC36	0hFFFFFFFF853C9
-------------	-----------------

Binaarisessa kantalukutilassa:

0b100101►Base10	37
not 0b100101	
0b11111111111111111111111111111111►	
not 0b100101►Base10	-38

Voit syöttää kokonaisluvun minkä tahansa luvun kantalukuna. Binaarisen syötteen edelle tulee merkitä etumerkki 0b ja heksadesimaalisen syötteen edelle 0h. Ilman etuliitettä kokonaislukua käsitellään desimaalilukuna (kantaluku 10).

Jos syötät desimaalikokonaisluvun, joka on etumerkillisen, 64 bitin binaarimuodon lukualueen ulkopuolella, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle. Lisätietoja, katso ►Base2, sivu 18.

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ►.

Huomaa: Binaarisessa syötteessä voi olla korkeintaan 64 numeroa (etuliitettä 0b ei lasketa). Heksadesimaalisessa syötteessä voi olla korkeintaan 16 numeroa.

nPr()

Katalogi >

$\text{nPr}(\text{Laus1}, \text{Laus2}) \Rightarrow \text{lauseke}$

Kokonaisluvulle *Laus1* ja *Laus2* sekä *Laus1* $\geq$ *Laus2* ≥ 0 , **nPr()** on *Laus1*:n asioiden permutaatioiden lukumäärä, joita otetaan *Laus2*:n verran kerrallaan. Kumpikin argumentti voi olla kokonaisluku tai symbolinen lauseke.

$\text{nPr}(z, 3)$	$z \cdot (z-2) \cdot (z-1)$
$\text{Ans} z=5$	60
$\text{nPr}(z, -3)$	$\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$
$\text{nPr}(z, c)$	$\frac{z!}{(z-c)!}$
$\text{Ans} \cdot \text{nPr}(z-c, -c)$	1

$\text{nPr}(\text{Laus}, 0) \Rightarrow 1$

$\text{nPr}(\text{Laus}, \text{negKokonaisluku}) \Rightarrow 1 / ((\text{Laus}+1) \cdot (\text{Laus}+2) \dots (\text{lauseke} - \text{negKokonaisluku}))$

$\text{nPr}(\text{Laus}, \text{posKokonaisluku}) \Rightarrow \text{Laus} \cdot (\text{Laus}-1) \dots (\text{Laus}-\text{posKokonaisluku}+1)$

$\text{nPr}(\text{Laus}, \text{eiKokonaisluku}) \Rightarrow \text{Laus!} / (\text{Laus}-\text{eiKokonaisluku})!$

$\text{nPr}(\text{List1}, \text{List2}) \Rightarrow \text{lista}$

Luo listan permutaatioista kahden listan sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia listoja.

$\text{nPr}(\{5,4,3\}, \{2,4,2\})$	$\{20,24,6\}$
------------------------------------	---------------

$\text{nPr}(\text{Matriisi1}, \text{Matriisi2}) \Rightarrow \text{matriisi}$

$\text{nPr}\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$
---	---

Luo matriisin permutaatioista kahden matriisin sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia matriiseja.

npv()

Katalogi >

npv(*Korkoprosentti*,*CFO*,*CFLista* [*,CFFrekv*])

Talouslaskentatoiminto, joka laskee nettonykyarvon; tulevien ja poistuvien kassavirtojen nykyisten arvojen summan. Jos npv:n tulos on positiivinen, investointi on kannattava.

Korkoprosentti on prosentti, jolla kassavirtoja (rahan kustannusta) vähennetään yhden jakson aikana.

CFO on alkuperäinen kassavirta aikana 0; arvon on oltava kokonaisluku.

CFLista on lista kassavirtamääristä ensimmäisen kassavirran *CFO* jälkeen.

CFFrekv on lista, jossa jokainen elementti määrittää esiintymisfrekvenssin ryhmitetyille (peräkkäiselle) kassavirtamäärälle, joka on *CFListan* vastaava elementti. Oletusarvo on 1; jos syötät arvoja, niiden on oltava positiivisia kokonaislukuja < 10,000.

$list1 := \{ 6000, -8000, 2000, -3000 \}$	
$\{ 6000, -8000, 2000, -3000 \}$	
$list2 := \{ 2, 2, 2, 1 \}$	$\{ 2, 2, 2, 1 \}$
$npv(10, 5000, list1, list2)$	4769.91

nSolve()

Katalogi >

nSolve(*Yhtälö*,*Muutt*[=*Arvaus*]) $\Rightarrow$ luku tai virhe_merkkijono

nSolve(*Yhtälö*,*Muutt*[=*Arvaus*],*alaraja*)
 $\Rightarrow$ luku tai virhe_merkkijono

nSolve(*Yhtälö*,*Muutt* [*=Arvaus*],*alaraja*,*yläraja*) $\Rightarrow$ luku tai virhe_merkkijono

nSolve(*Yhtälö*,*Muutt*[=*Arvaus*]) | *alaraja* $\leq$ *Muutt* $\leq$ *yläraja*

$nSolve(x^2 + 5 \cdot x - 25 = 9, x)$	3.84429
$nSolve(x^2 = 4, x = -1)$	-2.
$nSolve(x^2 = 4, x = 1)$	2.

Huomaa: Jos ratkaisuja on useita, voit yrittää löytää tietyn ratkaisun käyttämällä apuna arvausta.

$\Rightarrow$ luku tai virhe_merkkijono

Etsii iteratiivisesti yhtä likimääräistä numeerista ratkaisua *Yhtälön* yhdelle muuttujalle. Määritä muuttuja seuraavasti:

muuttuja

– tai –

muuttuja = *reaaliluku*

Esimerkiksi x kelpaa ja samoin $x=3$.

nSolve() on usein paljon nopeampi kuin **solve()** tai **zeros()**, erityisesti jos operaattoria " $|$ " käytetään haun rajaamisessa pienelle välille, joka sisältää täsmälleen yhden yksinkertaisen ratkaisun.

nSolve() yrittää määrittää joko yhden pisteen, jossa jäännös on nolla, tai kaksi toisiaan suhteellisen lähellä olevaa pistettä, jossa jäännöksen etumerkki on vastakkainen ja jäännöksen magnitudi ei ole liian suuri. Jos funktio ei pysty määrittämään tätä kohtuullisella otospisteiden määrällä, se antaa vastauksena merkkijonon "no solution found" (yhtään ratkaisua ei löydy).

Huomaa: Katso myös **cSolve()**, **cZeros()**, **solve()**, ja **zeros()**.

$\text{nSolve}(x^2+5\cdot x-25=9,x) x<0$	-8.84429
$\text{nSolve}\left(\frac{(1+r)^{24}-1}{r}=26,r\right) r>0 \text{ and } r<0.25$	0.006886
$\text{nSolve}(x^2=-1,x)$	"No solution found"

O

OneVar

OneVar [**1**],*X1*,[*Frekv*],[*Luokka*,*Sisällytä*]]

OneVar [*n*],*X1*,*X2*[*X3*[...[,*X20*]]]

Laskee yhden muuttujan tilaston enintään 20 listasta. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-listaa lukuunottamatta.

X-argumentit ovat datalistoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan X :n arvon esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on numeeristen luokkakoodien lista vastaaville X :n arvoille.

Sisällytä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Tyhjä elementti jossakin listassa X , *Frekv* tai *Luokka* saa aikaan, että kaikkien listojen vastaava elementti on tyhjä. Tyhjä elementti jossakin listassa $X1$ - $X20$ saa aikaan, että kaikkien listojen vastaava elementti on tyhjä. Lisätietoja tyhjästä elementeistä, katso sivu 249.

Tulosmuuttuja	Kuvaus
stat. $\bar{X}$	x :n arvojen keskiarvo
stat. Σx	x :n arvojen summa
stat. Σx^2	x^2 :n arvojen summa
stat.sx	x :n otoksen keskihajonta
stat. x	x :n perusjoukon keskihajonta
stat.n	Datapisteiden lukumäärä
stat.MinX	x :n arvojen minimi
stat. $Q_1 X$	x :n ensimmäinen neljännes
stat.MedianX	x :n mediaani
stat. $Q_3 X$	x :n kolmas neljännes
stat.MaxX	x :n arvojen maksimi
stat.SSX	x :n keskiarvon poikkeamien neliöiden summa

BooleanLaus1 or *BooleanLaus2* antaa vastauksena *Boolean* lausekkeen *BooleanList1* or *BooleanList2* antaa vastauksena *Boolean* listan *BooleanMatriisi1* or *BooleanMatriisi2* antaa vastauksena *Boolean* matriisin

Määrittää totuusarvon tosi tai epätosi tai antaa vastauksena sievennetyn muodon alkuperäisestä syötteestä.

Antaa vastauksen tosi, jos jompikumpi tai molemmat lausekkeet ovat tosia. Antaa vastauksen epätosi, jos kumpikin lauseke on epätosi.

Huomaa: Katso xor.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määrittysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Kokonaisluku1

or *Kokonaisluku2* ⇒ *kokonaisluku*

Vertaa kahta reaalikokonaislukua bitti bitiltä or-operaation avulla. Sisäisesti kumpikin kokonaisluku muunnetaan etumerkilliseksi, 64 bitin binaariluvuksi. Kun vastaavia bittejä verrataan, tulos on 1, jos jompikumpi bitti on 1; tulos on 0 vain, jos kumpikin bitti on 0. Laskettu arvo edustaa bittituloksia, ja se näkyy kantalukutilan mukaisesti.

Kokonaisluvut voi syöttää minkä tahansa luvun kantalukuna. Binaarisen syötteen edelle tulee merkitä etumerkki 0b ja heksadesimaalisen syötteen edelle 0h. Jos etumerkkiä ei ole, kokonaislukuja käsitellään desimaalilukuina (kantaluku 10).

Jos syötät desimaalikokonaisluvun, joka on liian suuri etumerkilliselle, 64 bitin binaarimuodolle, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle.

$x \geq 3$ or $x \geq 4$	$x \geq 3$
--------------------------	------------

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
--------	-----------------------------------

Heksadesimaalisessa kantalukutilassa:

0h7AC36 or 0h3D5F	0h7BD7F
-------------------	---------

Tärkeää: Nolla, ei O-kirjain.

Binaarisessa kantalukutilassa:

0b100101 or 0b100	0b100101
-------------------	----------

Huomaa: Binaarisessa syötteessä voi olla korkeintaan 64 numeroa (etuliitettä 0b ei lasketa). Heksadesimaalisessa syötteessä voi olla korkeintaan 16 numeroa.

or (tai)Katalogi > 

Jos syötät desimaalikonaisluvun, joka on etumerkillisen, 64 bitin binaarimuodon lukualueen ulkopuolella, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle. Lisätietoja, katso ▶**Base2**, sivu 18.

Huomaa: Katso **xor**.

ord()Katalogi > 

ord(*Merkkijono*)⇒*kokonaisluku*

ord(*Listal*)⇒*lista*

Määrittää merkkijonon *Merkkijono* ensimmäisen merkin numerokoodin tai luo listan jokaisen listaelementin ensimmäisistä merkeistä.

ord ("hello")	104
char (104)	"h"
ord (char (24))	24
ord ({"alpha", "beta"})	{ 97, 98 }

P**P ▶ Rx()**Katalogi > 

P ▶ Rx(*rLaus*, *θLaus*)⇒*lauseke*

P ▶ Rx(*rLista*, *θLista*)⇒*lista*

P ▶ Rx(*rMatriisi*, *θMatriisi*)⇒*matriisi*

Määrittää paria
(*r*, *θ*) vastaavan x-koordinaatin.

Huomaa: Argumentti *θ* tulkitaan joko aste-, graadi- tai radiaanikulmaksi valittuna olevan kulmatilan mukaisesti. Jos argumentti on lauseke, voit ohittaa kulmatila-asetuksen väliaikaisesti käyttämällä merkintää °, ° tai °.

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla **P@>Rx (...)**.

Radiaanikulmatilassa:

P ▶ Rx (<i>r</i> , <i>θ</i>)	$\cos(\theta) \cdot r$
P ▶ Rx (4,60°)	2
P ▶ Rx ({-3,10,1.3}, { $\frac{\pi}{3}, \frac{\pi}{4}, 0$ })	$\left\{ \frac{-3}{2}, 5\sqrt{2}, 1.3 \right\}$

P ▶ Ry()Katalogi > 

P ▶ Ry(*rLaus*, *θLaus*)⇒*lauseke*

Radiaanikulmatilassa:

P► Ry(*rLista*, *θLista*)⇒*lista*P► Ry(*rMatriisi*, *θMatriisi*)⇒*matriisi*

Määrittää paria (*r*, *θ*) vastaavan y-koordinaatin.

Huomaa: Argumentti *θ* tulkitaan joko aste-, radiaani- tai graadikulmaksi valittuna olevan kulmatilan mukaisesti. Jos argumentti on lauseke, voit ohittaa kulmatila-asetuksen väliaikaisesti käyttämällä merkintää °, °G tai °r.

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla P@>Ry (...).

$$P\blacktriangleright Ry(r, \theta) \quad \sin(\theta) \cdot r$$

$$P\blacktriangleright Ry(4, 60^\circ) \quad 2 \cdot \sqrt{3}$$

$$P\blacktriangleright Ry\left(\{-3, 10, 1.3\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}\right) \quad \left\{\frac{-3 \cdot \sqrt{3}}{2}, -5 \cdot \sqrt{2}, 0\right\}$$

PassErr

PassErr

Ohittaa virheen siirtyen seuraavalle tasolle.

Jos järjestelmän muuttuja *errCode* on nolla, **PassErr** ei tee mitään.

Else-lauseessa lohossa **Try...Else...EndTry** tulee käyttää komentoa **ClrErr** tai **PassErr**. Jos virhe on tarkoitus käsitellä tai jättää huomiotta, käytä komentoa **ClrErr**. Jos et tiedä, mitä tehdä virheen suhteen, lähetä se seuraavaan virheenkäsittelijään käyttämällä komentoa **PassErr**. Jos odottavia **Try...Else...EndTry**-virheenkäsittelijöitä ei ole enää, virheen valintaikkuna tulee näkyviin normaalisti.

Huomaa: Katso myös **ClrErr**, sivu 26, ja **Try**, sivu 201.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Esimerkki **PassErr**-komennosta, katso esimerkki 2 **Try**-komennon kohdalla, sivu 201.

piecewise()

[Katalogi >](#) 

piecewise(*Laus1* [, *Ehto1* [, *Laus2* [, *Ehto2* [, ...]]]])

Laatii määritelmät paloittain määritellylle funktiolle listan muodossa. Voit luoda paloittain määriteltujen funktioiden määrittäksiä myös mallin avulla.

Huomaa: Katso myös **Paloittain määritellyn funktion malli**, sivu 3.

Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$	Done
$p(1)$	1
$p(-1)$	undef

poissCdf()

[Katalogi >](#) 

poissCdf(λ , *alaraja*, *yläraja*) $\Rightarrow$ luku, jos *alaraja* ja *yläraja* ovat lukuja, lista, jos *alaraja* ja *yläraja* ovat listoja

poissCdf(λ , *yläraja*) (kun $P(0 \leq X \leq \text{yläraja}) \Rightarrow$ luku, jos *yläraja* on luku, lista, jos *yläraja* on lista

Laskee kumulatiivisen todennäköisyyden diskreetille Poissonin jakaumalle, jolla on määritetty keskiarvo λ .

Kun $P(X \leq \text{yläraja})$, aseta *alaraja*=0

poissPdf()

[Katalogi >](#) 

poissPdf(λ , *XArvo*) $\Rightarrow$ luku, jos *XArvo* on luku, lista, jos *XArvo* on lista

Laskee todennäköisyyden diskreetille Poissonin jakaumalle, jolla on määritetty keskiarvo λ .

► Polar (►Polaarinen)

[Katalogi >](#) 

Vektori ► **Polar**

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla **@>Polar**.

Näyttää *vektorin* polaarisessa muodossa [*r* $\angle$ θ]. Vektorin on oltava kooltaan 2, ja se voi olla rivi tai sarake.

[1 3] ► Polar	[3.16228 $\angle$ 1.24905]
[x y] ► Polar	$\left[\sqrt{x^2 + y^2} \angle \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right) \right]$

Huomaa: ► **Polar** on näyttömuodon ohje, ei muunnosfunktio. Voit käyttää komentoa ainoastaan syöterivin lopussa, eikä se päivitä *ans*:n arvoa.

Huomaa: Katso myös ► **Rect**, sivu 154.

kompleksiArvo ► **Polaarinen**

Näyttää *kompleksiVektorin* polaarisisessä muodossa.

- Astekulmatilassa vastauksena on $(r \angle \theta)$.
- Radiaanikulmatilassa vastauksena on $re^{i\theta}$.

kompleksiArvolla voi olla mikä tahansa kompleksilukumuoto. Syöte $re^{i\theta}$ aiheuttaa kuitenkin virheen astekulmatilassa.

Huomaa: Polaarisisessä syötteessä $(r \angle \theta)$ on käytettävä sulkeita.

Radiaanikulmatilassa:

$$(3+4 \cdot i) \text{►Polar} \quad e^{i \cdot \left(\frac{\pi}{2} - \tan^{-1} \left(\frac{3}{4} \right) \right)} \cdot 5$$

$$\left(4 \angle \frac{\pi}{3} \right) \text{►Polar} \quad e^{\frac{i \cdot \pi}{3}} \cdot 4$$

Graadikulmatilassa:

$$(4 \cdot i) \text{►Polar} \quad (4 \angle 100.)$$

Astekulmatilassa:

$$(3+4 \cdot i) \text{►Polar} \quad \left(5 \angle 90 - \tan^{-1} \left(\frac{3}{4} \right) \right)$$

polyCoeffs()

polyCoeffs(*Poly* [,*Muutt*])⇒*lista*

Laatii listan polynomin *Poly* kertoimista muuttujan *Muutt* suhteen.

Poly on oltava polynomilauseke muuttujassa *Muutt*. Suositeltavaa on, ettet jätä muuttujaa *Muutt* pois, ellei *Poly* ole lauseke yhdessä muuttujassa.

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

$$\text{polyCoeffs}((x-1)^2 \cdot (x+2)^3) \quad \{1, 4, 1, -10, -4, 8\}$$

Laventaa polynomin ja valitsee *x*:n poisjätetyille muuttujalle *Muutt*.

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

polyDegree()

polyDegree(Poly [,Muutt]) ⇒ arvo

Määrittää polynomilausekkeen *Poly* asteen muuttujan *Muutt* suhteen. Jos jätät muuttujan *Muutt* pois, **polyDegree()**-funktio valitsee oletusarvon polynomien *Poly* sisältämistä muuttujista.

Poly on oltava polynomilauseke muuttujassa *Muutt*. Suositeltavaa on, ettet jätä muuttujaa *Muutt* pois, ellei *Poly* ole lauseke yhdessä muuttujassa.

$\text{polyDegree}(5)$	0
$\text{polyDegree}(\ln(2) + \pi, x)$	0
Vakiopolynomit	
$\text{polyDegree}(4 \cdot x^2 - 3 \cdot x + 2, x)$	2
$\text{polyDegree}((x-1)^2 \cdot (x+2)^3)$	5
$\text{polyDegree}\left(\left(x+y^2+z^3\right)^2, x\right)$	2
$\text{polyDegree}\left(\left(x+y^2+z^3\right)^2, y\right)$	4
$\text{polyDegree}\left(\left(x-1\right)^{10000}, x\right)$	10000

Asteesta voidaan ottaa juuri, vaikka kertoimista ei voida. Tämä johtuu siitä, että asteesta voidaan ottaa juuri laventamatta polynomia.

polyEval()

polyEval(Lista1, Laus1) ⇒ lauseke
polyEval(Lista1, Lista2) ⇒ lauseke

$\text{polyEval}(\{a, b, c\}, x)$	$a \cdot x^2 + b \cdot x + c$
$\text{polyEval}(\{1, 2, 3, 4\}, 2)$	26
$\text{polyEval}(\{1, 2, 3, 4\}, \{2, -7\})$	$\{26, -262\}$

polyEval()Katalogi > 

Tulkitsee ensimmäisen argumentin laskeva-asteisen polynomin kertoimeksi ja antaa vastauksena polynomin, josta on laskettu toisen argumentin arvo.

polyGcd()Katalogi > 

polyGcd(Laus1,Laus2)⇒*lauseke*

Laskee kahden argumentin suurimman yhteisen jakajan.

Laus1:n ja *Laus2*:n on oltava polynomilausekkeita.

Lista-, matriisi- ja Boolean argumentteja ei sallita.

$\text{polyGcd}(100,30)$	10
$\text{polyGcd}(x^2-1,x-1)$	$x-1$
$\text{polyGcd}(x^3-6\cdot x^2+11\cdot x-6,x^2-6\cdot x+8)$	$x-2$

polyQuotient()Katalogi > 

polyQuotient(Poly1,Poly2 [,Muutt])
⇒*lauseke*

Laskee polynomin *Poly1* osamäärän jaettuna polynomilla *Poly2* määritetyn muuttujan *Muutt* suhteen.

Poly1:n ja *Poly2*:n on oltava polynomilausekkeita muuttujassa *Muutt*. Suositeltavaa on, ettet jätä muuttujaa *Muutt* pois, elleivät *Poly1* ja *Poly2* ole lausekkeita yhdessä ja samassa muuttujassa.

$\text{polyQuotient}(x-1,x-3)$	1
$\text{polyQuotient}(x-1,x^2-1)$	0
$\text{polyQuotient}(x^2-1,x-1)$	$x+1$
$\text{polyQuotient}(x^3-6\cdot x^2+11\cdot x-6,x^2-6\cdot x+8)$	x
$\text{polyQuotient}((x-y)\cdot(y-z),x+y+z,x)$	$y-z$
$\text{polyQuotient}((x-y)\cdot(y-z),x+y+z,y)$	$2\cdot x-y+2\cdot z$
$\text{polyQuotient}((x-y)\cdot(y-z),x+y+z,z)$	$-(x-y)$

polyRemainder()Katalogi > 

polyRemainder(Poly1,Poly2 [,Muutt])
⇒*lauseke*

$\text{polyRemainder}(x-1,x-3)$	2
$\text{polyRemainder}(x-1,x^2-1)$	$x-1$
$\text{polyRemainder}(x^2-1,x-1)$	0

polyRemainder()

Katalogi > 

Laskee jakojäännöksen polynomista *Poly1*, joka on jaettu polynomilla *Poly2* määritetyn muuttujan *Muutt* suhteen.

Poly1:n ja *Poly2*:n on oltava polynomilausekkeita muuttujassa *Muutt*. Suositeltavaa on, ettei jätä muuttujaa *Muutt* pois, elleivät *Poly1* ja *Poly2* ole lausekkeita yhdessä ja samassa muuttujassa.

$$\begin{array}{l} \text{polyRemainder}((x-y) \cdot (y-z), x+y+z, x) \\ \quad \quad \quad -(y-z) \cdot (2 \cdot y+z) \\ \hline \text{polyRemainder}((x-y) \cdot (y-z), x+y+z, y) \\ \quad \quad \quad -2 \cdot x^2 - 5 \cdot x \cdot z - 2 \cdot z^2 \\ \hline \text{polyRemainder}((x-y) \cdot (y-z), x+y+z, z) \\ \quad \quad \quad (x-y) \cdot (x+2 \cdot y) \end{array}$$

polyRoots()

Katalogi > 

polyRoots(*Poly*, *Muutt*) $\Rightarrow$ lista

polyRoots(*Kertoinlista*) $\Rightarrow$ lista

Ensimmäinen syntaksi, **polyRoots(*Poly*, *Muutt*)**, laskee polynomin *Poly* reaalityyppien listan muuttujan *Muutt* suhteen. Jos reaalityyppien lista on tyhjä, tuloksena on tyhjä lista: { }.

Poly on oltava polynomi yhdessä muuttujassa.

Toinen syntaksi, **polyRoots(*Kertoinlista*)**, askee reaalityyppien listan kertoimille, jotka sisältyvät *Kertoinlistaan*.

Huomaa: Katso myös **cPolyRoots()**, sivu 38.

$$\begin{array}{l} \text{polyRoots}(y^3+1, y) \quad \quad \quad \{-1\} \\ \hline \text{cPolyRoots}(y^3+1, y) \\ \quad \quad \quad \left\{ -1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i \right\} \\ \hline \text{polyRoots}(x^2+2 \cdot x+1, x) \quad \quad \quad \{-1, -1\} \\ \hline \text{polyRoots}(\{1, 2, 1\}) \quad \quad \quad \{-1, -1\} \end{array}$$

PowerReg

Katalogi > 

PowerReg *X*, *Y* [, *Frekv*] [, *Luokka*, *Sisällytä*]

Laskee potenssiregression $y = (a \cdot (x)^b)$ listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-listaa lukuunottamatta.

X ja *Y* ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen *X* ja *Y* esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle *X*- ja *Y* -datalle.

Sisällytä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a \cdot (x)^b$
stat.a, stat.b	Regressiokertoimet
stat.r ²	Muunnettujen tietojen lineaarimäärittelyn kerroin
stat.r	Muunnettujen tietojen korrelaatiokerroin ($\ln(x)$, $\ln(y)$)
stat.Resid	Potenssimalliin liittyvät jäännökset
stat.ResidTrans	Muunnettujen tietojen lineaariseen sovitukseen liittyvät jäännökset
stat.XReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

Malli käyttäjän määrittämän ohjelman luomista varten. Käytetään komennon **Define**, **Define LibPub** tai **Define LibPriv** kanssa.

Lohko voi olla yksi lauseke tai sarja lausekkeita, jotka on erotettu toisistaan kaksoispisteellä (:), tai sarja eri riveillä olevia lausekkeita.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määrittysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

```
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()

Katso $\Pi()$, sivu 235.

Product (PI)

Katso $\Pi()$, sivu 235.

product()

product(Lista[, Alku[, loppu]])⇒*lauseke*

Laskee *Listan* sisältämien elementtien tulon. *Alku* ja *Loppu* ovat valinnaisia. Ne määrittävät elementtien alueen.

product(MatriisiI[, Alku[, loppu]])
⇒*matriisi*

Laskee rivivektorin, joka sisältää *MatriisiI*:n sarakkeiden elementtien tulot. *Alku* ja *Loppu* ovat valinnaisia. Ne määrittävät rivialueen.

product({ 1,2,3,4 })	24
product({ 2,x,y })	2·x·y
product({ { 4,5,8,9 },2,3 })	40
product($\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$)	$\begin{bmatrix} 28 & 80 & 162 \end{bmatrix}$
product($\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},1,2$)	$\begin{bmatrix} 4 & 10 & 18 \end{bmatrix}$

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

propFrac()

propFrac(*Laus* *l*, *Muutt*) ⇒ *lauseke*

propFrac(*rationaali_luku*) laskee *rationaali_luvun* kokonaisluvun summana ja murtolukuna, jolla on sama etumerkki ja suurempi nimittäjä kuin osoittaja.

propFrac(*rationaali_lauseke*, *Muutt*) laskee aitojen suhdelukujen summan ja polynomin muuttujan *Muutt* suhteen. Muuttujan *Muutt* aste nimittäjässä on suurempi kuin muuttujan *Muutt* aste osoittajassa kussakin aidossa suhdeluvussa. Muuttujan *Muutt* samanaiset potenssit kerätään. Termit ja niiden tekijät lajitellaan siten, että *Muutt* on päämuuttuja.

Jos *Muutt* jätetään pois, varsinaisen murtoluvun lavennus tehdään pääasiallisimmalla muuttujalla. Polynomiosan kertoimet tehdään sen jälkeen aidoiksi ensin niiden pääasiallisimman muuttujan suhteen ja niin edelleen.

Rationaalilausekkeissa **propFrac()** on nopeampi, mutta vähemmän äärimmäinen vaihtoehto kuin **expand()**.

propFrac()-funktion avulla voit esittää sekalukuja ja suorittaa sekalukujen yhteen- ja vähennyslaskua.

$$\begin{array}{rcl} \text{propFrac}\left(\frac{4}{3}\right) & & 1 + \frac{1}{3} \\ \text{propFrac}\left(\frac{-4}{3}\right) & & -1 - \frac{1}{3} \end{array}$$

$$\begin{array}{rcl} \text{propFrac}\left(\frac{x^2+x+1}{x+1} + \frac{y^2+y+1}{y+1}, x\right) & & \frac{1}{x+1} + x + \frac{y^2+y+1}{y+1} \\ \text{propFrac}(\text{Ans}) & & \frac{1}{x+1} + x + \frac{1}{y+1} + y \end{array}$$

$$\begin{array}{rcl} \text{propFrac}\left(\frac{11}{7}\right) & & 1 + \frac{4}{7} \\ \text{propFrac}\left(3 + \frac{1}{11} + 5 + \frac{3}{4}\right) & & 8 + \frac{37}{44} \\ \text{propFrac}\left(3 + \frac{1}{11} - \left(5 + \frac{3}{4}\right)\right) & & -2 - \frac{29}{44} \end{array}$$

Q

QR *Matriisi*, *qMatNimi*, *rMatNimi*[, *Tol*]

Suorittaa Householderin QR-dekomponoinnin reaaililuku- tai kompleksilukumatriisista. Tuloksena ovat Q- ja R-matriisit tallentuvat määritettyihin *MatNimi*-muuttujiin. Q-matriisi on jakamaton. R-matriisi on yläkolmiomatriisi.

Valinnaisesti kaikkia matriisielementtejä käsitellään nollana, jos niiden itseisarvo on pienempi kuin *Tol*. Tätä toleranssia käytetään vain, jos matriisissa on liukulukusyötteitä eikä se sisällä symbolisia muuttujia, joille ei ole määritetty arvoa. Muussa tapauksessa *Tol*-komentoa ei huomioida.

- Jos käytät painikkeita   tai **Automaattinen tai likimääräinen** -tilan valintaa Approximate (Likimääräinen), laskut suoritetaan liukulukuaritmetiikalla.
- Jos *Tol* jätetään pois tai sitä ei käytetä, oletusarvoinen toleranssi lasketaan seuraavasti:
 $5E-14 \cdot \max(\dim(\text{Matriisi})) \cdot \text{rowNorm}(\text{Matriisi})$

QR-hajotus lasketaan numeerisesti Householderin transformaatioiden avulla. Symbolinen ratkaisu lasketaan Gram-Schmidtin menetelmällä. Matriisin *qMatNimi* sarakkeet ovat ne ortonormitetut perusvektorit, jotka sijoittuvat *matriisin* määrittämään tilaan.

Liukuluku (9.) muuttujassa m1 aiheuttaa sen, että vastaukset lasketaan liukulukumuodossa.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
QR <i>m1,qm,rm</i> Done	
<i>qm</i>	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
QR <i>m1,qm,rm</i> Done	
<i>qm</i>	$\begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} \sqrt{m^2+o^2} & \frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{m \cdot p - n \cdot o}{\sqrt{m^2+o^2}} \end{bmatrix}$

QuadReg

QuadReg *X,Y* [, *Frekv*] [, *Luokka*, *Sisällytä*]

Laskee 2. asteen polynomiregression $y = a \cdot x^2 + b \cdot x + c$ listausta X ja Y frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle X - ja Y -datalle..

Sisällytä on yhden tai usemman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Regressiokertoimet
stat.R ²	Määrittäyskerroin
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatun <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

QuartReg *X,Y* [, *Frekv*] [, *Luokka*,
Sisällytä]]

Laskee 4. asteen polynomiregression
 $y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$ elistoista *X* ja *Y*
 frekvenssillä *Frekv*. Tulosten yhteenveto
 tallentuu *stat.results*-muuttujaan.
 (Katso sivu sivu 186.)

Kaikkien listojen on oltava samankokoisia
Sisällytä-listaa lukuunottamatta.

X ja *Y* ovat riippumattomien ja riippuvien
 muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista.
 Jokainen *Frekv*:n elementti määrittää
 kunkin vastaavan datapisteen *X* ja *Y*
 esiintymisfrekvenssin. Oletusarvo on 1.
 Kaikkien elementtien on oltava
 kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle
X- ja *Y* -dataalle.

Sisällytä on yhden tai usemman
 luokkakoodin lista. Vain ne datayksiköt,
 joiden luokkakoodi sisältyy tähän listaan,
 ovat mukana laskutoimituksessa.

Lisätietoja listassa olevien tyhjien
 elementtien vaikutuksesta, katso Tyhjät
 elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Regressiokertoimet
stat.R ²	Määrittyskerroin
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

R►Pθ()Luettelo > **R►Pθ** (*xLaus*, *yLaus*) ⇒ *lauseke***R►Pθ** (*xLista*, *yLista*) ⇒ *lista***R►Pθ** (*xMatriisi*, *yMatriisi*) ⇒ *matriisi*

Laskee yhtäpitävän θ-koordinaatin argumenttiparille (x,y).

Huomaa: Vastaus lasketaan aste-, graadi- tai radiaanikulmana käytössä olevan kulmatila-asetuksen mukaisesti.

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla **R@>Ptheta** (...).

Astekulmatilassa:

$$\text{R►P}\theta(x,y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Graadikulmatilassa:

$$\text{R►P}\theta(x,y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Radiaanikulmatilassa:

$$\text{R►P}\theta(3,2) \quad \tan^{-1}\left(\frac{2}{3}\right)$$

$$\text{R►P}\theta\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right) \\ \left[0 \quad \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} \quad 0.643501 \right]$$

R►Pr()Luettelo > **R►Pr** (*xLaus*, *yLaus*) ⇒ *lauseke***R►Pr** (*xLista*, *yLista*) ⇒ *lista***R►Pr** (*xMatriisi*, *yMatriisi*) ⇒ *matriisi*

Laskee yhtäpitävän r-koordinaatin argumenttiparille (x,y).

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla **R@>Pr** (...).

Radiaanikulmatilassa:

$$\text{R►Pr}(3,2) \quad \sqrt{13}$$

$$\text{R►Pr}(x,y) \quad \sqrt{x^2+y^2}$$

$$\text{R►Pr}\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right) \\ \left[3 \quad \frac{\sqrt{\pi^2+256}}{4} \quad 2.5 \right]$$

►RadLuettelo > *Laus*►**Rad** ⇒ *lauseke*

Muuntaa argumentin radiaanikulmayksikköön.

Astekulmatilassa:

$$(1.5) \text{►Rad} \quad (0.02618)^r$$

Graadikulmatilassa:

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>Rad.

(1.5)►Rad

(0.023562)^r**rand()**Luettelo > **rand()** ⇒ lauseke**rand(Kokeiden lkm)** ⇒ lista

Aseta satunnainen siemenluku.

rand() laskee satunnaisen arvon välillä 0 ja 1.

RandSeed 1147

Done

rand(2)

{ 0.158206, 0.717917 }

rand(Kokeiden lkm) laskee listan, joka sisältää *Kokeiden lkm* satunnaista arvoa välillä 0 ja 1.

randBin()Luettelo > **randBin(n, p)** ⇒ lauseke**randBin(n, p, Kokeiden lkm)** ⇒ lista

randBin(80,0,5)

42

randBin(n, p) laskee satunnaisen reaaliluvun määrätystä binomijakaumasta.

randBin(80,0,5,3)

{ 41, 32, 39 }

randBin(n, p, Kokeiden lkm) laskee listan, joka sisältää *Kokeiden lkm* satunnaista reaalilukua määrätystä binomijakaumasta.

randInt()Luettelo > **randInt**

(
Muutt.alaraja
,Muutt.yläaraja) ⇒
lauseke

randInt

(
Muutt.alaraja
,Muutt.yläaraja
,Kokeiden lkm) ⇒
lista

randInt(3,10)

5

randInt(3,10,4)

{ 9, 7, 5, 8 }

randInt()

Luettelo > 

randInt

(
Muutt.alaraja
,Muutt.yläraja)
laskee satunnaisen
kokonaisluvun
rajojen *Muutt.alaraja*
ja *Muutt.yläraja*
määräämällä
kokonaislukuvälillä.

randInt

(
Muutt.alaraja
,Muutt.yläraja
,Kokeiden lkm)
laskee listan, joka
sisältää *Kokeiden*
lkm satunnaista
kokonaislukua
määrätyllä välillä.

randMat()

Luettelo > 

**randMat(numRivit, numSarakkeet) ⇒
matriisi**

Laskee välillä -9 ja 9 määrätylle mitalle
kokonaislukumatriisin.

Kummankin argumentin on sievennyttävä
kokonaisluvuksi.

RandSeed 1147	Done		
randMat(3,3)	8	-3	6
	-2	3	-6
	0	4	-6

Huomaa: Tämän matriisin arvot muuttuvat
aina, kun painat .

randNorm()

Luettelo > 

randNorm(μ , σ) ⇒ lauseke
randNorm(μ , σ , *Kokeiden lkm*) ⇒ lista

randNorm(μ , σ) laskee desimaaliluvun
määrätystä normaali jakaumasta. Luku voi
olla mikä tahansa reaali luku, mutta se
keskittyy voimakkaasti välille [$\mu-3\cdot\sigma$,
 $\mu+3\cdot\sigma$].

randNorm(μ , σ , *Kokeiden lkm*) listan (list),
joka sisältää *Kokeiden lkm* desimaalilukua
määrätystä normaali jakaumasta.

RandSeed 1147	Done
randNorm(0,1)	0.492541
randNorm(3,4,5)	-3.54356

randPoly()

Luettelo > 

randPoly(*Muutt*, *Aste*) $\Rightarrow$ *lauseke*

Laskee polynomin muuttujasta *Muutt* määrättyssä järjestyksessä *Aste*. Kertoimet ovat satunnaisia kokonaislukuja välillä -9 ja 9 . Johtava kerroin ei ole 0 .

Asteen on oltava $0-99$.

RandSeed 1147	Done
$\text{randPoly}(x,5)$	$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$

randSample()

Luettelo > 

randSample(*List*, *Kokeiden lkm*[, *noRepl*])
 $\Rightarrow$ *lista*

Laskee satunnaisesta otoksesta listan, joka koostuu valinnoista *Kokeiden lkm* listasta *List* valinnaisena otoksen korvaaminen muulla (*noRepl*= 0), tai ilman otoksen korvaamista (*noRepl*= 1). Oletusarvona on otoksen korvaaminen

Define <i>list3</i> ={1,2,3,4,5}	Done
Define <i>list4</i> =randSamp(<i>list3</i> ,6)	Done
<i>list4</i>	{2,3,4,3,1,2}

RandSeed

Luettelo > 

RandSeed *Luku*

Jos *Luku* = 0 , tämä asettaa siemenluvut satunnaislukugeneraattorin tehdasasetuksiin. Jos *Luku* $\neq 0$, sitä käytetään luomaan kaksi siemenlukua, jotka tallennetaan järjestelmän muuttujiin siemen1 ja siemen2

RandSeed 1147	Done
rand()	0.158206

real()

Luettelo > 

real(*Lausl*) $\Rightarrow$ *lauseke*

Laskee argumentin reaali osan.

Huomaa: Kaikkia määrittelemättömiä muuttujia käsitellään reaali muuttujina. Katso myös **imag()**, page 94.

real(*Listl*) $\Rightarrow$ *lista*

Laskee kaikkien alkutekijöiden reaali osat

real(*Matriisi l*) $\Rightarrow$ *matriisi*

Laskee kaikkien alkutekijöiden reaali osat.

$\text{real}(2+3 \cdot i)$	2
$\text{real}(z)$	z
$\text{real}(x+i \cdot y)$	x
$\text{real}(\{a+i \cdot b, 3, i\})$	{a,3,0}
$\text{real}\left(\begin{bmatrix} a+i \cdot b & 3 \\ c & i \end{bmatrix}\right)$	$\begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$

Vectr ► Rect

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>Rect.

Näyttää *Vektorin* suorakulmamuodossa [x, y, z] Vektorin on oltava kooltaan 2 tai 3, ja se voi olla rivi tai sarake.

Huomaa: ►Rect on näyttömuodon ohje, ei muunnosfunktio. Voit käyttää komentoa ainoastaan syöterivin lopussa, eikä se päivitä arvoa *ans*.

Huomaa: Katso myös ►Polar, sivu 139.

kompleksiArvo ► Rect

Näyttää kompleksisen arvon *complexValues* suorakulmaisessa muodossa a+bi. Arvolla *complexValue* voi olla mikä tahansa kompleksinen muoto. Syöte $\text{re}^{i\theta}$ aiheuttaa kuitenkin virheen astekulmatilassa.

Huomaa: Polaarisisä syötteessä ($r \angle \theta$) on käytettävä sulkeita.

$$\left(\begin{bmatrix} 3 & \angle \frac{\pi}{4} & \angle \frac{\pi}{6} \end{bmatrix} \right) \text{►Rect} \quad \left[\begin{array}{ccc} \frac{3 \cdot \sqrt{2}}{4} & \frac{3 \cdot \sqrt{2}}{4} & \frac{3 \cdot \sqrt{3}}{2} \\ \frac{a \cdot \cos(b) \cdot \sin(c)}{a \cdot \sin(b) \cdot \sin(c)} & a \cdot \sin(b) \cdot \sin(c) & a \cdot \cos(c) \end{array} \right]$$

Radiaanikulmatilassa:

$$\left(\begin{bmatrix} 4 \cdot e^{\frac{\pi}{3}} \end{bmatrix} \right) \text{►Rect} \quad 4 \cdot e^{\frac{\pi}{3}} \quad \left(\left(4 \angle \frac{\pi}{3} \right) \right) \text{►Rect} \quad 2 + 2 \cdot \sqrt{3} \cdot i$$

Graadikulmatilassa:

$$\left((1 \angle 100) \right) \text{►Rect} \quad i$$

Astekulmatilassa:

$$\left((4 \angle 60) \right) \text{►Rect} \quad 2 + 2 \cdot \sqrt{3} \cdot i$$

Huomaa: Näppäilläksesi merkin $\angle$ valitse se Luettelon symboliluettelosta.

ref()

ref(MatriisiI[, Tol]) ⇒ *matriisi*

Laskee matriisin *MatriisiI* rivi-echelon-muodossa.

$$\text{ref} \left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix} \right) \quad \left[\begin{array}{cccc} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{array} \right]$$

Valinnaisesti kaikkia matriisin elementtejä käsitellään nollana, jos niiden itseisarvo on pienempi kuin *Tol*. Tätä toleranssia käytetään vain, jos matriisissa on liukulukusyötteitä eikä se sisällä symbolisia muuttujia, joille ei ole määrätty arvoa. Muussa tapauksessa komentoa *Tol* ei huomioida.

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$
ref(<i>m1</i>)	$\begin{bmatrix} 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix}$

- Jos käytät painikkeita **ctrl** **enter** tai asetat tilan **Auto or Approximate**, laskenta suoritetaan liukulukuaritmetiikalla.
- Jos *Tol* jätetään pois tai sitä ei käytetä, oletusarvoinen toleranssi lasketaan seuraavasti:
 $5E-14 \cdot \max(\dim(\text{Matriisi } I)) \cdot \text{rowNorm}(\text{Matriisi } I)$

Vältä määrittelemättömiä alkutekijöitä matriisissa *Matriisi I*. Tuloksena saattaa olla odottamattomia vastauksia.

Jos esimerkiksi *a* on määrittelemätön seuraavassa lausekkeessa, näkyviin tulee varoitusviesti, ja vastaus näyttää seuraavalta:

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) = \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Varoitus näkyy, koska yleistetty alkutekijä $1/a$ ei olisi mahdollinen lausekkeelle $a=0$.

Voit välttää tämän tallentamalla etukäteen arvon *a*:lle tai käyttämällä rajoittavaa ("|")-operaattoria arvon korvaamiseksi, kuten on esitetty seuraavassa esimerkissä.

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mid a=0\right) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Huomaa: Katso myös **rref()**, page 164.

RefreshProbeVars

Pääset mittausravoon kaikista TI-Basic-ohjelmaan kytketyistä antureista.

**StatusVar
Value**

Tila

statusVar
=0

Normaali (jatka ohjelmalla)

Vernier DataQuest™ on tiedonkeruutilassa.

statusVar
=1

Huomaa: Vernier DataQuest™ -sovelluksen on oltava metrisessä tilassa, jotta tämä komento toimisi.

statusVar
=2

Vernier DataQuest™ -sovellusta ei ole käynnistetty.

statusVar
=3

The Vernier DataQuest™-sovellus on käynnistetty, mutta et ole yhdistänyt mitään antureita.

Esimerkki

Lämmön määrittäminen=

Prgm

© Tarkista, onko järjestelmä valmiina

RefreshProbeVars-tila

Jos tila=0, niin

Käyttötila "valmis"

Muuttujalle n,1,50

RefreshProbeVars-tila

lämpötila:=mittari.lämpötila

Disp "Lämpötila: ",lämpötila

Jos lämpötila>30, niin

Disp "Liian kuuma"

LopetaJos

© Odota 1 sekunti otoksien välillä

Odota 1

LopetaKoska

Tai

Disp "Ei valmis". Yritä myöhemmin uudelleen"

EndIf

EndPrgm

Huomaa: Tätä voi käyttää myös TI-Innovator™-laitteen keskiössä.

remain()

Luettelo > 

remain(*Laus1*, *Laus2*) $\Rightarrow$ *lauseke*

remain(*Lista1*, *Lista2*) $\Rightarrow$ *lista*

remain(*Matriisi1*, *Matriisi2*) $\Rightarrow$ *matriisi*

Laskee ensimmäisen argumentin jäännöksen toisen argumentin suhteen seuraavien identtisten yhtälöiden määritelmien mukaisesti:

$\text{remain}(x,0) \quad x$

$\text{remain}(x,y) \quad x - y \cdot \text{Part}(x/y)$

Tämän seurauksena huomaa, että **remain**($-x,y$) = **remain**(x,y). Vastaus on joko nolla tai samanmerkkinen kuin ensimmäinen argumentti.

Huomaa: Katso myös **mod()**, sivu 121.

$\text{remain}(7,0)$	7
$\text{remain}(7,3)$	1
$\text{remain}(-7,3)$	-1
$\text{remain}(7,-3)$	1
$\text{remain}(-7,-3)$	-1
$\text{remain}(\{12,-14,16\},\{9,7,-5\})$	$\{3,0,1\}$

$\text{remain}\left(\begin{pmatrix} 9 & -7 \\ 6 & 4 \end{pmatrix}, \begin{pmatrix} 4 & 3 \\ 4 & -3 \end{pmatrix}\right)$	$\begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix}$
--	---

Pyydä

Luettelo > 

Pyydä *promptString*, *muutt*[, *NäytäLippu* [, *statusVar*]]

Pyydä *promptString*, *func*(*arg1*, ...*argn*) [, *NäytäLippu* [, *statusVar*]]

Ohjelmointikomento: Keskeyttää ohjelman ja näyttää valintaruudun, jossa on viesti *kehoteimerkkijono* sekä syöttöruutu käyttäjän antamaa vastausta varten.

Kun käyttäjä kirjoittaa vastauksen ja klikkaa **OK**-painiketta, syöttöruudun sisältö määrätään muuttujalle *var*.

Jos käyttäjä klikkaa **Cancel**, ohjelma etenee hyväksymättä mitään syötteitä. Ohjelma käyttää muuttujan *muutt* aikaisempaa arvoa, jos *var* on jo määritelty.

Valinnainen *NäytäLippu* -argumentti voi olla mikä tahansa lauseke.

- Jos *NäytäLippu* jätetään pois, tai jos se sievennyy arvoon **1**, kehoteviesti ja käyttäjän vastaus näkyvät laskimen historiassa.
- Jos *NäytäLippu* sievennyy arvoon **0**, kehote ja vastaus eivät näy historiassa.

Määrittele ohjelma:

Määrittele *pyydä_demo()*=Ohjelma

Pyydä "Säde: ",*r*

Disp "Ala = ",*pi***r*²

EndPrgm

Aja ohjelma ja kirjoita vastaus:

Pyydä_demo()



Vastaus **OK**-näppäimen painamisen jälkeen:

Säde: 6/2

Ala = 28.2743

Valinnainen *statusVar* argumentti laskee ohjelmalle tavan määritellä, kuinka käyttäjä on sulkenut valintaikkunan. Huomaa, että *statusVar* vaatii *NäytäLippu*-argumentin.

- Jos käyttäjä on klikannut **OK** tai painanut **Enter** tai **Ctrl+Enter**, muuttuja *statusVar* asetetaan arvoon **1**.
- Muuten muuttujastatus*Var* asetetaan arvoon **0**.

func()-argumentin avulla ohjelma voi tallentaa käyttäjän vastauksen funktion määritelmäksi. Tämä syntaksi toimii ikään kuin käyttäjä suorittaisi komennon:

Määrittele *func(arg1, ...argn)* =
käyttäjän vastaus

Sen jälkeen ohjelma voi käyttää määriteltyä funktiota *func()*. Komennon *promptString* pitäisi opastaa käyttäjää syöttämään sopiva *käyttäjän vastaus*, joka täydentää funktion määritelmän.

Huomaa: Voit käyttää Pyydä komento käyttäjän määrittelemässä ohjelmassa, mutta ei funktiossa.

Pysäytä ohjelma, joka sisältää komennon **Request** loputtoman silmukan sisällä:

- **Kämmenlaite:** Pidä -painiketta painettuna ja paina toistuvasti -painiketta.
- **Windows®:** Pidä **F12**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **Macintosh®:** Pidä **F5**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **iPad®:** Sovellus näyttää kehotuksen. Voit jatkaa odottamista tai peruuttaa.

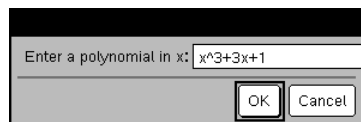
Huomaa: Katso myös **RequestStr**, page 158.

Määrittele ohjelma:

```
Määrittele polynomi()=Ohjelma
  Pyyntö "Syötä polynomi x:ään:",p
(x)
  Disp "Reaalilukujuuret
ovat:",polyRoots(p(x),x)
EndPrgm
```

Aja ohjelma ja kirjoita vastaus:

polynomi()



Tulos, kun x^3+3x+1 on syötetty ja valittu **OK**:

Reaalilukujuuret ovat: $\{-0.322185\}$

RequestStr *kehotemerkkijono, muutt[, NäytäLippu]*

Määrittele ohjelma:

Ohjelmointikomento: Toimii samalla tavalla kuin **Request**-komennon ensimmäinen syntaksi paitsi, että käyttäjän vastaus tulkitaan aina merkkijonoksi.

Vastakohtaisesti **Request**-komento tulkitsee vastauksen lausekkeeksi, jollei käyttäjä merkitse sitä lainausmerkkien ("") sisään.

Huomaa: Voit käyttää **RequestStr** -komentoa käyttäjän määrittelemässä ohjelmassa, mutta ei funktiossa.

Pysäytä ohjelma, joka sisältää **RequestStr** komento äärettömässä silmukassa:

- **Kämmenlaite:** Pidä  -painiketta painettuna ja paina toistuvasti  -painiketta.
- **Windows®:** Pidä **F12**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **Macintosh®:** Pidä **F5**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **iPad®:** Sovellus näyttää kehotuksen. Voit jatkaa odottamista tai peruuttaa.

Huomaa: Katso myös **Pyydä**, page 157.

```
Määrittele requestStr_demo()=Ohjelma
  RequestStr "Oma nimesi:",nimi,0
  Disp "Vastauksessa on ",dim
(nimi)," merkkiä."
EndPrgm
```

Aja ohjelma ja kirjoita vastaus:

requestStr_demo()



Vastaus **OK**-näppäimen painamisen jälkeen (Huomaa, että jos **NäytäLippu**-argumentti on **0**, kehote ja vastaus eivät näy historiassa):

requestStr_demo()

Vastauksessa on 5 merkkiä.

Return [Laus]

Laskee lausekkeen *Laus* funktion tuloksena. Käytetään lohkon **Func...EndFunc** sisällä.

Huomaa: Käytä **Return** ilman argumenttia lohossa **Prgm...EndPrgm** poistuaaksesi ohjelmasta.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

```
Define factorial (nn)=
Func
Local answer,counter
1 → answer
For counter,1,nn
answer·counter → answer
EndFor
Return answer
EndFunc

factorial (3)
```

6

right()

oikea(*ListI*[, *Num*]) ⇒ *lista*

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

Laskee oikeanpuoleisimmat *Num*-alkutekijät, jotka sisältyvät listaan *List1*.

Jos jätät pois komennon *Num*, laskee kaiken listasta *List1*.

right(lädemerkkijono[, Num]) ⇒ *merkkijono*

right("Hello",2)	"lo"
------------------	------

Laskee oikeanpuoleisimmat *Num*-merkit, jotka sisältyvät merkkijonoon *lähdemerkkijono*

Jos jätät pois komennon *Num*, laskee kaiken *lähdemerkkijonosta*.

right(Vertailu) ⇒ *lauseke*

right(x<3)	3
------------	---

Laskee yhtälön tai epäyhtälön oikean puolen.

rk23 ()

Luettelo >

rk23(Laus, Var, depVar, {Var0, VarMax}, depVar0, VarStep[, diftol]) ⇒ *matriisi*

Differentiaaliyhtälö:

$y' = 0.001 \cdot y \cdot (100 - y)$ ja $y(0) = 10$

rk23(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol]) ⇒ *matriisic*

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	

rk23(ListOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol]) ⇒ *matriisi*

Jos haluat nähdä koko vastauksen, paina ► ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ►.

Käyttää Runge-Kutta-menetelmää järjestelmän ratkaisuun

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

Sama yhtälö, jossa *diftol* on asetettu 1.E-6

funktiolla *depVar(Var0)=depVar0* välillä [*Var0*,*VarMax*]. Laskee tulokseksi matriisiin, jonka ensimmäinen rivi määrittelee *Var* tulosarvot muuttujan *VarStep* määrittelemällä tavalla. Toinen rivi määrää ensimmäisen ratkaisukomponentin arvon vastaavissa *Var*-arvoissa, jne.

rk23(0.001·y·(100−y),t,y,{0,100},10,1,1.E-6)					
0.	1.	2.	3.	4.	
10.	10.9367	11.9495	13.0423	14.2189	

Vertaile edellä olevaa tulosta CAS:n täsmälliseen tulokseen, joka on saatu käyttämällä deSolve()- ja seqGEN()-funktioita:

deSolve(y'=0.001·y·(100−y) and y(0)=10,t,y)
$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}$

Expr on oikea puoli, joka määrittelee tavallisen differentiaaliyhtälön (ODE).

SystemOfExpr on oikeiden puolten ryhmä, joka määrittelee ODE-yhtälöiden järjestelmän (vastaa riippuvaisten muuttujien järjestystä kohdassa *ListOfDepVars*).

ListOfExpr on oikeiden puolten luettelo, joka määrittelee ODE-yhtälöiden järjestelmän (vastaa riippuvaisten muuttujien järjestystä kohdassa *ListOfDepVars*).

Var on riippumaton muuttuja.

ListOfDepVars on riippuvaisten muuttujien luettelo.

{*Var0*, *VarMax*} on kahden elementin lista, joka määrää funktion integroimaan muuttujasta *Var0* to *VarMax*.

ListOfDepVars0 on lista riippuvaisten muuttujien alkuarvoja.

Jos *VarStep* on arvoltaan nollasta eroava numero: merkki(*VarStep*) = merkki (*VarMax-Var0*) ja ratkaisut lasketaan muuttujalla *Var0+i*VarStep* kaikille *i*=0,1,2,... niin, että *Var0+i*VarStep* on alueella [*muutt0*,*VarMax*] (muuttujalla *VarMax* ei ehkä ole ratkaisuarvoa).

jos *VarStep* sievenee nolllaksi, ratkaisut lasketaan "Runge-Kutta" -muuttujan *Muutt* arvoilla.

difftol on virhetoleranssi (oletuksena 0.001).

$$\text{seqGen}\left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}, t, y, \{0, 100\}\right) \\ \{10., 10.9367, 11.9494, 13.0423, 14.2189, 15.49\}$$

Yhtälöryhmä:

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

kun *y1(0)*=2 and *y2(0)*=5

$$\text{rk23}\left(\begin{cases} -y1+0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right) \\ \begin{array}{ccccc} 0. & 1. & 2. & 3. & 4. \\ 2. & 1.94103 & 4.78694 & 3.25253 & 1.82848 \\ 5. & 16.8311 & 12.3133 & 3.51112 & 6.27245 \end{array}$$

root(Laus) ⇒ *juuri*

root(Laus1, Laus2) ⇒ *juuri*

root(Laus) laskee neliöjuuren lausekkeelle *Laus*.

$$\begin{array}{cc} \sqrt[3]{8} & 2 \\ \sqrt[3]{3} & \frac{1}{3} \\ \sqrt[3]{3.} & 1.44225 \end{array}$$

root(*Laus1*, *Laus2*) laskee lausekkeena *Laus2* lausekkeen *Laus1* juuren. *Laus1* voi olla reaalinen tai kompleksinen liukulukuvakio, kokonaisluku tai kompleksinen rationaalilukuvakio tai yleinen symbolinen lauseke.

Huomaa: Katso myös **N:s juuri -malli**, sivu 2.

rotate()

Luettelo >

rotate(*Kokonaisluku1* [, *KiertojenLkm*]) ⇒ *kokonaisluku*

Kiertää bittejä binaarisessa kokonaisluvussa. Voit syöttää luvun *Kokonaisluku1* minä tahansa kantelukuna; se muunnetaan automaattisesti 64 bitin binaarimuotoon. Jos *Kokonaisluku1* on liian suuri tälle muodolle, symmetrinen modulo-operaatio sovittaa sen alueelle sopivaksi. Katso lisätietoa kohdasta ► **Kantaluku2**, sivu 18.

Jos *KiertojenLkm* on positiivinen, kierto on vasemmalle. Jos *KiertojenLkm* on negatiivinen, kierto on oikealle. Oletusarvo on -1 (kierrä oikealle yksi bitti).

Esimerkki, kierrossa oikealle:

Jokainen bitti kiertyy oikealle.

0b00000000000001111010110000110101

Oikeanpuoleisin bitti siirtyy vasemmaisimmaksi.

tuottaa:

0b100000000000000111101011000011010

Vastaus näkyy kantelukutilan mukaisesti.

rotate(*Listal* [, *KiertojenLkm*]) ⇒ *lista*

Laskee *Listal*:n kopion, jota on kierretty oikealle tai vasemmalle *KiertojenLkm*:n elementtien määrittämän määrän. Ei muuta *Listal*.

Binaarisessa kantelukutilassa:

rotate(0b11111111111111111111111111111111)	
0b100000000000000000000000000000000001	
rotate(256,1)	0b1000000000

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

Heksadesimaalisessa kantelukutilassa:

rotate(0h78E)	0h3C7
rotate(0h78E, -2)	0h8000000000000001E3
rotate(0h78E, 2)	0h1E38

Tärkeää: Binaariluvun edelle tulee aina merkitä etumerkki 0b ja heksadesimaaliluvun edelle 0h (nolla, ei O-kirjain).

Desimaalisessa kantelukutilassa:

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}

Jos *KiertojenLkm* on positiivinen, kierto on vasemmalle. Jos *KiertojenLkm* on negatiivinen, kierto on oikealle. Oletusarvo on -1 (kierrä oikealle yksi elementti).

rotate(*MerkkijonoI* [, *KiertojenLkm*]) $\Rightarrow$ *merkkijono*

Laskee *MerkkijonoI*:n kopion, jonka on kiertänyt oikealle tai vasemmalle *KiertojenLkm*:n merkkien määrittämän määrän. Ei muuta *MerkkijonoI*:tä

Jos *KiertojenLkm* on positiivinen, kierto on vasemmalle. Jos *KiertojenLkm* on negatiivinen, kierto on oikealle. Oletusarvo on -1 (kierrä oikealle yksi merkki).

rotate("abcd")	"dabc"
rotate("abcd", -2)	"cdab"
rotate("abcd", 1)	"bcda"

round()

round(*LausI* [, *numeroa*]) $\Rightarrow$ *lauseke*

Laskee argumentin pyöristettynä jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

numeroiden on oltava kokonaisluku välillä 0–12. Jos *numeroita* ei sisällytetä, laskee argumentin pyöristettynä 12 merkitykselliseen numeroon.

Huomaa: Näytettävät numerot -tila saattaa vaikuttaa näyttöön.

round(*ListaI* [, *numerot*]) $\Rightarrow$ *lista*

Pyöristää alkutekijöiden listan jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

round(*MatriisiI* [, *numerot*]) $\Rightarrow$ *matriisi*

Pyöristää alkutekijöiden matriisin jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

round(1.234567,3)	1.235
-------------------	-------

round($\{\pi, \sqrt{2}, \ln(2)\}, 4$)	$\{3.1416, 1.4142, 0.6931\}$
---	------------------------------

round($\begin{bmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{bmatrix}, 1$)	$\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$
--	--

rowAdd()Luettelo > 

rowAdd(Matriisi1, rIndeksi1, rIndeksi2)
 $\Rightarrow$ matriisi

Laskee matriisin *Matriisi1* kopion
 rivirIndeksi2 korvattuna rivien
 summalla rIndeksi1 ja rIndeksi2.

$$\begin{array}{l} \text{rowAdd} \left(\begin{bmatrix} 3 & 4 \\ -3 & -2 \end{bmatrix}, 1, 2 \right) \quad \begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix} \\ \text{rowAdd} \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2 \right) \quad \begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix} \end{array}$$

rowDim()Luettelo > 

rowDim(Matriisi) $\Rightarrow$ lauseke

Laskee *Matriisin* sisältämien rivien
 lukumäärän.

Huomaa: Katso myös **colDim()**, sivu 27.

$$\begin{array}{l} \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \\ \text{rowDim}(m1) \quad 3 \end{array}$$

rowNorm()Luettelo > 

rowNorm(Matriisi) $\Rightarrow$ lauseke

Laskee *Matriisin* riveillä olevien
 elementtien itseisarvojen summien
 maksimin.

Huomaa: Kaikkien matriisien alkutekijöiden
 on sievennyttävä luvuiksi. Katso myös
colNorm(), sivu 27.

$$\begin{array}{l} \text{rowNorm} \left(\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix} \right) \quad 25 \end{array}$$

rowSwap()Luettelo > 

rowSwap(Matriisi1, rIndeksi1, rIndeksi2)
 $\Rightarrow$ matriisi

Laskee matriisin *Matriisi1*, jossa rivien
 rIndeksi1 ja rIndeksi2 paikkoja on
 vaihdettu.

$$\begin{array}{l} \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow mat \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \\ \text{rowSwap}(mat, 1, 3) \quad \begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix} \end{array}$$

rref()Luettelo > 

rref(Matriisi1[, Tol]) $\Rightarrow$ matriisi

Laskee matriisin *Matriisi1* pelkistetyn rivi-
 echelon-muodon.

$$\begin{array}{l} \text{rref} \left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix} \right) \quad \begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ & 1 & 0 & \frac{147}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix} \end{array}$$

Valinnaisesti kaikkia matriisin elementtejä käsitellään nollana, jos niiden itseisarvo on pienempi kuin *Tol*. Tätä toleranssia käytetään vain, jos matriisissa on liukulukusyötteitä eikä se sisällä symbolisia muuttujia, joille ei ole määritelty arvoa. Muussa tapauksessa komentoa *Tol* ei huomioida.

 $\text{rref}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

- Jos käytät painikkeita   tai **Auto or Approximate** -tilan valintaa Approximate (Likimääräinen), laskenta suoritetaan liukulukuaritmetiikalla.
- Jos *Tol* jätetään pois tai sitä ei käytetä, oletusarvoinen toleranssi lasketaan seuraavasti:
 $5E-14 \cdot \max(\dim(\text{Matriisi } I)) \cdot \text{rowNorm}(\text{Matriisi } I)$

Huomaa: Katso myös **ref()**, page 154.

S

sec()

 -painike

sec(LausI) ⇒ lauseke

Astekulmatilassa:

sec(ListaI) ⇒ lista

Laskee *LausI*:n sekantin tai määrittää listan, joka sisältää *Listai*:n kaikkien elementtien sekantit.

$$\frac{\sec(45)}{\sec(\{1, 2, 3, 4\})} \sqrt{2} \left\{ \frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)} \right\}$$

Huomaa: Argumentti tulkitaan aste-, graadi- tai radiaanikulmaksi käytössä olevan kulmatila-asetuksen mukaisesti. Voit ohittaa kulmatilan väliaikaisesti painikkeilla $^{\circ}$, $^{\text{G}}$ tai $^{\text{r}}$.

sec⁻¹()

 -painike

sec⁻¹(LausI) ⇒ lauseke

Astekulmatilassa:

sec⁻¹(ListaI) ⇒ lista

$$\text{sec}^{-1}(1) \quad 0$$

sec⁻¹()**trig** -painike

Määrittää kulman, jonka sekantti on *LausI*, tai määrittää listan, joka sisältää *Listal*:n kaikkien elementtien käänteissekantit.

Graadikulmatilassa:

$$\sec^{-1}(\sqrt{2}) \quad 50$$

Radiaanikulmatilassa:

$$\sec^{-1}(\{1,2,5\}) \quad \left\{0, \frac{\pi}{3}, \cos^{-1}\left(\frac{1}{5}\right)\right\}$$

Huomaa: Vastaus lasketaan aste-, graadi- tai radiaanikulmana käytössä olevan kulmatila-asetuksen mukaisesti.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arcsec (...)**.

sech()**Katalogi** > **sech**(*LausI*) ⇒ *lauseke*

$$\operatorname{sech}(3) \quad \frac{1}{\cosh(3)}$$

sech(*Listal*) ⇒ *lista*

Laskee *LausI*:n hyperbolisen sekantin tai määrittää listan, joka sisältää *Listal*:n kaikkien elementtien hyperboliset sekantit.

$$\operatorname{sech}(\{1,2,3,4\}) \quad \left\{\frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)}\right\}$$

sech⁻¹()**Katalogi** > **sech⁻¹**(*LausI*) ⇒ *lauseke*

Radiaanikulma- ja

suorakulmakompleksitilassa:

sech⁻¹(*Listal*) ⇒ *lista*

Laskee *LausI*:n käänteisen hyperbolisen sekantin tai määrittää listan, joka sisältää *Listal*:n kaikkien elementtien käänteiset hyperboliset sekantit.

$$\operatorname{sech}^{-1}(1) \quad 0$$

$$\operatorname{sech}^{-1}(\{1,2,2.1\}) \quad \left\{0, \frac{2 \cdot \pi}{3} \cdot i, 8. \cdot 10^{-15} + 1.07448 \cdot i\right\}$$

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arcsech (...)**.

Send *lausTaiMerkkijono1* [, *lausTaiMerkkijono2*] ...

Ohjelmointikomento: Lähettää yhden tai useamman TI-Innovator™ Hub komennon liitettyyn laitteeseen.

*lausTaiMerkkijono*n täytyy olla validi TI-Innovator™ Hub Komento. Tavallisesti *lausTaiMerkkijono* sisältää "**SET** ..."komento laitteen tai "**READ ...**" hallitsemiseksi komento tiedon kysymiseksi.

Argumentit lähetetään laitteeseen peräkkäisinä.

Huomio: Komentoa **Send** voi käyttää käyttäjän määrittelemän ohjelman sisällä mutta ei funktion sisällä.

Huomio: Katso myös **Get** (sivu 82), **GetStr** (sivu 89), ja **eval()** (sivu 65).

Esimerkki: Sytytä sisäänrakennetun RGB-ledin sininen väri 0,5 sekunniksi.

Send "SET COLOR.BLUE ON TIME .5"	Done
----------------------------------	------

Esimerkki: Kysy laitteen sisäänrakennetun valotasanturin tämänhetkinen arvo. **Get**-komento noutaa arvon ja sijoittaa sen muuttujaan *lightval*.

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

Esimerkki: Lähetä laskettu taajuus laitteen sisäänrakennettuun kaiuttimeen. Käytä erikoismuuttujaa *iostr.SendAns* näyttääksesi laitekomennon ilmaus arvioituna.

$n:=50$	50
$m:=4$	4
Send "SET SOUND eval(m·n)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

seq() (sekv)

Katalogi >

seq(*Laus*, *Muutt*, *Matala*, *Korkea*[, *Askel*])
⇒*lista*

Lisää muuttujan *Var* arvoa arvosta *Low* arvoon *High* välillä *Step*, laskee *Expr*, ja antaa tulokset luettelona. Muuttujan *Var* alkuperäinen sisältö on edelleen tallessa funktion **seq()** suorittamisen jälkeen.

Oletusarvo *Step* = 1.

$\text{seq}\left(n^2, n, 1, 6\right)$	$\{1, 4, 9, 16, 25, 36\}$
$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$	$\left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	$\frac{1968329}{1270080}$

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina  .

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

iPad®: Pidä **enter** ja valitse .

$$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right) \quad 1.54977$$

seqGen()

seqGen(*Expr*, *Var*, *depVar*, {*Var0*, *VarMax*}, [*ListOfInitTerms*], [*VarStep*], [*CeilingValue*]]) ⇒ *lista*

Luo termiluettelon sekvenssille *depVar* (*Var*)=*Expr* seuraavasti: Lisää riippumattoman muuttujan *Var* arvoa arvosta *Var0* arvoon *VarMax* portailla *VarStep*, laskee *depVar*(*Var*) muuttujan *Var* vastaaville arvoille käyttäen *Expr*-kaavaa ja *ListOfInitTerms* -luetteloa, ja antaa tulokset luettelona

seqGen(*ListOrSystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, [*MatrixOfInitTerms*], [*VarStep*], [*CeilingValue*]]) ⇒ *matrix*

Luo termimatriisin sekvenssien *ListOfDepVars*(*Var*)=*ListOrSystemOfExpr* ryhmälle (tai listalle) seuraavasti: Lisää riippumattoman muuttujan *Var* arvoa arvosta *Var0* arvoon *VarMax* portailla *VarStep*, laskee *ListOfDepVars*(*Var*) muuttujan *Var* vastaaville arvoille käyttäen *Expr*-kaavaa ja *MatrixOfInitTerms* -luetteloa, ja antaa tulokset matriisina.

Muuttujan *Var* alkuperäinen sisältö on edelleen tallessa funktion **seqGen()** suorittamisen jälkeen

Oletusarvo *VarStep* = 1.

Luo sekvenssin 5 ensimmäistä termiä $u(n) = u(n-1)^2/2$, kun $u(1)=2$ ja $VarStep=1$.

$$\text{seqGen}\left(\frac{\{u(n-1)\}^2}{n}, n, u, \{1, 5\}, \{2\}\right) \\ \left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$$

Esimerkki, jossa $Var0=2$:

$$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right) \\ \left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$$

Esimerkki, jossa aloitustermi on symbolinen:

$$\text{seqGen}\{u(n-1)+2, n, u, \{1, 5\}, \{a\}\} \\ \{a, a+2, a+4, a+6, a+8\}$$

Kahden sekvenssin ryhmä:

$$\text{seqGen}\left\{\left\{\frac{1}{n}, \frac{u_2(n-1)}{2} + u_1(n-1)\right\}, n, \{u_1, u_2\}, \{1, 5\}, \left[-\frac{1}{2}\right]\right\} \\ \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix}$$

Huomaa: Muuttujaa Tyhjä () yllä olevassa aloitustermimatriisissa käytetään ilmoittamaan, että $u_1(n)$:n aloitustermi on laskettu käyttämällä täsmällistä sekvenssikaavaa $u_1(n)=1/n$.

seqn()Katalogi > 

seqn(*Expr*(*u*, *n* [, *ListOfInitTerms* [, *nMax* [, *CeilingValue*]]]) $\Rightarrow$ *list*

Luo termiluettelon sekvenssille $u(n)=Expr(u, n)$ seuraavasti: Lisää muuttujan *n* arvoa arvosta 1 arvoon *nMax* 1 välein, laskee lausekkeen $u(n)$ muuttujan *n* vastaaville arvoille käyttäen $Expr(u, n)$ -kaavaa ja *ListOfInitTerms* -luetteloa ja antaa tulokset luettelona.

seqn(*Expr*(*n* [, *nMax* [, *CeilingValue*]]]) $\Rightarrow$ *list*

Luo termiluettelon ei-rekursiiviselle sekvenssille $u(n)=Expr(u, n)$ seuraavasti: Lisää muuttujan *n* arvoa arvosta 1 arvoon *nMax* 1 välein, laskee lausekkeen $u(n)$ muuttujan *n* vastaaville arvoille käyttäen $Expr(n)$ -kaavaa ja antaa tulokset luettelona.

Jos arvo *nMax* puuttuu, *nMax* asetetaan arvoon 2500.

Jos *nMax*=0, *nMax* asetetaan arvoon 2500.

Huomaa: **seqn()** hakee funktion **seqGen()** kun *n0*=1 ja *nstep*=1

Luo sekvenssin 6 ensimmäistä termiä $u(n) = u(n-1)/2$, kun $u(1)=2$.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right) \quad \left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right) \quad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

series()Katalogi > 

series(*Lausl*, *Muutt*, *Aste* [, *Piste*])
 $\Rightarrow$ *muuttuja*

series(*Lausl*, *Muutt*, *Aste* [, *Piste*]) |
Muutt>*Piste* $\Rightarrow$ *lauseke*

series(*Lausl*, *Muutt*, *Aste* [, *Piste*]) |
Muutt<*Piste* $\Rightarrow$ *lauseke*

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1\right) \quad \frac{1}{2} \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$$

$$\text{series}\left(\frac{-1}{e^{z-}}, z, 1\right) \quad z-1$$

$$\text{series}\left(\left(1+\frac{1}{n}\right)^n, n, 2, \infty\right) \quad e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$$

Laskee yleisen tyypistetyn potenssisarjaesityksen *LausI*:stä, joka on lavennettu *Piste*en ympärillä asteen *Aste* kautta. *Aste* voi olla mikä tahansa rationaaliluku. Lausekkeen (*Muutt* – *Piste*) tuloksena olevilla potensseilla voi olla negatiiviset ja/tai murtolukueksponentit. Näiden potenssien kertoimet voivat sisältää lausekkeen (*Muutt* – *Piste*) logaritmeja ja muita *Muutt*:n funktioita, joita hallitsevat kaikki lausekkeen (*Muutt* – *Piste*) potenssit, joilla on sama eksponentin etumerkki.

*Piste*en oletusarvo on 0. *Piste* voi olla ∞ tai $-\infty$, jolloin lavennus tapahtuu asteen *Aste* kautta lausekkeessa $1/(\text{Muutt} - \text{Piste})$.

series(...) antaa tuloksena “**series(...)**”, ellei se pysty määrittämään tällaista esitystä, kuten olennaisille erikoispisteille, esim. $\sin(1/z)$, kun $z=0$, $e^{-1/z}$, kun $z=0$, tai e^z , kun $z = \infty$ tai $-\infty$.

Jos sarjassa tai yhdellä sen derivaatoista on hyppypäjätkuvuus kohdassa *Piste*, tulos sisältää todennäköisesti alalausekkeita, jotka ovat muotoa $\text{sign}(\dots)$ tai $\text{abs}(\dots)$ reaaliselle kehittelmän muuttujalle tai $(-1)^{\text{floor}(\dots \cdot \text{angle}(\dots) \dots)}$ kompleksille kehittelmän muuttujalle, joka on merkkiin “_” päättyvä muuttuja. Jos tarkoituksesi on käyttää sarjaa vain *Piste*en toisella puolella oleville arvoille, siinä tapauksessa liitä sopiva lauseke, “| *Muutt* > *Piste*”, “| *Muutt* < *Piste*”, “| *Muutt* ≥ *Piste*” tai “*Muutt* ≤ *Piste*”, jotta saat yksinkertaisemman vastauksen.

series() voi antaa symbolisia likiarvoja määrittämättömille integraaleille ja määrättyille integraaleille, joille ei muuten voida saada symbolisia ratkaisuja.

series() jakautuu 1. argumentin listoihin ja matriiseihin.

series() on **taylor()**-funktion yleistetty versio.

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5\right), x > 0$$

$$\frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$$

$$\text{series}\left(\int \frac{\sin(x)}{x} dx, x, 6\right)$$

$$x - \frac{x^3}{18} + \frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7\right)$$

$$\frac{x^3}{2} - \frac{x^5}{24} + \frac{29 \cdot x^7}{720}$$

$$\text{series}\left(\left(1 + e^x\right)^2, x, 2, 1\right)$$

$$(e+1)^2 + 2 \cdot e \cdot (e+1) \cdot (x-1) + e \cdot (2 \cdot e+1) \cdot (x-1)^2$$

Kuten on havainnollistettu oikeanpuoleisessa viimeisessä esimerkissä, tuloksen alapuolella olevat series(...)-funktion tuottamat näyttörutiinit voivat järjestää termit uudelleen siten, että dominanttitermi ei ole vasemmanpuoleisin.

Huomaa: Katso myös **dominantTerm()**, sivu 58.

setMode()

setMode(tilaNimiKokonaisluku, asetusKokonaisluku) ⇒ *kokonaisluku*
setMode(lista) ⇒ *kokonaislukulista*

Toimii vain funktiossa tai ohjelmassa.

setMode(tilaNimiKokonaisluku, asetusKokonaisluku) asettaa tilan *tilaNimiKokonaisluku* väliaikaisesti uuteen asetukseen *asetusKokonaisluku* ja määrittää kokonaisluvun, joka vastaa kyseisen tilan alkuperäistä asetusta. Muutos on rajoitettu ohjelman/funktion suorittamisen ajalle.

tilaNimiKokonaisluku määrittää asetettavan tilan. Sen on oltava jokin alla olevan taulukon tilaa kuvaavista kokonaisluvuista.

asetusKokonaisluku määrittää tilan uuden asetuksen. Sen on oltava jokin seuraavista asetettavalle tilalle varatuista asetusta kuvaavista kokonaisluvuista.

setMode(lista)-komennolla voit muuttaa useita asetuksia. *lista* sisältää tilaa ja asetusta kuvaavat kokonaislukuparit. **setMode(lista)** luo samanlaisen listan, jonka kokonaislukuparit kuvaavat alkuperäisiä tiloja ja asetuksia.

Näytä π :n likiarvo käyttäen Näytettävät numerot -tilan oletusasetusta ja näytä sen jälkeen π asetuksella Kiinteä2. Tarkista, että oletusarvo palautuu ohjelman suorittamisen jälkeen.

Define <i>prog1()</i> =Prgm	Done
Disp approx(π)	
setMode(1,16)	
Disp approx(π)	
EndPrgm	
<i>prog1()</i>	
	3.14159
	3.14
	Done

Jos olet tallentanut kaikki tila-asetukset komennolla **getMode(0)** → *muutt*, voit palauttaa nämä asetukset komennolla **setMode(muutt)** aina funktion tai ohjelman sulkemiseen saakka. Katso **getMode()**, sivu 88.

Huomaa: Nykyiset tila-asetukset siirtyvät haettuihin alarutiineihin. Jos jokin alarutiini muuttaa tila-asetusta, tilamuutos häviää, kun ohjaus palautuu hakurutiiniin.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Tilan nimi	Tilaa vastaava kokonaisluku	Asetuksia vastaavat kokonaisluvut
Näytettävät numerot	1	1=Liukuva, 2=Liukuva1, 3=Liukuva2, 4=Liukuva3, 5=Liukuva4, 6=Liukuva5, 7=Liukuva6, 8=Liukuva7, 9=Liukuva8, 10=Liukuva9, 11=Liukuva10, 12=Liukuva11, 13=Liukuva12, 14=Kiinteä0, 15=Kiinteä1, 16=Kiinteä2, 17=Kiinteä3, 18=Kiinteä4, 19=Kiinteä5, 20=Kiinteä6, 21=Kiinteä7, 22=Kiinteä8, 23=Kiinteä9, 24=Kiinteä10, 25=Kiinteä11, 26=Kiinteä12
Kulma	2	1=Radiaani, 2=Aste, 3=Graadi
Eksponenttimuoto	3	1=Normaali, 2=Kymmenpotenssi, 3=Tekninen
Reaali- tai kompleksiluku	4	1=Reaali, 2=Suorakulma, 3=Polaarinen
Automaattinen tai likimääräinen.	5	1=Automaattinen, 2=Likimääräinen, 3=Täsmällinen
Vektorimuoto	6	1=Suorakulma, 2=Sylinteri, 3=Pallo
Kantaluku	7	1=Desimaali, 2=Heksagonaalinen, 3=Binaarinen
Yksikköjärjestelmä	8	1=SI, 2=Eng/US

shift(*Kokonaisluku* [, *SiirtojenLkm*])
 $\Rightarrow$ *kokonaisluku*

Siirtää binaarisen kokonaisluvun bittejä. Voit syöttää *Kokonaisluku*1:n minä tahansa kantelukuna; se muunnetaan automaattisesti etumerkilliseen 64 bitin binaarimuotoon. Jos *Kokonaisluku*1 on liian suuri tälle muodolle, symmetrinen modulo-operaatio sovittaa sen alueelle sopivaksi. Lisätietoja, katso ►**Base2**, sivu 18.

Jos *SiirtojenLkm* on positiivinen, siirto tapahtuu vasemmalle. Jos *SiirtojenLkm* on negatiivinen, siirto tapahtuu oikealle. Oletusarvo on -1 (siirrä yksi bitti oikealle).

Oikealle tapahtuvassa siirrossa oikeanpuoleisin bitti pudotetaan, ja 0 ja 1 lisätään vastaamaan vasemmanpuoleista bittiiä. Vasemmalle tapahtuvassa siirrossa vasemmanpuoleisin bitti pudotetaan, ja 0 lisätään vastaamaan oikeanpuoleisinta bittiiä.

Esimerkki siirrosta oikealle:

Jokainen bitti siirtyy oikealle.

0b0000000000000111101011000011010

Lisää 0:n, jos vasemmanpuoleisin bitti on 0, tai 1:n, jos vasemmanpuoleisin bitti on 1.

tuottaa:

0b00000000000000111101011000011010

Vastaus näkyy kantelukutilan mukaisesti. Alkunollia ei näytetä.

shift(*Listal* [, *SiirtojenLkm*]) $\Rightarrow$ *lista*

Luo *Listal*:n kopion, jota on siirretty oikealle tai vasemmalle *SiirtojenLkm*:n elementtien määrittämän määrän. Ei muuta *Listal*:ä.

Binaarisessa kantelukutilassa:

shift(0b1111010110000110101)	0b111101011000011010
shift(256,1)	0b1000000000

Heksadesimaalisessa kantelukutilassa:

shift(0h78E)	0h3C7
shift(0h78E,-2)	0h1E3
shift(0h78E,2)	0h1E38

Tärkeää: Binaariluvun edelle tulee aina merkitä etumerkki 0b ja heksadesimaaliluvun edelle 0h (nolla, ei O-kirjain).

Desimaalisessa kantelukutilassa:

shift({1,2,3,4})	{undef,1,2,3}
shift({1,2,3,4},-2)	{undef,undef,1,2}
shift({1,2,3,4},2)	{3,4,undef,undef}

Jos *SiirtojenLkm* on positiivinen, siirto tapahtuu vasemmalle. Jos *SiirtojenLkm* on negatiivinen, siirto tapahtuu oikealle. Oletusarvo on -1 (siirrä yksi elementti oikealle).

Elementit, jotka siirto vie *listan* alkuun tai loppuun, asettuvat symboliksi "undef".

shift(Merkkijono1 [,SiirtojenLkm])
 $\Rightarrow$ merkkijono

Luo *Merkkijono1*:n kopion, jota on siirretty oikealle tai vasemmalle *SiirtojenLkm*:n merkkien määrittämän määrän. Ei muuta *Merkkijono1*:ä.

Jos *SiirtojenLkm* on positiivinen, siirto tapahtuu vasemmalle. Jos *SiirtojenLkm* on negatiivinen, siirto tapahtuu oikealle. Oletusarvo on -1 (siirrä yksi merkki oikealle).

Merkit, jotka siirto vie *merkkijonon* alkuun tai loppuun, muuttuvat välilyönneiksi.

shift("abcd")	" abc "
shift("abcd", 2)	" ab "
shift("abcd", 1)	"bcd "

sign() (etumerkki)

sign(Laus1) $\Rightarrow$ lauseke

sign(Lista1) $\Rightarrow$ lista

sign(Matriisi1) $\Rightarrow$ matriisi

Kun kyseessä on reaali- tai kompleksilukulauseke *Laus1*, antaa vastauksena *Laus1/abs(Laus1)*, kun *Laus1* $\neq$ 0.

Vastaus on 1, jos *Laus1* on positiivinen. Vastaus on -1, jos *Laus1* on negatiivinen.

sign(0) edustaa kompleksialueen yksikköpiiriä.

sign(-3.2)	-1.
sign({2,3,4,-5})	{1,1,1,-1}
sign(1+ x)	1

Jos kompleksilukutila on Reaali:

sign($\begin{bmatrix} -3 & 0 & 3 \end{bmatrix}$)	$\begin{bmatrix} -1 & \pm 1 & 1 \end{bmatrix}$
--	--

Kun kyseessä on lista tai matriisi, määrittää kaikkien elementtien etumerkit.

simult()

simult(kerroinMatriisi, vakioVektori[, tol]) ⇒ matriisi

Määrittää sarakevektorin, joka sisältää lineaarisen yhtälöryhmän ratkaisut.

Huomaa: Katso myös **linSolve()**, sivu 107.

kerroinMatriisin on oltava neliömatriisi, joka sisältää yhtälöiden kertoimet.

vakioVektorissa on oltava sama rivimäärä (sama koko) kuin *kerroinMatriisissa* ja sen tulee sisältää vakiot.

Valinnaisesti kaikkia matriisielementtejä käsitellään nollana, jos niiden itseisarvo on pienempi kuin *Tol*. Tätä toleranssia käytetään vain, jos matriisissa on liukulukusyötteitä eikä se sisällä symbolisia muuttujia, joille ei ole määritetty arvoa. Muussa tapauksessa *Tol*-komentoa ei huomioida.

- Jos asetat **Automaattinen tai likimääräinen** -tilan valintaan Approximate (Likimääräinen), laskut suoritetaan liukulukuaritmetiikalla.
- Jos *Tol* jätetään pois tai sitä ei käytetä, oletusarvoinen toleranssi lasketaan seuraavasti:
 $5E-14 \cdot \max(\dim(\text{kerroinMatriisi})) \cdot \text{rowNorm}(\text{kerroinMatriisi})$

simult(kerroinMatriisi, vakioMatriisi[, tol]) ⇒ matriisi

Ratkaisee lineaarisia yhtälöryhmiä, joissa jokaisessa ryhmässä on samat yhtälöiden kertoimet mutta eri vakiot.

Jokaisen *vakioMatriisin* sarakkeen tulee sisältää jonkin yhtälöryhmän vakiot. Jokainen tulomatriisin sarake sisältää vastaavan ryhmän ratkaisun.

Ratkaise yhtälöstä x ja y :

$$x + 2y = 1$$

$$3x + 4y = -1$$

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

Ratkaisu on $x = -3$ and $y = 2$.

Ratkaise:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{array}{l} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \text{matxI} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\ \text{simult}\left(\text{matxI}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \quad \begin{bmatrix} -(2 \cdot b - d) \\ a \cdot d - b \cdot c \\ 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix} \end{array}$$

Ratkaise:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right) \quad \begin{bmatrix} -3 & -7 \\ 2 & 9 \\ 2 & 2 \end{bmatrix}$$

Ensimmäisessä ryhmässä $x=-3$ ja $y=2$.Toisessa ryhmässä $x=-7$ ja $y=9/2$.

►sin

Laus ►sin

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>sin.

Näyttää *Laus*:n kulman sinin. Tämä on näytön muunnosoperaattori. Sitä voidaan käyttää vain syöterivin lopussa.

►sin alentaa kaikkia lausekkeen $\cos(\dots)$ modulo $1-\sin(\dots)^2$ potensseja, siten että jäljelle jäävien lausekkeen $\sin(\dots)$ potenssien eksponentit ovat alueella (0, 2). Tulos ei täten sisällä lauseketta $\cos(\dots)$, jos ja vain jos $\cos(\dots)$ esiintyy lausekkeessa korotettuna vain parillisiin potensseihin.

Huomaa: Tätä muunnosoperaattoria ei tueta aste- eikä graadikulmatilassa. Ennen kuin käytät sitä, varmista, että kulmatila on asetettu radiaaneiksi ja että *Laus* ei sisällä eksplisiittisiä viittauksia aste- tai graadikulmiin.

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

sin()

sin(Laus I) ⇒ lauseke

Astekulmatilassa:

sin(Lista I) ⇒ lista

sin(Laus I) määrittää argumentin sinin lausekkeena.

sin(Lista I) määrittää listan, joka sisältää *Listal*:n kaikkien elementin sinit.

Huomaa: Argumentti tulkitaan aste-, graadi- tai radiaanikulmaksi käytössä olevan kulmatilan mukaisesti. Voit ohittaa kulmatila-asetuksen väliaikaisesti käyttämällä merkintää $^{\circ}$, $^{\text{G}}$ tai $^{\text{r}}$.

$\sin\left(\frac{\pi}{4}\right)$	$\frac{\sqrt{2}}{2}$
$\sin(45)$	$\frac{\sqrt{2}}{2}$
$\sin(\{0,60,90\})$	$\left\{0, \frac{\sqrt{3}}{2}, 1\right\}$

Graadikulmatilassa:

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

Radiaanikulmatilassa:

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

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

sin(neliömatriisi I) ⇒ neliömatriisi

Määrittää *neliömatriisi I*:n matriisin sinin. Tämä ei ole sama kuin kunkin elementin sinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisi I:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Radiaanikulmatilassa:

$$\sin \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

sin⁻¹(Laus I) ⇒ lauseke

sin⁻¹(Lista I) ⇒ lista

sin⁻¹(Laus I) määrittää lausekkeena kulman, jonka sini on *Laus I*.

sin⁻¹(Lista I) määrittää listan, joka sisältää *Lista I*:n kaikkien elementtien käänteissinit.

Huomaa: Vastaus lasketaan aste-, graadi- tai radiaanikulmana käytössä olevan kulmatila-asetuksen mukaisesti.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arcsin (...)**.

sin⁻¹(neliömatriisi I) ⇒ neliömatriisi

Laskee *neliömatriisi I*:n matriisin käänteissinin. Tämä ei ole sama kuin kunkin elementin käänteissinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

Astekulmatilassa:

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

Graadikulmatilassa:

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

Radiaanikulmatilassa:

$$\sin^{-1}(\{0,0.2,0.5\}) \quad \{0,0.201358,0.523599\}$$

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

$$\sin^{-1} \begin{pmatrix} 1 & 5 \\ 4 & 2 \end{pmatrix} \begin{bmatrix} -0.174533-0.12198 \cdot i & 1.74533-2.35591 \cdot i \\ 1.39626-1.88473 \cdot i & 0.174533-0.593162 \cdot i \end{bmatrix}$$

$\sin^{-1}()$ **trig -painike**

neliömatriisi l :n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

 $\sinh()$ **Katalogi > ** **$\sinh(Laus\ l) \Rightarrow lauseke$** $\sinh(1.2)$ 1.50946 **$\sinh(Lista\ l) \Rightarrow lista$** $\sinh(\{0, 1.2, 3.\})$ $\{0, 1.50946, 10.0179\}$

$\sinh(Laus\ l)$ laskee argumentin hyperbolisen kosinin lausekkeena.

$\sinh(Lista\ l)$ määrittää listan, joka sisältää $Listal$:n jokaisen elementin hyperbolisen sinin.

 $\sinh(neliömatriisi\ l) \Rightarrow neliömatriisi$

Laskee *neliömatriisi* l :n matriisin hyperbolisen sinin. Tämä ei ole sama kuin kunkin elementin hyperbolisen sinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **$\cos()$** .

neliömatriisi l :n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Radiaanikulmatilassa:

$$\sinh \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$
 $\sinh^{-1}()$ **Katalogi > ** **$\sinh^{-1}(Laus\ l) \Rightarrow lauseke$** $\sinh^{-1}(0)$ 0 **$\sinh^{-1}(Lista\ l) \Rightarrow lista$** $\sinh^{-1}(\{0, 2.1, 3.\})$ $\{0, 1.48748, \sinh^{-1}(3)\}$

$\sinh^{-1}(Laus\ l)$ laskee argumentin käänteisen hyperbolisen sinin lausekkeena.

$\sinh^{-1}(Lista\ l)$ määrittää listan, joka sisältää $Listal$:n jokaisen elementin käänteiset hyperboliset sinin.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **$\arcsinh$** (...).

 $\sinh^{-1}(neliömatriisi\ l) \Rightarrow neliömatriisi$

Radiaanikulmatilassa:

Laskee *neliömatrissi* I :n matriisin käänteisen hyperbolisen sinin. Tämä ei ole sama kuin kunkin elementin käänteisen hyperbolisen sinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

neliömatrissi I :n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

SinReg

SinReg X , Y [, [*Iteraatiot*] [, [*Jakso*] [, [*Luokka*, [*Sisällytyä*]]

Laskee siniregression listoista X ja Y . Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällytyä*-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Iteraatiot on arvo, joka määrittää ratkaisun yrityskertojen (1-16) enimmäismäärän. Mikäli sitä ei määritetä, oletuksena käytetään arvoa 8. Suuremmilla arvoilla saadaan tyypillisesti parempi tarkkuus, mutta suoritus aika on pitempi ja päin vastoin.

Jakso määrittää arvioidun jakson. Mikäli sitä ei käytetä, X :n arvojen välisen eron tulisi olla sama ja arvojen tulisi olla peräkkäisessä järjestyksessä. Jos määrität *Jakson*, x :n arvojen väliset erot voivat olla erisuuria.

Luokka on luokkakoodien lista vastaavalle X - ja Y -datalle..

Sisällytyä on yhden tai useamman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

SinReg:n tulos esitetään aina radiaaneina riippumatta kulmatilan asetuksesta.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Regressiokertoimet
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatus <i>Y Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

solve()

solve(Yhtälö, Muutt) ⇒ Boolean lauseke

solve(Yhtälö, Muutt=Arvaus) ⇒ Boolean lauseke

solve(Epäyhtälö, Muutt) ⇒ Boolean lauseke

Laskee yhtälön tai epäyhtälön mahdollisia reaali- ja kompleksilukuratkaisuja muuttujalle *Muutt*. Tavoitteena on tuottaa kaikkien ratkaisujen ehdotuksia. Sellaisia yhtälöitä tai epäyhtälöitä voi kuitenkin esiintyä, joille ratkaisujen määrä on ääretön.

Ratkaisuehdotukset eivät välttämättä ole reaalisia äärellisiä ratkaisuja joillekin arvojen yhdistelmille määrittämättömissä muuttujissa.

Automaattinen tai likimääräinen -tilan Auto (Automaattinen) -asetuksessa tavoitteena on tuottaa täsmällisiä ratkaisuja, jos ne ovat tiiviitä, ja näitä lisätään iteratiivisten hakujen avulla likiarvoaritmetiikalla, jos täsmälliset ratkaisut ovat epäkäytännöllisiä.

$$\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x)$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a} \text{ or } x = \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a}$$

$$\text{Ans} | a=1 \text{ and } b=1 \text{ and } c=1$$

$$x = \frac{-1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ or } x = \frac{-1}{2} - \frac{\sqrt{3}}{2} \cdot i$$

$$\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x)$$

$$x=a \text{ or } x=-0.567143$$

Koska suurin yhteinen tekijä peruutetaan oletusarvoisesti suhdelukujen osoittajasta ja nimittäjästä, ratkaisut voivat olla ratkaisuja, jotka ovat ainoastaan yhden tai kummankin puolen rajalla.

Täsmälliset ratkaisut ovat epätodennäköisiä tyyppiä $\geq$, $\leq$, $<$ tai $>$ oleville epäyhtälöille, ellei epäyhtälö ole lineaarinen ja sisältää ainoastaan muuttujan *Muutt*.

Exact (Täsmällinen) -tilassa osat, joita ei voida ratkaista, annetaan implisiittisenä yhtälönä tai epäyhtälönä.

Käytä rajoittavaa ("|")-operaattoria rajoittaaksesi ratkaisuväliä ja/tai muita yhtälössä tai epäyhtälössä esiintyviä muuttujia. Kun löydät ratkaisun yhdeltä väliltä, voit poistaa tämän välin seuraavista hakutoiminnoista epäyhtälöoperaattoreiden avulla.

jos reaali-lukuratkaisuja ei löydy, vastauksen totuusarvoksi määritetään epätosi. vastauksen totuusarvoksi tulee tosi, jos **solve()** pystyy määrittämään, että mikä tahansa muuttujan *Muutt* äärellinen reaaliarvo sopii yhtälöön tai epäyhtälöön.

Koska **solve()** tuottaa aina Boolean vastauksen, komentojen "and", "or" ja "not" avulla voit yhdistää **solve()**-funktion tuloksia toisiinsa tai muihin Boolean lausekkeisiin.

Ratkaisut voivat sisältää uuden yksilöllisen määrittämättömän vakion, joka on muotoa n_j , jossa j on kokonaisluku väliltä 1-255. Tällaiset muuttujat tarkoittavat mielivaltaista kokonaislukua.

Reaali-lukutilassa murtolukupotenssit, joilla on pariton nimittäjä, viittaavat ainoastaan reaali-lukualueeseen. Muussa tapauksessa haaralausekkeet, kuten murtolukupotenssit, logaritmit ja käänteiset trigonometriset funktiot, viittaavat vain päähaaraan. Näin ollen **solve()** tuottaa vain ratkaisuja, jotka vastaavat tätä yhtä reaali-luku- tai päähaaraa.

$$(x+1) \cdot \frac{x-1}{x-1} + x - 3 \quad 2 \cdot x - 2$$

$$\text{solve}(5 \cdot x - 2 \geq 2 \cdot x, x) \quad x \geq \frac{2}{3}$$

$$\text{exact}(\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x)) \quad e^x + x = 0 \text{ or } x = a$$

Radiaani-kulmatilassa:

$$\text{solve}\left(\tan(x) = \frac{1}{x}, x\right) | x > 0 \text{ and } x < 1 \quad x = 0.860334$$

$$\begin{array}{ll} \text{solve}(x = x + 1, x) & \text{false} \\ \text{solve}(x = x, x) & \text{true} \end{array}$$

$$2 \cdot x - 1 \leq 1 \text{ and } \text{solve}(x^2 \neq 9, x) \quad x \neq -3 \text{ and } x \leq 1$$

Radiaani-kulmatilassa:

$$\text{solve}(\sin(x) = 0, x) \quad x = n \cdot \pi$$

$$\begin{array}{ll} \text{solve}\left(x^{\frac{1}{3}} = -1, x\right) & x = -1 \\ \text{solve}(\sqrt{x} = 2, x) & \text{false} \\ \text{solve}(-\sqrt{x} = 2, x) & x = 4 \end{array}$$

Huomaa: Katso myös **cSolve()**, **cZeros()**, **nSolve()** ja **zeros()**.

solve(*Yht1* and *Yht2* [and...],
MuuttTaiArvaus1,
MuuttTaiArvaus2 [, ...]) $\Rightarrow$ Boolean lauseke

solve(*Yhtälöryhmä*, *MuuttTaiArvaus1*,
MuuttTaiArvaus2 [, ...]) $\Rightarrow$ Boolean lauseke

solve{*Yht1*, *Yht2* [,...]} {*MuuttTaiArvaus1*,
MuuttTaiArvaus2 [, ...]} $\Rightarrow$ Boolean lauseke

Antaa ehdotuksia reaaililukuratkaisuista samanaikaisille algebrallisille yhtälöille, joissa jokainen *MuuttTaiArvaus* määrittää ratkaistavan muuttujan.

Voit erottaa yhtälöt **and**-operaattorilla tai voit syöttää *Yhtälöryhmän* jonkin katalogin mallin avulla. *MuuttTaiArvaus*-argumenttien on oltava sama kuin yhtälöiden lukumäärä. Voit halutessasi määrittää muuttujan ensimmäisen arvauksen. Jokaisen *MuuttTaiArvaus*-komennon on oltava muodossa:

muuttuja

– tai –

muuttuja = *reaaliluku* tai *ei-reaaliluku*

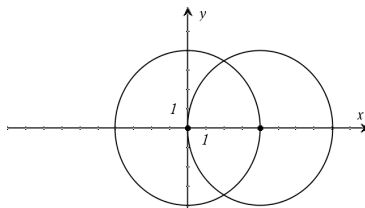
Esimerkiksi *x* kelpaa ja samoin *x*=3.

Jos kaikki yhtälöt ovat polynomeja, ja jos ET määritä ensimmäisiä arvauksia, **solve()** käyttää leksikaalista Gröbner/Buchbergerin eliminaatiomenetelmää yrittäessään määrittää kaikki reaaililukuratkaisut.

Oletetaan esimerkiksi, että origossa on ympyrän säde *r*, ja toinen ympyrän säde *r* on keskitetty kohtaan, jossa ensimmäinen ympyrä leikkaa positiivisen *x*-akselin. Määritä leikkauskohdat **solve()**-funktion avulla.

$$\text{solve}(y=x^2-2 \text{ and } x+2 \cdot y=-1, \{x,y\})$$

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



Kuten on kuvattu merkinnällä r oikealla olevassa esimerkissä, samanaikaisissa polynomiyhtälöissä voi olla ylimääräisiä muuttujia, joilla ei ole arvoja, vaan ne edustavat tiettyjä numeerisia arvoja, jotka voidaan korvata myöhemmin.

Voit ottaa mukaan lisäksi ratkaisumuuttujia (tai käyttää niitä sijalla), joita ei esiinny yhtälöissä. Voit esimerkiksi ottaa mukaan muuttujan z ratkaisumuuttujaksi ulottaaksesi edellisen esimerkin näin säteen r kahteen samansuuntaiseen leikkaavaan sylinteriin.

Sylinteriratkaisut havainnollistavat, miten ratkaisujen sarjat voivat sisältää mielivaltaisia vakioita, jotka ovat muotoa ck , jossa k on kokonaislukuilte väliltä 1-255.

Polynomisarjoissa laskun suoritus aika tai muistin käyttö voivat riippua merkittävästi ratkaisumuuttujien järjestyksestä. Jos ensimmäinen valintasi kuluttaa muistia, tai et jaksa odottaa vastausta, yritä järjestää muuttujat uudelleen yhtälöihin ja/tai *muuttTaiArvaus*-listaan.

Jos et ota mukaan arvauksia, ja jos jokin yhtälöistä on ei-polynominen missä tahansa muuttujassa, mutta kaikki yhtälöt ovat lineaarisia kaikissa ratkaisumuuttujissa, **solve()** käyttää Gaussin eliminointia yrittäessään määrittää kaikkia reaaliuratkaisuja.

Jos yhtälöryhmä ei ole polynominen kaikilta muuttujiltaan eikä lineaarinen ratkaisumuuttujiltaan, **solve()** määrittää korkeintaan yhden ratkaisun käyttäen likimääräistä iteratiivista menetelmää. Tässä ratkaisumuuttujien lukumäärän on oltava sama kuin yhtälöiden lukumäärä, ja kaikkien muiden yhtälöiden sisältämien muuttujien on sievennyttävä luvuiksi.

Jokainen ratkaisumuuttuja alkaa arvausarvostaan, mikäli se on määritetty; muussa tapauksessa se alkaa arvosta 0.0.

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=\frac{-\sqrt{3}\cdot r}{2}$$

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y=$$

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

$$\text{solve}\left(x+e^z\cdot y=1 \text{ and } x-y=\sin(z), \{x,y\}\right)$$

$$x=\frac{e^z\cdot \sin(z)+1}{e^z+1} \text{ and } y=\frac{-(\sin(z)-1)}{e^z+1}$$

$$\text{solve}\left(e^z\cdot y=1 \text{ and } -y=\sin(z), \{y,z\}\right)$$

$$y=2.812\text{E-}10 \text{ and } z=21.9911 \text{ or } y=0.001871$$

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

$$\text{solve}\left(e^z\cdot y=1 \text{ and } -y=\sin(z), \{y,z=2\cdot \pi\}\right)$$

$$y=0.001871 \text{ and } z=6.28131$$

Arvausten avulla voit etsiä lisäratkaisuja yksi kerrallaan. Suppenemista varten arvauksen on mahdollisesti oltava melko lähellä ratkaisua.

SortA

SortA *Listal* [, *Lista2*] [, *Lista3*] ...

SortA *Vektori1* [, *Vektori2*] [, *Vektori3*] ...

Lajittelee ensimmäisen argumentin elementit nousevaan järjestykseen.

Jos otat mukaan lisää argumentteja, lajittelee kunkin argumentin elementit siten, että niiden uudet paikat vastaavat ensimmäisen argumentin elementtien uusia paikkoja.

Kaikkien argumenttien on oltava lista- tai vektorinimiä. Kaikkien argumenttien on oltava samankokoisia.

Ensimmäisen argumentin sisältämät tyhjät elementit siirtyvät alas. Lisätietoja tyhjästä elementeistä, katso sivu 249.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
SortA <i>list1</i>	Done
<i>list1</i>	$\{1,2,3,4\}$
$\{4,3,2,1\} \rightarrow list2$	$\{4,3,2,1\}$
SortA <i>list2</i> , <i>list1</i>	Done
<i>list2</i>	$\{1,2,3,4\}$
<i>list1</i>	$\{4,3,2,1\}$

SortD

SortD *Listal* [, *Lista2*] [, *List3*] ...

SortD *Vektori1* [, *Vektori2*] [, *Vektori3*] ...

Muuten samanlainen kuin **SortA** paitsi, että **SortD** lajittelee elementit laskevaan järjestykseen.

Ensimmäisen argumentin sisältämät tyhjät elementit siirtyvät alas. Lisätietoja tyhjästä elementeistä, katso sivu 249.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
SortD <i>list1</i> , <i>list2</i>	Done
<i>list1</i>	$\{4,3,2,1\}$
<i>list2</i>	$\{3,4,1,2\}$

►Sphere

Vektori ►**Sphere**

Huom: Vastauksen pakottaminen likimääräiseksi:

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>Sphere.

Näyttää rivi- tai sarakevektorin pallonmuotoisena [ρ $\angle\theta$ $\angle\phi$].

Vektorin on oltava kooltaan 3, ja se voi olla rivi- tai sarakevektori.

Huomaa: ►Sphere on näyttömuodon ohje, ei muunnosfunktio. Voit käyttää komentoa ainoastaan syöterivin lopussa.

Kämmenlaite: Paina  .

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

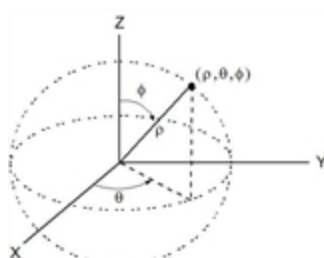
iPad®: Pidä **enter** ja valitse .

$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ ►Sphere
 $\begin{bmatrix} 3.74166 & \angle 1.10715 & \angle 0.640522 \end{bmatrix}$

$\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix}$ ►Sphere
 $\begin{bmatrix} 3.60555 & \angle 0.785398 & \angle 0.588003 \end{bmatrix}$

Paina 

$\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix}$ ►Sphere
 $\begin{bmatrix} \sqrt{13} & \angle \frac{\pi}{4} & \angle \sin^{-1}\left(\frac{2 \cdot \sqrt{13}}{13}\right) \end{bmatrix}$



sqrt()

sqrt(Lausl)⇒lauseke

sqrt(Listal)⇒lista

Laskee argumentin neliöjuuren.

Kun kyseessä on lista, laskee kaikkien Listal:n elementtien neliöjuuret.

Huomaa: Katso myös **Neliöjuurimalli**, sivu 1.

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

stat.results

Näyttää tilastollisen laskutoimituksen tulokset.

Vastaukset näytetään nimiarvoparien sarjana. Näytetyt nimenomaiset nimet riippuvat viimeksi sievennetystä tilastofunktiosta tai komennosta.

Voit kopioida nimen tai arvon ja liittää sen muihin paikkoihin.

Huomaa: Älä määritä muuttujia, joilla on sama nimi kuin tilastoanalyysissä käytettävillä muuttujilla. Joissakin tapauksissa tästä voi olla seurauksena virhetilanne. Tilastoanalyysissä käytettävät muuttujanimet on esitetty alla olevassa taulukossa.

$xlist = \{1, 2, 3, 4, 5\}$	$\{1, 2, 3, 4, 5\}$
$ylist = \{4, 8, 11, 14, 17\}$	$\{4, 8, 11, 14, 17\}$
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.σ _x	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.σ _y	stat.tList
stat.b6	stat.e	stat.MSReg	stat.σ _{x1}	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.σ _{x2}	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σ _x	stat.X̄
stat.b9	stat.FBlock	stat. $\hat{p}$	stat.Σ _x ²	stat.X̄ ₁
stat.b10	stat.Fcol	stat. $\hat{p}_1$	stat.Σ _{xy}	stat.X̄ ₂
stat.bList	stat.FInteract	stat. $\hat{p}_2$	stat.Σ _y	stat.X̄Diff
stat.χ ²	stat.FreqReg	stat. $\hat{p}$ Diff	stat.Σ _y 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.ȳ
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat.ŷ
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat.ŷList

stat.Cupper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CupperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

Huomaa: Aina kun Listat & Taulukot -sovellus laskee tilastolaskujen vastauksia, se kopioi "stat." -ryhmän muuttujat "stat#." -ryhmään, jossa # on luku, jota lisätään automaattisesti. Tällä tavoin voit säilyttää aikaisemmat tulokset suorittaessasi useita laskutoimituksia.

stat.values

Katalogi >

stat.values

Katso esimerkki kohdassa
stat.results.

Näyttää matriisin viimeksi sievennetyille tilastofunktiolle tai -komennolle lasketuista arvoista.

Toisin kuin **stat.results**, **stat.values** jättää pois arvoihin liittyvät nimet.

Voit kopioida arvon ja liittää sen muihin paikkoihin.

stDevPop()

Katalogi >

stDevPop(Lista[, frekvLista]) ⇒ *lauseke*

Laskee Lista:n sisältämien elementtien perusjoukon keskihajonnan.

Jokainen *frekvListan* elementti näyttää Listan vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: Listassa tulee olla vähintään kaksi elementtiä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjästä elementeistä, katso sivu 249.

stDevPop(MatriisiI[, frekvMatriisi])
⇒ *matriisi*

Laskee rivivektorin *MatriisiI*:n sarakkeiden perusjoukon keskihajonnoista.

Radiaanikulma- ja automaattisissa tiloissa:

$$\text{stDevPop}\left(\left\{a, b, c\right\}\right) = \frac{\sqrt{2 \cdot \left(a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2\right)}}{3}$$

$$\text{stDevPop}\left(\left\{1, 2, 5, -6, 3, -2\right\}\right) = \frac{\sqrt{465}}{6}$$

$$\text{stDevPop}\left(\left\{1, 3, 2, 5, -6, 4\right\}, \left\{3, 2, 5\right\}\right) = 4.11107$$

$$\text{stDevPop}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right) = \begin{bmatrix} \frac{4 \cdot \sqrt{6}}{3} & \frac{\sqrt{78}}{3} & \frac{2 \cdot \sqrt{6}}{3} \end{bmatrix}$$

$$\text{stDevPop}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}\right) = \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}$$

$$[2.52608 \quad 5.21506]$$

Jokainen *frekvMatriisin* elementti näyttää *Matriisi1*:n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Matriisi1* :ssä on oltava vähintään kaksi riviä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

stDevSamp()

stDevSamp(Lista[,frekvLista]) ⇒ lauseke

Laskee *Listan* sisältämien elementtien otoksen keskihajonnan.

Jokainen *frekvListan* elementti näyttää *Listan* vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Listassa* tulee olla vähintään kaksi elementtiä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

stDevSamp(Matriisi1[,frekvMatriisi])
⇒ *matriisi*

Laskee rivivektorin *Matriisi1*:n sarakkeiden otosten keskihajonnoista.

Jokainen *frekvMatriisin* elementti näyttää *Matriisi1*:n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Matriisi1* :ssä on oltava vähintään kaksi riviä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

$$\begin{aligned} \text{stDevSamp}(\{a, b, c\}) &= \frac{\sqrt{3 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3} \\ \text{stDevSamp}(\{1, 2, 5, -6, 3, -2\}) &= \frac{\sqrt{62}}{2} \\ \text{stDevSamp}(\{1.3, 2.5, -6.4\}, \{3, 2, 5\}) &= 4.33345 \end{aligned}$$

$$\begin{aligned} \text{stDevSamp}\left(\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{pmatrix}\right) &= \begin{bmatrix} 4 & \sqrt{13} & 2 \end{bmatrix} \\ \text{stDevSamp}\left(\begin{pmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{pmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right) &= \begin{bmatrix} 2.7005 & 5.44695 \end{bmatrix} \end{aligned}$$

Stop	Katalogi > 	
Stop	$i:=0$	0
Ohjelmointikomento: Pysäyttää ohjelman.	Define $progI()$ =Prgm	Done
Stop ei ole sallittu funktioissa.	For $i,1,10,1$	
	If $i=5$	
	Stop	
	EndFor	
	EndPrgm	
Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.	$progI()$	Done
	i	5

Store	Katso → (tallenna), sivu 246.
-------	-------------------------------

string() (merkkijono)	Katalogi > 	
$string(Laus) \Rightarrow merkkijono$	$string(1.2345)$	"1.2345"
Sieventää lausekkeen <i>Laus</i> ja antaa vastauksen merkkijonona.	$string(1+2)$	"3"
	$string(\cos(x)+\sqrt{3})$	"cos(x)+√(3)"

subMat()	Katalogi > 	
$subMat(MatriisiI[, alkurivi] [, alkusarake] [, loppurivi] [, loppusarake]) \Rightarrow matriisi$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
Laskee <i>MatriisiI</i> :n määritetyn alimatriisin.	$subMat(m1,2,1,3,2)$	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
Oletusarvot: <i>alkurivi</i> =1, <i>alkusarake</i> =1, <i>loppurivi</i> =viimeinen rivi, <i>endCol</i> =viimeinen sarake.	$subMat(m1,2,2)$	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)	Katso $\Sigma()$, sivu 236.
-------------	------------------------------

sum()Katalogi > **sum(Lista[, Alku[, Loppu]])**⇒lausekeLaskee *Listan* elementtien summan.*Alku* ja *Loppu* ovat valinnaisia. Ne määrittävät elementtien alueen.Mikä tahansa tyhjä argumentti tuottaa tyhjän vastauksen. *Listassa* olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 249.**sum(MatriisiI[, Alku[, Loppu]])**⇒matriisiLaskee rivivektorin, joka sisältää *MatriisiI*:n sarakkeiden elementtien summat.*Alku* ja *Loppu* ovat valinnaisia. Ne määrittävät rivialueen.Mikä tahansa tyhjä argumentti tuottaa tyhjän vastauksen. *MatriisiI*:ssä olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 249.

$\text{sum}\{\{1,2,3,4,5\}\}$	15
$\text{sum}\{\{a,2\cdot a,3\cdot a\}\}$	$6\cdot a$
$\text{sum}(\text{seq}(n,n,1,10))$	55
$\text{sum}\{\{1,3,5,7,9\},3\}$	21

$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},2,3\right)$	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

sumIf()Katalogi > **sumIf(Lista,Kriteerit[, SummaLista])**
⇒arvoLaskee kaikkien niiden *Listan* sisältämien elementtien kumuloituneen summan, jotka vastaavat määritettyjä kriteereitä *Kriteerit*. Voit halutessasi antaa kumuloivat elementit määrittämällä vaihtoehdoisen listan, *summaLista*.*List*a voi olla lauseke, lista tai matriisi. *SummaListalla*, mikäli se määritetään, on oltava samat mitat kuin *Listalla*.*Kriteeri* voi olla:

- Arvo, lauseke tai merkkijono. Esimerkiksi **34** kumuloi vain niitä *Listan* elementtejä, jotka sieventyvät arvoon 34.
- Boolean lauseke, joka sisältää symbolin ? kunkin elementin paikanpitäjänä. Esimerkiksi lauseke **?<10** kumuloi vain

$\text{sumIf}(\{1,2,e,3,\pi,4,5,6\},2.5<?<4.5)$	$e+\pi+7$
$\text{sumIf}(\{1,2,3,4\},2<?<5,\{10,20,30,40\})$	70

niitä *Listan* elementtejä, jotka ovat alle 10.

Kun jokin *Listan* elementti vastaa kriteereitä *Kriteerit*, elementti lisätään kumuloituvaan summaan. Jos sisällyttät funktioon *summaListan*, summaan lisätäänkin sen sijaan vastaava *summaListan* elementti.

Listat & Taulukot -sovelluksessa voit käyttää solualueita *Listan* ja *summaListan* tilalla.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

Huomaa: Katso myös **countIf()**, sivu 37.

system(*Yht1* [, *Yht2* [, *Yht3* [, ...]]])

$$\text{solve} \left\{ \begin{array}{l} x+y=0 \\ x-y=8 \end{array} \right. \quad x=4 \text{ and } y=-4$$

system(*Laus1* [, *Laus2* [, *Laus3* [, ...]]])

Laskee yhtälöryhmän listaksi muotoiltuna. Voit luoda yhtälöryhmän myös mallin avulla.

Huomaa: Katso myös **Yhtälöryhmä**, sivu 4.

T

Matriisi $I^T \Rightarrow$ *matriisi*

Laskee *MatriisiI*:n transponoidun liittokompleksimatriisin.

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @t.

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

$$\begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array}^T \quad \begin{array}{|c|c|} \hline a & c \\ \hline b & d \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|} \hline 1+i & 2+i \\ \hline 3+i & 4+i \\ \hline \end{array}^T \quad \begin{array}{|c|c|c|} \hline 1-i & 3-i \\ \hline 2-i & 4-i \\ \hline \end{array}$$

tan() -painike**tan(Laus I)** ⇒ lauseke

Astekulmatilassa:

tan(Lista I) ⇒ lista**tan(Laus I)** laskee argumentin tangentin lausekkeena.**tan(Lista I)** määrittää *Listal*:n kaikkien elementtien tangenttien listan.

Huomaa: Argumentti tulkitaan aste-, graadi- tai radiaanikulmaksi käytössä olevan kulmatilan mukaisesti. Voit ohittaa kulmatila-asetuksen väliaikaisesti käyttämällä merkintää °, ° tai °.

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45)$	1
$\tan(\{0,60,90\})$	$\{0,\sqrt{3},\text{undef}\}$

Graadikulmatilassa:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(50)$	1
$\tan(\{0,50,100\})$	$\{0,1,\text{undef}\}$

Radiaanikulmatilassa:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45^\circ)$	1
$\tan\left(\left\{\pi,\frac{\pi}{3},\pi,\frac{\pi}{4}\right\}\right)$	$\{0,\sqrt{3},0,1\}$

tan(neliömatriisi I) ⇒ neliömatriisi

Laskee *neliömatriisi I*:n matriisin tangentin. Tämä ei ole sama kuin kunkin elementin tangentin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisi I:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Radiaanikulmatilassa:

$\tan\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$	$\begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$
--	--

tan⁻¹() -painike**tan⁻¹(Laus I)** ⇒ lauseke

Astekulmatilassa:

tan⁻¹(Lista I) ⇒ lista

tan⁻¹(Laus I) laskee kulman, jonka tangentti on *Laus I*, määrittäen vastauksen lausekkeena.

$\tan^{-1}(1)$	45
----------------	----

Graadikulmatilassa:

$\tan^{-1}()$

 -painike

$\tan^{-1}(\text{Listal})$ luo listan *Listal*:n jokaisen elementin käänteistangenteista.

Huomaa: Vastaus lasketaan aste-, graadi- tai radiaanikulmana käytössä olevan kulmatila-asetuksen mukaisesti.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **$\arctan(\dots)$** .

$\tan^{-1}(\text{neliömatriisi } l)$ $\Rightarrow$ *neliömatriisi*

Laskee *neliömatriisi l*:n matriisin käänteistangentin. Tämä ei ole sama kuin kunkin elementin käänteisen tangentin laskeminen. Laskentamenetelmä on kuvattu kohdassa **$\cos()$** .

neliömatriisi l:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

$\tan^{-1}(1)$ 50

Radiaanikulmatilassa:

$\tan^{-1}(\{0,0,2,0,5\})$ $\{0,0.197396,0.463648\}$

Radiaanikulmatilassa:

$\tan^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$

-0.083658	1.26629	0.62263
0.748539	0.630015	-0.070012
1.68608	-1.18244	0.455126

tangentLine()

Katalogi > 

tangentLine(Lausl,Muutt,Piste) $\Rightarrow$ *lauseke*

tangentLine(Lausl,Muutt=Piste) $\Rightarrow$ *lauseke*

Määrittää tangenttisuoran *Lausl*:n kuvaamasta käyrästä pisteessä, joka on määritetty kohtaan *Muutt=Piste*.

Varmista, että riippumatonta muuttujaa ei ole määritetty. Esimerkiksi, jos $f_1(x):=5$ and $x:=3$, tällöin **tangentLine(f1(x),x,2)** antaa vastauksen "epätosi".

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

Katalogi > 

tanh(Lausl) $\Rightarrow$ *lauseke*

tanh(Listal) $\Rightarrow$ *lista*

tanh(Lausl) laskee argumentin hyperbolisen tangentin lausekkeena.

tanh(Listal) luo listan *Listal*:n jokaisen elementin hyperbolisista tangenteista.

$\tanh(1.2)$	0.833655
$\tanh(\{0,1\})$	$\{0,\tanh(1)\}$

tanh(neliömatriisi I) ⇒ neliömatriisi

Laskee *neliömatriisi I*:n matriisin hyperbolisen tangentin. Tämä ei ole sama kuin kunkin elementin hyperbolisen tangentin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisi I:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Radiaanikulmatilassa:

$$\tanh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

tanh⁻¹()**tanh⁻¹(Laus I) ⇒ lauseke****tanh⁻¹(Lista I) ⇒ lista**

tanh⁻¹(Laus I) laskee argumentin käänteisen hyperbolisen tangentin lausekkeena.

tanh⁻¹(Lista I) luo listan *Lista I*:n jokaisen elementin käänteisistä hyperbolisista tangenteista.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **arctanh** (...).

tanh⁻¹(neliömatriisi I) ⇒ neliömatriisi

Laskee *neliömatriisi I*:n matriisin käänteisen hyperbolisen tangentin. Tämä ei ole sama kuin kunkin elementin käänteisen hyperbolisen tangentin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisi I:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Suorakulmakompleksimuodossa:

$$\tanh^{-1}(0) \quad 0$$

$$\tanh^{-1}(\{1, 2, 1, 3\})$$

$$\left\{ \text{undef}, 0.518046 - 1.5708 \cdot i, \frac{\ln(2)}{2} - \frac{\pi}{2} \cdot i \right\}$$

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

$$\tanh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.099353 + 0.164058 \cdot i & 0.267834 - 1.4908 \\ -0.087596 - 0.725533 \cdot i & 0.479679 - 0.94730 \\ 0.511463 - 2.08316 \cdot i & -0.878563 + 1.7901 \end{bmatrix}$$

Jos haluat nähdä koko vastauksen, paina ▲ ja siirrä sen jälkeen osoitinta painikkeilla ◀ ja ▶.

taylor()Katalogi > 

taylor(*Laus* *l*, *Muutt*, *Aste* [, *Piste*])
 $\Rightarrow$ lauseke

Laskee pyydetyn Taylorin polynomin. Polynomi sisältää kokonaislukuasteiden ei-nollatermejä nollasta arvoon *Aste* saakka komennossa (*Muutt* miinus *Piste*). **taylor()** antaa vastauksena itsensä, jos tämän asteen työstettyjä potenssisarjoja ei ole, tai jos tässä vaadittaisiin negatiivisia tai murtolukueksponentteja. Käytä substituutiota ja/tai väliaikaista kertomista funktion (*Muutt* miinus *Piste*) potenssilla, kun haluat määrittää yleisempiä potenssisarjoja.

Piste en oletusarvo on nolla, ja se on lavenuspiste.

$$\begin{array}{l} \text{taylor}(e^{\sqrt{x}}, x, 2) \qquad \text{taylor}(e^{\sqrt{x}}, x, 2, 0) \\ \text{taylor}(e^t, t, 4) | t = \sqrt{x} \qquad \frac{3}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1 \\ \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right) \qquad \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right) \\ \text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}, x\right) \\ -x^3 - x^2 - x - \frac{1}{x} - 1 \end{array}$$

tCdf()Katalogi > 

tCdf(*alaraja*, *yläraja*, *df*) $\Rightarrow$ luku, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

Laskee Studentin *t*-todennäköisyysjakauman *alarajan* ja *ylärajan* välillä määritetyille vapausasteelle *df*.

Kun $P(X \leq \text{yläraja})$, aseta *alaraja* = $-\infty$.

tCollect()Katalogi > 

tCollect(*Laus* *l*) $\Rightarrow$ lauseke

Laskee lausekkeen, jossa tulot ja sinien ja kosinien kokonaislukupotenssit on muunnettu useiden kulmien sinien ja kosinien lineaarisiksi kombinaatioksi, kulmien summiksi ja kulmien erotuksiksi. Transformaatio muuntaa trigonometriset polynomit niiden harmonisten lineaarisiksi kombinaatioksi.

$$\begin{array}{l} \text{tCollect}((\cos(\alpha))^2) \qquad \frac{\cos(2 \cdot \alpha) + 1}{2} \\ \text{tCollect}(\sin(\alpha) \cdot \cos(\beta)) \qquad \frac{\sin(\alpha - \beta) + \sin(\alpha + \beta)}{2} \end{array}$$

Joskus **tCollect()** pystyy suorittamaan haluamasi laskutoimituksen, kun oletusarvoinen trigonometrinen sievennys ei siihen pysty. **tCollect()** pyrkii kääntämään funktion **tExpand()** aikaansaamat transformaatiot. Joskus **tExpand()**-funktion soveltaminen **tCollect()**-funktion tulokseen, tai päin vastoin, kahdessa eri vaiheessa yksinkertaistaa lauseketta.

tExpand()

tExpand(*Laus l*) $\Rightarrow$ *lauseke*

Laskee lausekkeen, jossa kokonaislukuja olevien monikulmien sinit ja kosinit, kulmien summat ja kulmien erotukset on laajennettu. Identtisen yhtälön $(\sin(x))^2 + (\cos(x))^2 = 1$ vuoksi mahdollisia ekvivalenttisia vastauksia on useita. Tämän vuoksi jokin vastaus voi olla erilainen kuin muissa julkaisuissa esitetty vastaus.

Joskus **tExpand()** pystyy suorittamaan haluamasi laskutoimituksen, kun oletusarvoinen trigonometrinen sievennys ei siihen pysty. **tExpand()** pyrkii kääntämään funktion **tCollect()** aikaansaamat transformaatiot. Joskus **tCollect()**-funktion soveltaminen **tExpand()**-funktion tulokseen, tai päin vastoin, kahdessa eri vaiheessa yksinkertaistaa lauseketta.

Huomaa: Astetilan skaalaus arvolla $\pi/180$ häiritsee **tExpand()**-funktion kykyä tunnistaa laajennettavia muotoja. Parhaan tuloksen saavuttamiseksi **tExpand()**-funktiota tulee käyttää radiaanikulmatilassa.

$$\begin{aligned} \text{tExpand}(\sin(3 \cdot \phi)) &= 4 \cdot \sin(\phi) \cdot (\cos(\phi))^2 - \sin(\phi) \\ \text{tExpand}(\cos(\alpha - \beta)) &= \cos(\alpha) \cdot \cos(\beta) + \sin(\alpha) \cdot \sin(\beta) \end{aligned}$$

Text

Text*kehotemerkkijono*[, *Näytälippu*]

Ohjelmointikomento: Keskeyttää ohjelman ja näyttää merkkijonon *kehotemerkkijono* valintaruudussa.

Määritä ohjelma, joka keskeytyy ja näyttää kunkin viidestä satunnaisluvusta valintaruudussa.

Kun käyttäjä valitsee **OK**-näppäimen, ohjelman suoritus jatkuu.

Valinnainen *lippu*-argumentti voi olla mikä tahansa lauseke.

- Jos *NäytäLippu* jätetään pois, tai jos se sievennyy arvoksi **1**, tekstimuotoinen viesti lisätään laskimen historiaan.
- Jos *NäytäLippu* sievennyy arvoon **0**, tekstimuotoista viestiä ei lisätä historiaan.

Jos ohjelma vaatii käyttäjän kirjoittaman vastauksen, katso **Request**, sivu 157, tai **RequestStr**, sivu 158.

Huomaa: Tätä komentoa voi käyttää käyttäjän määrittämän ohjelman sisällä mutta ei funktion sisällä.

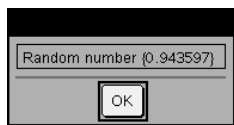
Paina mallin Prgm...EndPrgm jokaisen rivin lopussa näppäintä  näppäimen  sijaan. Tietokoneen näppäimistöllä **Alt**-näppäintä pidetään alhaalla ja painetaan **Enter**.

```
Define text_demo()=Prgm
  For i,1,5
    string:= "Random number " &
string(rand(i))
    Text string
  EndFor
EndPrgm
```

Suorita ohjelma:

text_demo()

Esimerkki yhdestä valintaruudusta:



TInterval *Lista[,Frekv[,CTaso]]*

(Datalistan syöte)

TInterval $\bar{x}, S_x, n[,CTaso]$

(Yhteenvetotilaston syöte)

Laskee *t*-luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Tuntemattoman perusjoukon keskiarvon luottamusväli
stat. $\bar{x}$	Satunnaisesta normaalijakaumasta olevan datasekvenssin otoksen keskiarvo
stat.ME	Virhemarginaali
stat.df	Vapausasteet
stat. σ_x	Otoksen keskihajonta
stat.n	Otoksen keskiarvon sisältävän datasekvenssin pituus

TInterval_2Samp

TInterval_2Samp *List1,Lista2[,Frekv1
[,Frekv2[,CTaso[,Poolaus]]]]*

(Datalistan syöte)

TInterval_2Samp $\bar{x}1, Sx1, n1, \bar{x}2, Sx2, n2$
[,CTaso[,Poolaus]]

(Yhteenvetotilaston syöte)

Laskee kahden otoksen t -luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Poolaus=1 poolaa varianssit; *Poolaus=0* ei poolaa variansseja.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Luottamusväli, joka sisältää jakauman luottamusvälin todennäköisyyden
stat. $\bar{x}1 - \bar{x}2$	Satunnaisesta normaalijakaumasta olevien datasekvenssien otosten keskiarvot

Tulosmuuttuja	Kuvaus
stat.ME	Virhemarginaali
stat.df	Vapausasteet
stat. $\bar{x}$ 1, stat. $\bar{x}$ 2	Satunnaisesta normaalijakaumasta olevien datasekvenssien otosten keskiarvot
stat. σ x1, stat. σ x2	<i>Lista 1:n</i> ja <i>Lista 2:n</i> otosten keskihajonnat
stat.n1, stat.n2	Otosten lukumäärä datasekvensseissä
stat.sp	Poolattu keskihajonta. Laskettu, kun <i>Poolaus</i> = KYLLÄ

tmpCnv()

Katalogi > 

tmpCnv(*Laus* °*lämpYksikkö*, _
°*lämpYksikkö*2)
⇒*lauseke* _°*lämpYksikkö*2

Muuntaa *Laus*:n määrittämän lämpötila-
arvon yksiköstä toiseen. Kelpaavat
lämpötilayksiköt ovat:

°CCelsius
°FFahrenheit
°KKelvin
°RRankine

Asteen merkin ° saat Katalogin
symboleista.

alaviivan _ voi syöttää painamalla  .

Esimerkiksi 100_°C muuntuu Fahrenheit-
asteiksi 212_°F.

Jos haluat muuttaa lämpötila-alueen, käytä
sen sijaan funktiota **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

Huomaa: Lämpötilan yksiköt voi valita
Katalogista.

ΔtmpCnv()

Katalogi > 

ΔtmpCnv(*Laus* °*lämpYksikkö*, _
°*lämpYksikkö*2)
⇒*lauseke* _°*lämpYksikkö*2

Huomaa: Voit syöttää tämän funktion
näppäimistöltä kirjoittamalla
deltaTmpCnv (...).

Merkin Δ saat Katalogin symboleista.

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

Δ tmpCnv()

Katalogi >

Muuttaa *Laus*:n määrittämän lämpötila-alueen (kahden lämpötila-arvon välisen eron) yksiköstä toiseen. Kelpaavat lämpötilayksiköt ovat:

_°Celsius
_°FFahrenheit
_°KKelvin
_°RRankine

Voit syöttää merkin ° valitsemalla sen symbolipaletista tai kirjoittamalla @d.

alaviivan _ voi syöttää painamalla .

1_°C ja 1_°K ovat samansuuruksia, samoin kuin 1_°F ja 1_R. 1_°C on kuitenkin 9/5 Fahrenheit-asteen arvosta 1_°F.

Esimerkiksi 100_°C:n alue (0_°C - 100_°C) vastaa Fahrenheit-asteiden aluetta 180_°F.

Jos haluat muuttaa jonkin lämpötila-arvon alueen sijaan, käytä funktiota **tmpCnv()**.

Huomaa: Lämpötilan yksiköt voi valita Katalogista.

tPdf()

Katalogi >

tPdf(*XArvo*,*df*) $\Rightarrow$ *luku*, jos *XArvo* on luku,
lista, jos *XArvo* on lista

Laskee todennäköisyysfunktio (pdf) Studentin *t*-jakaumalle määritetyllä *x*:n arvolla ja määritetyillä vapausasteilla *df*.

trace()

Katalogi >

trace(*neliömatriisi*) $\Rightarrow$ *lauseke*

Laskee *neliömatriisin* jäljityksen (päälävistäjän kaikkien elementtien summan).

$\text{trace}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}\right)$	15
$\text{trace}\left(\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}\right)$	$2 \cdot a$

```
Try
    lohko1
Else
    lohko2
EndTry
```

Suorittaa *lohko1*:n, ellei virhettä esiinny. Ohjelman suoritus siirtyy *lohko2*:een, jos *lohko1*:ssä esiintyy virhe. Järjestelmän muuttuja *errCode* sisältää virhekoodin, jotta ohjelma voi korjata virheen. Virhekoodien luettelo on esitetty kohdassa Virhekoodit ja viestit, sivu 256.

lohko1 ja *lohko2* voivat olla joko yksi lauseke tai sarja lausekkeita, jotka on erotettu toisistaan kaksoispisteellä (:).

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Esimerkki 2

Jos halaut nähdä kommentojen **Try**, **ClrErr** ja **PassErr** toiminnan, syötä oikealla näkyvä *eigenvals()*-ohjelma. Suorita ohjelma suorittamalla kukin seuraavista lausekkeista.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

$$\text{eigenvals}\left(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

Huomaa: Katso myös **ClrErr**, sivu 26, ja **PassErr**, sivu 138.

```
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

Define *eigenvals*(a,b)=Prgm

© Program *eigenvals*(A,B) displays eigenvalues of 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

tTest**tTest** $\mu 0, Lista[, Frekv[, Hypot]]$

(Datalistan syöte)

tTest $\mu 0, \bar{x}, sx, n, [Hypot]$

(Yhteenvetotilaston syöte)

Testaa hypoteesia yhden tuntemattoman perusjoukon keskiarvoon μ , kun perusjoukon keskihajontaa σ ei tunneta. Tulosten yhteen veto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Testaa $H_0: \mu = \mu 0$, jonkin seuraavan vaihtoehdon suhteen:

Kun $H: \mu < \mu 0$, aseta *Hypot*<0

Kun $H^a: \mu \neq \mu 0$ (oletus), aseta *Hypot*=0

Kun $H^a: \mu > \mu 0$, aseta *Hypot*>0

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.t	$(\bar{x} - \mu 0) / (\text{stdev} / \text{sqrt}(n))$
stat.PVal	Alin merkitsevyytaso, jolla nolahypoteesi voidaan hylätä
stat.df	Vapausasteet
stat. $\bar{x}$	<i>Listan</i> sisältämän datasekvenssin otoksen keskiarvo
stat.sx	Datasekvenssin otoksen keskihajonta
stat.n	Otoksen koko

tTest_2Samp**tTest_2Samp** *List1, List2[, Frekv1[, Frekv2[, Hypot[, Poolaus]]]]*

(Datalistan syöte)

tTest_2Samp $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2[, Hypot$
 $[, Poolaus]$

(Yhteenvetotilaston syöte)

Suorittaa kahden otoksen t -testin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Testaa $H_0: \mu_1 = \mu_2$, jonkin seuraavan vaihtoehdon suhteen:

Kun $H_0: \mu_1 < \mu_2$, aseta *Hypot*<0

Kun $H_a: \mu_1 \neq \mu_2$ (oletus), aseta *Hypot*=0

Kun $H_a: \mu_1 > \mu_2$, aseta *Hypot*>0

Poolaus=1 poolaa varianssit

Poolaus=0 ei poolaa variansseja

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.t	Keskiarvojen erotukselle laskettu vakio-ohjearvo
stat.PVal	Alin merkitsevyytaso, jolla nollahypoteesi voidaan hylätä
stat.df	t-tilaston vapausasteet
stat. $\bar{x}1$, stat. $\bar{x}2$	<i>List1</i> 1:n ja <i>List2</i> 2:n sisältämien datasekvenssien otosten keskiarvot
stat.sx1, stat.sx2	<i>List1</i> 1:n ja <i>List2</i> 2:n sisältämien datasekvenssien otosten keskihajonnat
stat.n1, stat.n2	Otosten koko
stat.sp	Poolattu keskihajonta. Laskettu, kun <i>Poolaus</i> =1

tvmFV()

tvmFV(*N, I, PV, Pmt, [PpY], [CpY], [PmtAt]*)
 $\Rightarrow arvo$

tvmFV(120,5,0,-500,12,12)

77641.1

Talouselaskentafunktio, joka laskee rahan tulevan arvon.

tvmFV()Katalogi > 

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 205. Katso myös **amortTbl()**, sivu 8.

tvmI()Katalogi > 

tvmI(*N,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*)
 $\Rightarrow arvo$

tvmI(240,100000,-1000,0,12,12)	10.5241
--------------------------------	---------

Talouselaskentafunktio, joka laskee vuosikoron.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 205. Katso myös **amortTbl()**, sivu 8.

tvmN()Katalogi > 

tvmN(*I,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*)
 $\Rightarrow arvo$

tvmN(5,0,-500,77641,12,12)	120.
----------------------------	------

Talouselaskentafunktio, joka laskee maksuerien lukumäärän.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 205. Katso myös **amortTbl()**, sivu 8.

tvmPmt()Katalogi > 

tvmPmt(*N,I,PV,FV,[PpY],[CpY],[PmtAt]*)
 $\Rightarrow arvo$

tvmPmt(60,4,30000,0,12,12)	-552.496
----------------------------	----------

Talouselaskentafunktio, joka laskee jokaisen maksuerän määrän.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 205. Katso myös **amortTbl()**, sivu 8.

tvmPV($N, I, Pmt, FV, [PpY], [CpY], [PmtAt]$)
 $\Rightarrow arvo$

tvmPV(48,4,-500,30000,12,12) -3426.7

Talouslaskentafunktio, joka laskee nykyarvon.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 205. Katso myös **amortTbl** (), sivu 8.

TVM-argumentti*	Kuvaus	Datatyyppi
N	Maksuerien lukumäärä	reaaliluku
I	Vuosikorko	reaaliluku
PV	Nykyarvo	reaaliluku
Pmt	Maksun määrä	reaaliluku
FV	Tuleva arvo	reaaliluku
PpY	Maksuerien määrä vuodessa, oletusarvo=1	kokonaisluku > 0
CpY	Korkojaksoja vuodessa, oletusarvo=1	kokonaisluku > 0
$PmtAt$	Erääntyvän maksun määrä kunkin jakson lopussa tai alussa, oletusarvo=loppu	kokonaisluku (0=loppu, 1=alku)

* Nämä rahan aika-arvon argumenttien nimet ovat samat kuin TVM-muuttujien nimet (kuten **tvm.pv** ja **tvm.pmt**), joita käytetään *Laskin*-sovelluksen talouslaskentatoiminnossa. Talouslaskentafunktioiden argumenttien arvot tai vastaukset eivät kuitenkaan tallennu TVM-muuttujiin.

TwoVar $X, Y, [Frekv] [, Luokka, Sisällyttä]$

Laskee kahden muuttujan tilastot. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Kaikkien listojen on oltava samankokoisia *Sisällyttä*-listaa lukuunottamatta.

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Frekv on valinnainen frekvenssiarvojen lista. Jokainen *Frekv*:n elementti määrittää kunkin vastaavan datapisteen X ja Y esiintymisfrekvenssin. Oletusarvo on 1. Kaikkien elementtien on oltava kokonaislukuja 0.

Luokka on luokkakoodien lista vastaavalle X - ja Y -datalle.

Sisällytä on yhden tai useamman luokkakoodin lista. Vain ne datayksiköt, joiden luokkakoodi sisältyy tähän listaan, ovat mukana laskutoimituksessa.

Tyhjä elementti jossakin listassa X , *Frekv* tai *Luokka* saa aikaan, että kaikkien listojen vastaava elementti on tyhjä. Tyhjä elementti jossakin listassa $X1$ - $X20$ saa aikaan, että kaikkien listojen vastaava elementti on tyhjä. Lisätietoja tyhjästä elementeistä, katso sivu 249.

Tulosmuuttuja	Kuvaus
stat. $\bar{X}$	x :n arvojen keskiarvo
stat. x	x :n arvojen summa
stat. $x2$	$x2$:n arvojen summa
stat. sx	x :n otoksen keskihajonta
stat. x	x :n perusjoukon keskihajonta
stat. n	Datapisteiden lukumäärä
stat. $\bar{Y}$	y :n arvojen keskiarvo
stat. Σy	y :n arvojen summa
stat. Σy^2	$y2$:n arvojen summa
stat. sy	y :n otoksen keskihajonta
stat. σy	y :n perusjoukon keskihajonta
stat. Σxy	$x \cdot y$ -arvojen summa
stat. r	Korrelaatiokerroin
stat. MinX	x :n arvojen minimi

unLock

Katalogi > 

unLock*Muutt1* [, *Muutt2*] [, *Muutt3*] ...
unLock*Muutt*.

Vapauttaa määritetyt muuttujat tai muuttujaryhmän. Lukittuja muuttujia ei voi muokata eikä poistaa.

Katso **Lock**, sivu 111, ja **getLockInfo()**, sivu 87.

<i>a</i> :=65	65
Lock <i>a</i>	Done
getLockInfo(<i>a</i>)	1
<i>a</i> :=75	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	Done
<i>a</i> :=75	75
DelVar <i>a</i>	Done

V

varPop()

Katalogi > 

varPop(*List* [, *frekvList*])⇒*lauseke*

Laskee *List*an perusjoukon varianssin.

Jokainen *frekvList*an elementti näyttää *List*an vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *List*assa tulee olla vähintään kaksi elementtiä.

Jos jokin elementti jommassakummassa listassa on tyhjä, kyseistä elementtiä ei huomioida, eikä toisessa listassa olevaa vastaavaa elementtiä myöskään huomioida. Lisätietoja tyhjästä elementeistä, katso sivu 249.

varPop({ 5,10,15,20,25,30 })	875
	12
Ans: 1.	72.9167

varSamp()

Katalogi > 

varSamp(*List* [, *frekvList*])⇒*lauseke*

Laskee *List*an otosten varianssin.

Jokainen *frekvList*an elementti näyttää *List*an vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *List*assa tulee olla vähintään kaksi elementtiä.

varSamp({ { <i>a</i> , <i>b</i> , <i>c</i> } })	$\frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$
varSamp({ { 1,2,5 }, { 6,3,-2 } })	31
	2
varSamp({ { 1,3,5 }, { 4,6,2 } })	68
	33

Jos jokin elementti jommassakummassa listassa on tyhjä, kyseistä elementtiä ei huomioida, eikä toisessa listassa olevaa vastaavaa elementtiä myöskään huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

varSamp(MatriisiI[,frekvMatriisi])
 $\Rightarrow$ *matriisi*

Laskee rivivektorin, joka sisältää *MatriisiI*:n kaikkien sarakkeiden otoksen varianssin.

Jokainen *frekvMatriisi*n elementti näyttää *MatriisiI*:n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Jos jokin elementti jommassakummassa matriisissa on tyhjä, kyseistä elementtiä ei huomioida, eikä toisessa matriisissa olevaa vastaavaa elementtiä myöskään huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 249.

Huomaa: *MatriisiI*:ssä on oltava vähintään kaksi elementtiä.

varSamp	$\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{pmatrix}$	$[4.75 \ 1.03 \ 4]$
varSamp	$\begin{pmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{pmatrix} \cdot \begin{pmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{pmatrix}$	$[3.91731 \ 2.08411]$

W

Wait

Luettelo >

Wait aikaSekunteina

Keskeyttää toiminnon *aikaSekunteina* sekunnin ajaksi.

Wait on erityisen käyttökelpoinen ohjelmassa, jossa tarvitaan lyhyttä viivettä pyydettyjen tietojen saamiseksi käyttöön.

Argumentin *aikaSekunteina* on oltava lausekkeen muodossa, joka yksinkertaistaa desimaaliarvoon välillä 0–100. Komento pyöristää tämän arvon lähimpään 0,1 sekuntiin.

Peruuttaaksesi **Wait** käynnissä olevan,

- **Kämmenlaite:** Pidä -painiketta painettuna ja paina toistuvasti .

Odottaaksesi 4 sekuntia:

Wait 4

Odottaaksesi 1/2 sekuntia:

Wait 0.5

Odottaaksesi 1,3 sekuntia käyttäen muuttujaa *seklukema*:

seklukema:=1.3

Wait seklukema

Tämä esimerkki sytyttää vihreän LED-valon päälle 0,5 sekunnin ajaksi ja sammuttaa sen sitten.

painiketta.

- **Windows®:** Pidä **F12**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **Macintosh®:** Pidä **F5**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **iPad®:** Sovellus näyttää kehotuksen. Voit jatkaa odottamista tai peruuttaa.

Huomaa: Voit käyttää komentoa **Wait** käyttäjän määrittelemän ohjelman sisällä mutta ei funktion sisällä.

Send “SET GREEN 1 ON”
Wait 0.5
Send “SET GREEN 1 OFF”

warnCodes ()

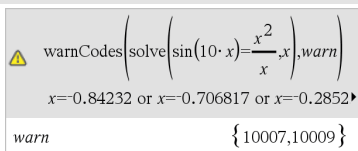
warnCodes(*Expr1*, *StatusVar*) $\Rightarrow$ *expression*

Laskee lausekkeen *Expr1*, antaa tuloksen ja varastoi mahdollisten luotujen varoitusten koodit *StatusVar* -luettelomuuttujaan. Jos varoituksia ei ole luotu, tämä funktio kohdistaa funktiolle *StatusVar* tyhjän luettelon.

Expr1 voi olla mikä tahansa sallittu TI-Nspire™:n tai TI-Nspire™ CAS:n matemaattinen lauseke. Et voi käyttää komentoa tai tehtävää *Expr1*-lausekkeena.

StatusVar:n arvon on oltava sallittu muuttujan nimi.

Katso varoituskoodien ja niihin liittyvien viestien luettelo sivulla sivu 263.



$$\text{warnCodes}\left(\text{solve}\left(\sin(10 \cdot x) = \frac{x^2}{x}, x\right), \text{warn}\right)$$

$$x = -0.84232 \text{ or } x = -0.706817 \text{ or } x = -0.2852$$

warn	{ 10007, 10009 }
------	------------------

Jos haluat nähdä koko vastauksen, paina $\blacktriangle$ ja siirrä sen jälkeen osoitinta painikkeilla $\blacktriangleleft$ ja $\blacktriangleright$.

when()

when(*Ehto*, *tosiTulos* [, *epätosiTulos*][, *tuntematonTulos*])
 $\Rightarrow$ *lauseke*

Määrittää totuusarvon *tosiTulos*, *epätosiTulos* tai *tuntematonTulos* riippuen siitä, onko *Ehto* tosi, epätosi vai tuntematon. Antaa vastauksena syötteen, jos oikean vastauksen määrittämiseen on liian vähän argumentteja.

when()

Katalogi > 

Jätä pois sekä *epätosiTulos* että *tuntematonTulos*, kun haluat, että lauseke määritetään vain alueella, jolla *Ehto* on tosi.

Käytä kommentoa **undef epätosiTulos**, kun haluat määrittää lausekkeen, joka piirtyy vain jollekin välille.

when() on hyödyllinen komento rekursiivisten funktioiden määrittämisessä.

when $\{x < 0, x + 3\} x = 5$	undef
---------------------------------	-------

when $\langle n > 0, n \cdot \text{factorial}(n-1), 1 \rangle \rightarrow \text{factorial}(n)$	<i>Done</i>
$\text{factorial}(3)$	6
$3!$	6

While

Katalogi >

While *Ehto* *Lohko*

EndWhile

Suorittaa *Lohkon* sisältämät lausekkeet,
mikäli *Ehto* on tosi.

Lohko voi olla joko yksi lauseke tai sarja lausekkeita, jotka on erotettu toisistaan kaksoispisteellä (:).

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määrittysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Define $sum_of_recip(n) = \text{Func}$	
Local $i, tempsum$	
$1 \rightarrow i$	
$0 \rightarrow tempsum$	
While $i \leq n$	
$tempsum + \frac{1}{i} \rightarrow tempsum$	
$i + 1 \rightarrow i$	
EndWhile	
Return $tempsum$	
EndFunc	
$sum_of_recip(3)$	$\frac{11}{6}$

x

xorKatalogi > 

```
BooleanLaus1xorBooleanLaus2 antaa
vastauksena Boolean
lausekkeenBooleanList1
xorBooleanList2 antaa vastauksena Boolean
listanBooleanMatriisi1
xorBooleanMatriisi2 antaa vastauksena
Boolean matriisin
```

true xor true	false
5>3 xor 3>5	true

Määrittää totuusarvoksi tosi, jos *BooleanLaus1* on tosi ja *BooleanLaus2* on epätosi, tai päin vastoin.

Määrittää totuusarvoksi epätosi, jos kumpikin argumentti on tosi tai kumpikin on epätosi. Antaa vastauksena sievennetyn Boolean lausekkeen, jos kummankaan argumentin totuusarvoa ei voi määrittää todeksi tai epätodeksi.

Huomaa: Katso **or**, sivu 136.

Kokonaisluku1 xor Kokonaisluku2 $\Rightarrow$ *kokonaisluku*

Vertaa kahta reaalikokonaislukua bitti bitiltä **xor**-operaation avulla. Sisäisesti kumpikin kokonaisluku muunnetaan etumerkilliseksi, 64 bitin binaariluvuksi. Kun vastaavia bittejä verrataan, tulos on 1, jos jompikumpi bitti (mutta ei molemmat) on 1; tulos on 0, jos kumpikin bitti on 0 tai kumpikin bitti on 1. Laskettu arvo edustaa bittituloksia, ja se näkyy kantelukutilan mukaisesti.

Kokonaisluvut voi syöttää minkä tahansa luvun kantelukuna. Binaarisen syötteen edelle tulee merkitä etumerkki 0b ja heksadesimaalisen syötteen edelle 0h. Jos etumerkkiä ei ole, kokonaislukuja käsitellään desimaalilukuina (kantaluksi 10).

Jos syötät desimaalikokonaisluvun, joka on etumerkillisen, 64 bitin binaarimuodon lukualueen ulkopuolella, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle. Lisätietoja, katso **►Base2**, sivu 18.

Huomaa: Katso **or**, sivu 136.

Z

Heksadesimaalisessa kantelukutilassa:

Tärkeää: Nolla, ei O-kirjain.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

Binaarisessa kantelukutilassa:

0b100101 xor 0b100	0b100001
--------------------	----------

Huomaa: Binaarisessa syötteessä voi olla korkeintaan 64 numeroa (etuliitettä 0b ei lasketa). Heksadesimaalisessa syötteessä voi olla korkeintaan 16 numeroa.

zeros()

zeros(Laus, Muutt) $\Rightarrow$ lista

zeros(Laus, Muutt=Arvaus) $\Rightarrow$ lista

$$\text{zeros}\left(a \cdot x^2 + b \cdot x + c, x\right) = \left\{ \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}, \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a} \right\}$$

$$a \cdot x^2 + b \cdot x + c | x = \text{Ans}[2] \quad 0$$

Määrittää listan muuttujan *Muutt* mahdollisista reaalilukuarvoista, joilla $Laus=0$. **zeros()** suorittaa tämän laskemalla yhtälön **explist(solve (Laus=0,Muutt),Muutt)**.

Joissakin tilanteissa funktion **zeros()** vastausmuoto on sopivampi kuin funktion **solve()**. Funktion **zeros()** vastausmuoto ei kuitenkaan pysty ilmaisemaan implisiittisiä ratkaisuja tai ratkaisuja, jotka vaativat epäyhtälöitä, eikä ratkaisuja, joihin ei sisälly muuttujaa *Muutt*.

Huomaa: Katso myös **cSolve()**, **cZeros()** ja **solve()**.

**zeros({Laus1, Laus2},
{muuttTaiArvaus1, muuttTaiArvaus2 [, ...
]})⇒matriisi**

Laskee mahdollisia reaalisia nollakohtia samanaikaisille algebrallisille yhtälöille, joissa jokainen *muuttTaiArvaus* määrittää ratkaistavan tuntemattoman arvon.

Voit halutessasi määrittää muuttujan ensimmäisen arvauksen. Jokaisen *muuttTaiArvaus*-komennon on oltava muodossa:

muuttuja

– tai –

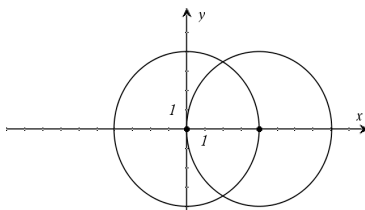
muuttuja = reaalii tai ei-reaali luku

Esimerkiksi x kelpaa ja samoin $x=3$.

Jos kaikki lausekkeet ovat polynomeja, ja ET määritä ensimmäisiä arvauksia, **zeros()** käyttää leksikaalista Gröbner/Buchbergerin eliminaatiomenetelmää yrittäessään määrittää kaikki reaaliset nollakohdat.

Oletetaan esimerkiksi, että origossa on ympyrän säde r , ja toinen ympyrän säde r on keskitetty kohtaan, jossa ensimmäinen ympyrä leikkaa positiivisen x -akselin. Määritä leikkauskohdat **zeros()**-funktion avulla.

$$\begin{array}{l} \text{exact}\left(\text{zeros}\left(a \cdot (e^x + x) \cdot (\text{sign}(x) - 1), x\right)\right) \quad \{\square\} \\ \text{exact}\left(\text{solve}\left(a \cdot (e^x + x) \cdot (\text{sign}(x) - 1) = 0, x\right)\right) \\ e^x + x = 0 \text{ or } x > 0 \text{ or } a = 0 \end{array}$$



Kuten on kuvattu merkinnällä r oikealla olevassa esimerkissä, samanaikaisissa polynomilausekkeissa voi olla ylimääräisiä muuttujia, joilla ei ole arvoja, vaan ne edustavat tiettyjä numeerisia arvoja, jotka voidaan korvata myöhemmin.

Jokainen tulomatriisin rivi edustaa vaihtoehtoista nollakohtaa, jossa komponentit on järjestetty samalla tavalla kuin *MuuttTaiArvaus*-listassa. Jos haluat määrittää rivin juuren, indeksoi matriisi [*riveittäin*].

Voit ottaa mukaan (tai käyttää niitä tilalla) myös tuntemattomia muuttujia, joita ei esiinny lausekkeissa. Voit esimerkiksi ottaa mukaan z :n tuntemattomana muuttujana ulottaaksesi edellisen esimerkin säteen r kahteen samansuuntaiseen leikkaavaan sylinteriin. Sylintereiden nollakohdat havainnollistavat, miten nollakohtien sarjat voivat sisältää mielivaltaisia vakioita, jotka ovat muotoa ck , jossa k on kokonaislukuliite väliltä 1-255.

Polynomisarjoissa laskutoimituksen suoritus aika tai muistin käyttö voivat riippua merkittävästi tuntemattomien muuttujien järjestyksestä. Jos ensimmäinen valintasi kuluttaa muistia, tai et jaksaa odottaa vastausta, yritä järjestää muuttujat uudelleen lausekkeisiin ja/tai *MuuttTaiArvaus*-listaan.

Jos et ota mukaan arvauksia, ja jokin lausekkeista on ei-polynominen missä tahansa muuttujassa, mutta kaikki muuttujat ovat lineaarisia kaikissa tuntemattomissa muuttujissa, **zeros()** käyttää Gaussin eliminointia yrittäessään määrittää kaikki reaaliset nollakohdat.

$$\text{zeros}\left(\left\{x^2+y^2-r^2,(x-r)^2+y^2-r^2\right\},\{x,y\}\right)$$

$$\begin{bmatrix} r & -\sqrt{3}\cdot r \\ 2 & 2 \\ r & \sqrt{3}\cdot r \\ 2 & 2 \end{bmatrix}$$

Määritä rivin 2 juuri:

$$\text{Ans}[2] \quad \begin{bmatrix} r & \sqrt{3}\cdot r \\ 2 & 2 \end{bmatrix}$$

$$\text{zeros}\left(\left\{x^2+y^2-r^2,(x-r)^2+y^2-r^2\right\},\{x,y,z\}\right)$$

$$\begin{bmatrix} r & -\sqrt{3}\cdot r & c1 \\ 2 & 2 & \\ r & \sqrt{3}\cdot r & c1 \\ 2 & 2 & \end{bmatrix}$$

$$\text{zeros}\left(\left\{x+e^z\cdot y-1,x-y-\sin(z)\right\},\{x,y\}\right)$$

$$\begin{bmatrix} e^z\cdot \sin(z)+1 & -(\sin(z)-1) \\ e^z+1 & e^z+1 \end{bmatrix}$$

Jos yhtälöryhmä ei ole polynominen kaikilta muuttujiltaan eikä lineaarinen tuntemattomilta muuttujiltaan, **zeros()** määrittää korkeintaan yhden nollakohdan käyttäen likimääräistä iteratiivista menetelmää. Tässä tuntemattomien muuttujien lukumäärän on oltava sama kuin lausekkeiden lukumäärä, ja kaikkien muiden lausekkeiden sisältämien muuttujien on sievennyttävä luvuiksi.

Jokainen tuntematon muuttuja alkaa arvausarvostaan, mikäli se on määritetty; muussa tapauksessa se alkaa arvosta 0.0.

Arvausten avulla voit etsiä lisää nollakohtia yksi kerrallaan. Suppenemista varten arvauksen on mahdollisesti oltava melko lähellä nollakohtaa.

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \{y, z\}\right)$$

0.041458	3.18306
0.001871	6.28131
4.76E-11	1796.99
2.E-13	254.469

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \{y, z = 2 \cdot \pi\}\right)$$

0.001871	6.28131
----------	---------

zInterval

zInterval $\sigma, \text{Lista}, \text{Frekv}, \text{CTaso}$

(Datalistan syöte)

zInterval $\sigma, \bar{x}, n, \text{CTaso}$

(Yhteenvetotilaston syöte)

Laskee z -luottamusvälin. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Tuntemattoman perusjoukon keskiarvon luottamusväli
stat. $\bar{x}$	Satunnaisesta normaalijakaumasta olevan datasekvenssin otoksen keskiarvo
stat.ME	Virhemarginaali
stat.sx	Otoksen keskihajonta

Tulosmuuttuja	Kuvaus
stat.n	Otoksen keskiarvon sisältävän datasekvenssin pituus
stat. σ	Datasekvenssin <i>Lista</i> tunnettu perusjoukon keskihajonta

zInterval_1Prop

Katalogi > 

zInterval_1Prop $x, n [, CTaso]$

Laskee yhden osuuden z -luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

x on ei-negatiivinen kokonaisluku.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Luottamusväli, joka sisältää jakauman luottamusvälin todennäköisyyden
stat. $\hat{p}$	Laskettu onnistumisten osuus
stat.ME	Virhemarginaali
stat.n	Otosten lukumäärä datasekvenssissä

zInterval_2Prop

Katalogi > 

zInterval_2Prop $x1, n1, x2, n2 [, CTaso]$

Laskee kahden osuuden z -luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

$x1$ ja $x2$ ovat ei-negatiivisia kokonaislukuja.

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Luottamusväli, joka sisältää jakauman luottamusvälin todennäköisyyden
stat. $\hat{p}$ Diff	Osuuksien välinen laskettu erotus

Tulosmuuttuja	Kuvaus
stat.ME	Virhemarginaali
stat. $\hat{p}1$	Arvio ensimmäisen näytteen osuudesta
stat. $\hat{p}2$	Arvio toisen näytteen osuudesta
stat.n1	Otoksen koko datasekvenssissä 1
stat.n2	Otoksen koko datasekvenssissä 2

zInterval_2Samp

Katalogi > 

zInterval_2Samp $\sigma_1, \sigma_2, Lista1, Lista2$
 $[, Frekv1[, Frekv2[, CTaso]]]$

(Datalistan syöte)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2[, CTaso]$

(Yhteenvetotilaston syöte)

Laskee kahden näytteen z-luottamusvälin.
Tulosten yhteenvedo tallentuu *stat.results*-
muuttujaan. (Katso sivu 186.)

Lisätietoja listassa olevien tyhjen
elementtien vaikutuksesta, katso Tyhjät
elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Luottamusväli, joka sisältää jakauman luottamusvälin todennäköisyyden
stat. $\bar{x}1$ - $\bar{x}2$	Satunnaisesta normaalijakaumasta olevien datasekvenssien otosten keskiarvot
stat.ME	Virhemarginaali
stat. $\bar{x}1$, stat. $\bar{x}2$	Satunnaisesta normaalijakaumasta olevien datasekvenssien otosten keskiarvot
stat. $\sigma x1$, stat. $\sigma x2$	<i>Lista 1</i> :n ja <i>Lista 2</i> :n otosten keskihajonnat
stat.n1, stat.n2	Otosten lukumäärä datasekvensseissä
stat.r1, stat.r2	Datasekvenssien <i>Lista 1</i> ja <i>Lista 2</i> tunnetut perusjoukon keskihajonnat

zTest $\mu_0, \sigma, \text{Lista}, [\text{Frekv}], [\text{Hypot}]$

(Datalistan syöte)

zTest $\mu_0, \sigma, \bar{x}, n, [\text{Hypot}]$

(Yhteenvetotilaston syöte)

Suorittaa z-testin frekvenssillä *frekvlista*.
Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Testaa $H_0: \mu = \mu_0$, jonkin seuraavan vaihtoehdon suhteen:

Kun $H_a: \mu < \mu_0$, aseta *Hypot*<0

Kun $H_a: \mu \neq \mu_0$ (oletus), aseta *Hypot*=0

Kun $H_a: \mu > \mu_0$, aseta *Hypot*>0

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$
stat.P Value	Pienin todennäköisyys, jolla nollahypoteesi voidaan hylätä
stat. $\bar{x}$	<i>Lista</i> n sisältämän datasekvenssin otoksen keskiarvo
stat.sx	Datasekvenssin otoksen keskihajonta. Lasketaan vain <i>Data</i> -syötteelle.
stat.n	Otoksen koko

zTest_1Prop**zTest_1Prop** $p_0, x, n, [\text{Hypot}]$

Laskee yhden osuuden z-testin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

x on ei-negatiivinen kokonaisluku.

Testaa $H_0: p = p_0$ jonkin seuraavan vaihtoehdon suhteen:

Kun $H_a: p > p_0$, aseta *Hypot*>0

Kun $H_a: p \neq p_0$ (oletus), aseta *Hypot*=0

Kun $H_a: p < p_0$, aseta *Hypot*<0

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.p0	Perusjoukon osuuden hypoteesiarvo
stat.z	Osuudelle laskettu vakio-ohjearvo
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat. $\hat{p}$	Arvioitu otoksen osuus
stat.n	Otoksen koko

zTest_2Prop $x1, n1, x2, n2, [Hypot]$

Laskee kahden osuuden z-testin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

$x1$ ja $x2$ ovat ei-negatiivisia kokonaislukuja.

Testaa $H_0: p1 = p2$ jonkin seuraavan vaihtoehdon suhteen:

Kun $H_0: p1 > p2$, aseta *Hypot*=0

Kun $H_a: p1 \neq p2$ (oletus), aseta *Hypot*=0

Kun $H_a: p < p0$, aseta *Hypot*<0

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.z	Osuuksien erotukselle laskettu vakio-ohjearvo
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat. $\hat{p}_1$	Arvio ensimmäisen näytteen osuudesta
stat. $\hat{p}_2$	Arvio toisen näytteen osuudesta
stat. $\hat{p}$	Poolattu arvio otoksen osuudesta
stat.n1, stat.n2	Kokeissa 1 ja 2 otettujen otosten lukumäärä

zTest_2Samp $\sigma_1, \sigma_2, Lista1, Lista2[, Frekv1$
 $[, Frekv2[, Hypot]]]$

(Datalistan syöte)

zTest_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2[, Hypot]$

(Yhteenvetotilaston syöte)

Laskee kahden otoksen z -testin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 186.)

Testaa $H_0: \mu_1 = \mu_2$, jonkin seuraavan vaihtoehdon suhteen:

Kun $H: \mu_1 < \mu_2$, aseta *Hypot*<0

Kun $H^a: \mu_1 \neq \mu_2$ (oletus), aseta *Hypot*=0

Kun $H^a_a: \mu_1 > \mu_2$, *Hypot*>0

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 249.

Tulosmuuttuja	Kuvaus
stat.z	Keskiarvojen erotukselle laskettu vakio-ohjearvo
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat. $\bar{x}1$, stat. $\bar{x}2$	<i>Lista1</i> :n ja <i>Lista2</i> :n sisältämien datasekvenssien otosten keskiarvot
stat.sx1, stat.sx2	<i>Lista1</i> :n ja <i>Lista2</i> :n sisältämien datasekvenssien otosten keskihajonnat
stat.n1, stat.n2	Otosten koko

Symbolit

+ (yhteenlasku)		+ painike
$Laus1 + Laus2 \Rightarrow lauseke$	56	56
Laskee kahden argumentin summan.	$56+4$	60
	$60+4$	64
	$64+4$	68
	$68+4$	72
$Listal + Lista2 \Rightarrow lista$ $Matriisi2 \Rightarrow matriisi$	$\left\{22,\pi,\frac{\pi}{2}\right\} \rightarrow I1$	$\left\{22,\pi,\frac{\pi}{2}\right\}$
Määrittää listan (tai matriisin), joka sisältää $Listal$:n ja $Lista2$:n (tai $Matriisi1$:n ja $Matriisi2$:n) vastaavien elementtien summat.	$\left\{10,5,\frac{\pi}{2}\right\} \rightarrow I2$	$\left\{10,5,\frac{\pi}{2}\right\}$
	$I1+I2$	$\{32,\pi+5,\pi\}$
	$Ans+\{\pi,5,\pi\}$	$\{\pi+32,\pi,0\}$
	$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$
$Laus + Listal \Rightarrow lista$	$15+\{10,15,20\}$	$\{25,30,35\}$
$Listal + Laus \Rightarrow lista$	$\{10,15,20\}+15$	$\{25,30,35\}$
Määrittää listan, joka sisältää $Laus$:n ja $Listal$:n kunkin elementin summat.		
$Laus + Matriisi1 \Rightarrow matriisi$		
$Matriisi1 + Laus \Rightarrow matriisi$	$20+\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
Laskee matriisin, jossa $Laus$ on lisätty jokaiseen elementtiin $Matriisi1$:n diagonaalimatriisissa. $Matriisi1$:n on oltava neliö.		
Huomaa: Käytä merkintää $+$ (piste plus), kun haluat lisätä lausekkeen jokaiseen elementtiin.		

- (vähennyslasku)		- painike
$Laus1 - Laus2 \Rightarrow lauseke$	$6-2$	4
Laskee $Laus1$ miinus $Laus2$.	$\pi-\frac{\pi}{6}$	$\frac{5\cdot\pi}{6}$

–(vähennyslasku)

 painike

$Listal - Lista2 \Rightarrow lista$

$$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\} = \left\{ 12, \pi - 5, 0 \right\}$$

$Matriisi1 - Matriisi2 \Rightarrow matriisi$

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

Vähentää $Listal$:n (tai $Matriisi2$:n) jokaisen elementin $Listal$:n (tai $Matriisi1$:n) vastaavasta elementistä ja antaa tuloksena vastaukset.

Argumenttien tulee olla mitoiltaan samanlaisia.

$Laus - Listal \Rightarrow lista$

$Listal - Laus \Rightarrow lista$

$$15 - \{ 10, 15, 20 \} = \{ 5, 0, -5 \}$$

$$\{ 10, 15, 20 \} - 15 = \{ -5, 0, 5 \}$$

Vähentää jokaisen $Listal$:n elementin lausekkeesta $Laus$ tai vähentää lausekkeen $Laus$ jokaisesta $Listal$:n elementistä ja antaa vastauksena tuloslistan.

$Laus - Matriisi1 \Rightarrow matriisi$

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

$Matriisi1 - Laus \Rightarrow matriisi$

$Laus - Matriisi1$ laskee matriisin $Laus$ kertaa identtinen matriisi miinus $Matriisi1$. $Matriisi1$:n on oltava neliö.

$Matriisi1 - Laus$ laskee lausekkeen $Laus$ matriisin kerrottuna identtisellä matriisilla, joka on vähennetty $Matriisi1$:stä. $Matriisi1$:n on oltava neliö.

Huomaa: Käytä merkintää $.-$ (piste miinus), kun haluat vähentää lausekkeen jokaisesta elementistä.

·(kertolasku)

 painike

$Laus1 \cdot Laus2 \Rightarrow lauseke$

$$2 \cdot 3 \cdot 4 \cdot 5 = 6 \cdot 9$$

Laskee kahden argumentin tulon.

$$x \cdot y \cdot x = x^2 \cdot y$$

$Listal \cdot Lista2 \Rightarrow lista$

$$\{ 1, 2, 3 \} \cdot \{ 4, 5, 6 \} = \{ 4, 10, 18 \}$$

Luo listan, joka sisältää $Listal$:n ja $Lista2$:n vastaavien elementtien tulot.

$$\left\{ \frac{2}{a}, \frac{3}{2} \right\} \cdot \left\{ a^2, \frac{b}{3} \right\} = \left\{ 2 \cdot a, \frac{b}{2} \right\}$$

Listojen tulee olla mitoiltaan samanlaisia.

·(kertolasku)

 painike

Matriisi1 · *Matriisi2* ⇒ *matriisi*

Laskee *Matriisi1*:n ja *Matriisi2*:n matriisitulon.

Matriisi1:n sarakkeiden lukumäärän on oltava sama kuin *Matriisi2*:n rivien lukumäärä.

Laus · *Listal* ⇒ *lista*

Listal · *Laus* ⇒ *lista*

Määrittää listan, joka sisältää lausekkeen *Laus* ja kunkin *Listal*:n elementin tulon.

Laus · *Matriisi1* ⇒ *matriisiMatriisi1*
· *Laus* ⇒ *matriisi*

Määrittää matriisin, joka sisältää lausekkeen *Laus* ja kunkin *Matriisi1*:n elementin tulon.

Huomaa: Käytä merkintää · (piste kerro), kun haluat kertoa lausekkeen jokaisella elementillä.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} = \begin{bmatrix} a+2 \cdot b+3 \cdot c & d+2 \cdot e+3 \cdot f \\ 4 \cdot a+5 \cdot b+6 \cdot c & 4 \cdot d+5 \cdot e+6 \cdot f \end{bmatrix}$$

$$\pi \cdot \{4,5,6\} = \{4 \cdot \pi, 5 \cdot \pi, 6 \cdot \pi\}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 = \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$\lambda \cdot \text{identity}(3) = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

/ (jakolasku)

 painike

Laus1 / *Laus2* ⇒ *lauseke*

Laskee osamäärän *Laus1* jaettuna *Laus2*:lla.

Huomaa: Katso myös **Murtolukumalli**, sivu 1.

Listal / *Listal* ⇒ *lista*

Määrittää listan, joka sisältää osamäärät laskutoimituksista *Listal* jaettuna *Listal*:lla.

Listojen tulee olla mitoiltaan samanlaisia.

Laus / *Listal* ⇒ *lista*

Listal / *Laus* ⇒ *lista*

Määrittää listan, joka sisältää osamäärät laskutoimituksista *Laus* jaettuna *Listal*:llä tai *Listal* jaettuna lausekkeella *Laus*.

$$\frac{2}{3.45} = .57971$$

$$\frac{x^3}{x} = x^2$$

$$\frac{\{1.,2,3\}}{\{4,5,6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

$$\frac{a}{\{3,a,\sqrt{a}\}} = \left\{\frac{a}{3}, 1, \sqrt{a}\right\}$$

$$\frac{\{a,b,c\}}{a \cdot b \cdot c} = \left\{\frac{1}{b \cdot c}, \frac{1}{a \cdot c}, \frac{1}{a \cdot b}\right\}$$

Matriisi1 / *Laus* $\Rightarrow$ *matriisi*

Laskee matriisin, joka sisältää osamäärät laskutoimituksesta *Matriisi1*/*Laus*.

$$\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{a \cdot b \cdot c} \quad \frac{\begin{bmatrix} 1 & 1 & 1 \\ b \cdot c & a \cdot c & a \cdot b \end{bmatrix}}$$

Huomaa: Käytä merkintää . / (piste jaa), kun haluat jakaa lausekkeen jokaisella elementillä.

Laus1 ^ *Laus2* $\Rightarrow$ *lauseke*

List1 ^ *List2* $\Rightarrow$ *lista*

Laskee ensimmäisen argumentin korotettuna toisen argumentin potenssiin.

Huomaa: Katso myös **Eksponenttimalli**, sivu 1.

Jos kyseessä on lista, laskee *List1*:n elementit korotettuna *List2*:n vastaavien elementtien potenssiin.

Reaalilukujen alueella murtolukupotenssit, joilla on supistetut eksponentit ja parittomat nimittäjät, käyttävät reaalista aluetta versus pääalue kompleksitilassa.

Laus ^ *List1* $\Rightarrow$ *lista*

Laskee lausekkeen *Laus* korotettuna *List1*:n elementtien potenssiin.

List1 ^ *Laus* $\Rightarrow$ *lista*

Laskee *List1*:n elementit korotettuna *Laus*:n potenssiin.

neliömatrissi1 ^ *kokonaisluku* $\Rightarrow$ *matriisi*

Laskee *neliömatrissi1*:n korotettuna kokonaisluvun *kokonaisluku* potenssiin.

neliömatrissi1:n on oltava neliömatrissi.

Jos *kokonaisluku* = -1, laskee käänteismatriisin.

Jos *kokonaisluku* < -1, laskee käänteismatriisin korotettuna sopivaan positiiviseen potenssiin.

$$\frac{4^2}{16} \quad \frac{\{a,2,c\}^{\{1,b,3\}}}{\{a,2^b,c^3\}}$$

$$\frac{p^{\{a,2,-3\}}}{\left\{p^a, p^2, \frac{1}{p^3}\right\}}$$

$$\frac{\{1,2,3,4\}^{-2}}{\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}}$$

$$\frac{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2}{\begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}}$$

$$\frac{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1}}{\begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}}$$

$$\frac{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2}}{\begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}}$$

x² (neliö)**x² painike***Laus*l² ⇒ lauseke

4^2	16
-------	----

Laskee argumentin neliön.

$\{2,4,6\}^2$	$\{4,16,36\}$
---------------	---------------

*Listal*l² ⇒ lista

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$	$\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$
---	--

Laskee listan, joka sisältää *Listal*:n elementtien neliöt.*neliömatrissi*l² ⇒ *matrissi*

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \cdot ^\wedge 2$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$
---	--

Laskee *neliömatrissi*l:n matriisinelion. Tämä ei ole sama kuin kunkin elementin neliön laskeminen. Käytä merkintää $\cdot^\wedge 2$, kun haluat laskea jokaisen elementin neliön.

.+ (piste lisää)**.+ painikkeet***Matriisi*1 .+ *Matriisi*2 ⇒ *matrissi*

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$
---	--

Laus .+ *Matriisi*1 ⇒ *matrissi*

$x \cdot + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$
--	--

*Matriisi*1 .+ *Matriisi*2 laskee matriisin, joka on *Matriisi*1:n ja *Matriisi*2:n vastaavien elementtiparien summa.

Laus .+ *Matriisi*1 laskee matriisin, joka on *Laus*:n ja kunkin *Matriisi*1:n elementin summa.

.- (piste-erotus)**.- painikkeet***Matriisi*1 .- *Matriisi*2 ⇒ *matrissi*

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$
---	--

Laus .-*Matriisi*1 ⇒ *matrissi*

$x \cdot - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$
--	--

*Matriisi*1 .-*Matriisi*2 laskee matriisin, joka on *Matriisi*1:n ja *Matriisi*2:n vastaavien elementtiparien välinen erotus.

Laus .-*Matriisi*1 laskee matriisin, joka on *Laus*:n ja kunkin *Matriisi*1:n elementin erotus.

. · (pistetulo)

  painikkeet

Matriisi1 . · *Matriisi2* $\Rightarrow$ *matriisi*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$$

Laus . · *Matriisi1* $\Rightarrow$ *matriisi*

$$x \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$$

Matriisi1 . · *Matriisi2* laskee matriisin, joka on *Matriisi1*:n ja *Matriisi2*:n vastaavien elementtiparien tulo.

Laus . · *Matriisi1* laskee matriisin, joka sisältää *Laus*:n ja kunkin *Matriisi1*:n elementin tulot.

. / (pisteosamäärä)

  painikkeet

Matriisi1 . / *Matriisi2* $\Rightarrow$ *matriisi*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \left(\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \right) = \begin{bmatrix} \frac{a}{c} & \frac{1}{2} \\ \frac{b}{5} & \frac{3}{d} \end{bmatrix}$$

Laus . / *Matriisi1* $\Rightarrow$ *matriisi*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \left(\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \right) = \begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$$

Matriisi1 . / *Matriisi2* laskee matriisin, joka on *Matriisi1*:n ja *Matriisi2*:n vastaavien elementtiparien osamäärä.

Laus . / *Matriisi1* laskee matriisin, joka on lausekkeen *Laus* ja kunkin *Matriisi1*:n elementin osamäärä.

. ^ (pistepotenssi)

  painikkeet

Matriisi1 . ^ *Matriisi2* $\Rightarrow$ *matriisi*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot ^{\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}} = \begin{bmatrix} a^c & 16 \\ b^5 & 3^d \end{bmatrix}$$

Laus . ^ *Matriisi1* $\Rightarrow$ *matriisi*

$$x \cdot ^{\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}} = \begin{bmatrix} x^c & x^4 \\ x^5 & x^d \end{bmatrix}$$

Matriisi1 . ^ *Matriisi2* laskee matriisin, jossa *Matriisi2*:n jokainen elementti on eksponentti *Matriisi1*:n vastaavalle elementille.

Laus . ^ *Matriisi1* laskee matriisin, jossa *Matriisi1*:n jokainen elementti on eksponentti lausekkeelle *Laus*.

- (negaatio)

 **painike**

$-Laus1 \Rightarrow lauseke$

$-Listal \Rightarrow lista$

$-Matriisi1 \Rightarrow matriisi$

Laskee argumentin negaation.

Kun kyseessä on lista tai matriisi, määrittää kaikkien elementtien negaatiot.

Jos argumentti on binaarinen tai heksadesimaalinen kokonaisluku, negaatio antaa kakkosen komplementin.

-2.43	-2.43
$-\{-1,0.4,1.2\}$	$\{1,-0.4,-1.2\}$
$-a \cdot -b$	$a \cdot b$

Binaarisessa kantelukutilassa:

```
-0b100101
0b11111111111111111111111111111111▶
```

Jos haluat nähdä koko vastauksen, paina **▶** ja siirrä sen jälkeen osoitinta painikkeilla **◀** ja **▶**.

% (prosentti)

  **painikkeet**

$Laus1 \% \Rightarrow lauseke$

$Listal \% \Rightarrow lista$

$Matriisi1 \% \Rightarrow matriisi$

argument

Määrittää 100

Kun kyseessä on lista tai matriisi, määrittää listan tai matriisin, jossa jokainen elementti on jaettu luvulla 100.

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina  .

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

iPad®: Pidä **enter** ja valitse .

13%	0.13
$\{\{1,10,100\}\}\%$	$\{0.01,0.1,1.\}$

= (on yhtä kuin)

 **painike**

$Laus1 = Laus2 \Rightarrow Boolean\ lauseke$

$Listal = Lista2 \Rightarrow Boolean\ lista$

$Matriisi1 = Matriisi2 \Rightarrow Boolean\ matriisi$

Antaa totuusarvon tosi, jos $Laus1$ määritetään olevan yhtä kuin $Laus2$.

Antaa totuusarvon epätosi, jos $Laus1$ määritetään olevan ei yhtä kuin $Laus2$.

Kaikissa muissa tapauksissa vastauksena on yhtälön sievennetty muoto.

Esimerkkifunktio, jossa on käytetty matematiikkatestisymboleita: $=$, $\neq$, $<$, $\leq$, $>$, $\geq$

= (on yhtä kuin)

painike

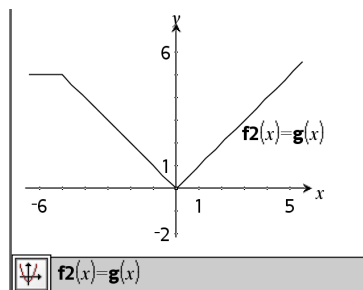
Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

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

Done

Funktion $g(x)$ kuvaajan piirtämisen tulos



≠ (ei yhtä kuin)

painikkeet

$Laus1 \neq Laus2 \Rightarrow \text{Boolean lauseke}$

$List1 \neq List2 \Rightarrow \text{Boolean lista}$

$Matriisi1 \neq Matriisi2 \Rightarrow \text{Boolean matriisi}$

Antaa totuusarvon tosi, jos $Laus1$ määritetään olevan ei yhtä kuin $Laus2$.

Antaa totuusarvon epätosi, jos $Laus1$ määritetään olevan yhtä kuin $Laus2$.

Kaikissa muissa tapauksissa vastauksena on yhtälön sievennetty muoto.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Katso esimerkki kohdasta “=” (on yhtä kuin).

$\neq$ (ei yhtä kuin)

  painikkeet

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla $\neq$

$<$ (pienempi kuin)

  painikkeet

$Laus1 < Laus2 \Rightarrow \text{Boolean lauseke}$

Katso esimerkki kohdasta “=” (on yhtä kuin).

$List1 < List2 \Rightarrow \text{Boolean lista}$

$Matriisi1 < Matriisi2 \Rightarrow \text{Boolean matriisi}$

Antaa totuusarvon tosi, jos $Laus1$ määritetään olevan pienempi kuin $Laus2$.

Antaa totuusarvon epätosi, jos $Laus1$ määritetään olevan suurempi tai yhtä suuri kuin $Laus2$.

Kaikissa muissa tapauksissa vastauksena on yhtälön sievennetty muoto.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

$\leq$ (pienempi tai yhtä suuri kuin)

  painikkeet

$Laus1 \leq Laus2 \Rightarrow \text{Boolean lauseke}$

Katso esimerkki kohdasta “=” (on yhtä kuin).

$List1 \leq List2 \Rightarrow \text{Boolean lista}$

$Matriisi1 \leq Matriisi2 \Rightarrow \text{Boolean matriisi}$

Antaa totuusarvon tosi, jos $Laus1$ määritetään olevan pienempi tai yhtä suuri kuin $Laus2$.

Antaa totuusarvon epätosi, jos $Laus1$ määritetään olevan suurempi kuin $Laus2$.

Kaikissa muissa tapauksissa vastauksena on yhtälön sievennetty muoto.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla $\leq$

> (suurempi kuin)

  painikkeet

$Laus1 > Laus2 \Rightarrow \text{Boolean lauseke}$

Katso esimerkki kohdasta “=” (on yhtä kuin).

$List1 > List2 \Rightarrow \text{Boolean lista}$

$Matriisi1 > Matriisi2 \Rightarrow \text{Boolean matriisi}$

Antaa totuusarvon tosi, jos *Laus1* määritetään olevan suurempi kuin *Laus2*.

Antaa totuusarvon epätosi, jos *Laus1* määritetään olevan pienempi tai yhtä suuri kuin *Laus2*.

Kaikissa muissa tapauksissa vastauksena on yhtälön sievennetty muoto.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

$\geq$ (suurempi tai yhtä suuri kuin)

  painikkeet

$Laus1 \geq Laus2 \Rightarrow \text{Boolean lauseke}$

Katso esimerkki kohdasta “=” (on yhtä kuin).

$List1 \geq List2 \Rightarrow \text{Boolean lista}$

$Matriisi1 \geq Matriisi2 \Rightarrow \text{Boolean matriisi}$

Antaa totuusarvon tosi, jos *Laus1* määritetään olevan suurempi tai yhtä suuri kuin *Laus2*.

Antaa totuusarvon epätosi, jos *Laus1* määritetään olevan pienempi kuin *Laus2*.

Kaikissa muissa tapauksissa vastauksena on yhtälön sievennetty muoto.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla $>=$

⇒ (looginen seuraus)

ctrl [= näppäimet

BooleanLaus1 ⇒ *BooleanLaus2* antaa vastauksena *Boolean lausekkeen*

$5 > 3$ or $3 > 5$ true

BooleanList1 ⇒ *BooleanList2* antaa vastauksena *Boolean listan*

$5 > 3$ ⇒ $3 > 5$ false

3 or 4 7

BooleanMatriisi1 ⇒ *BooleanMatriisi2* antaa vastauksena *Boolean matriisin*

3 ⇒ 4 -4

$\{1, 2, 3\}$ or $\{3, 2, 1\}$ $\{3, 2, 3\}$

$\{1, 2, 3\}$ ⇒ $\{3, 2, 1\}$ $\{-1, -1, -3\}$

Kokonaisluku1 ⇒ *Kokonaisluku2* antaa vastauksena *kokonaisluvun*

Arvioi lausekkeen not <argumentti1> or <argumentti2> ja antaa vastauksena tosi, epätosi tai yhtälön sievennetyn muodon.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla =>

⇔ (looginen kaksoisseuraus, XNOR)

ctrl [= näppäimet

BooleanLaus1 ⇔ *BooleanLaus2* antaa vastauksena *Boolean lausekkeen*

$5 > 3$ xor $3 > 5$ true

BooleanList1 ⇔ *BooleanList2* antaa vastauksena *Boolean listan*

$5 > 3$ ⇔ $3 > 5$ false

3 xor 4 7

BooleanMatriisi1 ⇔ *BooleanMatriisi2* antaa vastauksena *Boolean matriisin*

3 ⇔ 4 -8

$\{1, 2, 3\}$ xor $\{3, 2, 1\}$ $\{2, 0, 2\}$

$\{1, 2, 3\}$ ⇔ $\{3, 2, 1\}$ $\{-3, -1, -3\}$

Kokonaisluku1 ⇔ *Kokonaisluku2* antaa vastauksena *kokonaisluvun*

Antaa vastauksena XOR Boolean operaation negaation kahdesta argumentista. Antaa vastauksena totuusarvon tosi, epätosi tai yhtälön sievennetyn muodon.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla <=>

! (kertoma)

 **painike**

Laus! $\Rightarrow$ *lauseke*

5! 120

List! $\Rightarrow$ *lista*

$\{\{5,4,3\}\}!$ $\{120,24,6\}$

Matriisi! $\Rightarrow$ *matriisi*

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

Määrittää argumentin kertoman.

Jos kyseessä on lista tai matriisi, määrittää listan tai matriisin elementtien kertomista.

& (liitä)

  **painikkeet**

*Merkkijono*1 & *Merkkijono*2 $\Rightarrow$ *merkkijono*

"Hello "&"Nick" "Hello Nick"

Antaa vastauksena tekstimerkkijonon, joka on *Merkkijono*2 liitettynä *Merkkijono*1:een.

d() (derivaatta)

Katalogi > 

d(*Laus*1, *Muutt*[, *Aste*]) $\Rightarrow$ *lauseke*

$\frac{d}{dx}(f(x) \cdot g(x))$ $\frac{d}{dx}(f(x)) \cdot g(x) + \frac{d}{dx}(g(x)) \cdot f(x)$

d(*List*1, *Muutt*[, *Aste*]) $\Rightarrow$ *lista*

$\frac{d}{dy}\left(\frac{d}{dx}(x^2 \cdot y^3)\right)$ $6 \cdot y^2 \cdot x$

d(*Matriisi*1, *Muutt*[, *Aste*]) $\Rightarrow$ *matriisi*

$\frac{d}{dx}\left(\left\{x^2, x^3, x^4\right\}\right)$ $\left\{2 \cdot x, 3 \cdot x^2, 4 \cdot x^3\right\}$

Laskee ensimmäisen argumentin ensimmäisen derivaatan muuttujan *Muutt* suhteen.

Asteen, mikäli se otetaan mukaan, on oltava kokonaisluku. Jos aste on pienempi kuin nolla, vastaus on antiderivaatta.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **derivative** (...).

d() ei noudata normaalia sievennysmekanismia, jossa argumentit sievennetään täydellisesti, minkä jälkeen sovitetaan funktion määritelmää näihin täydellisesti sievennetyihin argumentteihin. Sen sijaan funktio **d()** suorittaa seuraavat vaiheet:

1. Sievennä toista argumenttia vain sen verran, että sen tuloksena ei ole ei-muuttuja.
2. Sievennä ensimmäistä argumenttia vain sen verran, ettei se hae mitään tallennettua arvoa vaiheessa 1 määritetyille muuttujalle.
3. Määritä vaiheen 2 tuloksen symbolinen derivaatta vaiheessa 1 saadun muuttujan suhteen.

Jos vaiheen 1 muuttujalla on tallennettu arvo tai rajoittavalla operaattorilla ("|") määritetty arvo, korvaa tämä arvo vastaukseen vaiheesta 3.

Huomaa: Katso myös

Ensimmäinen derivaatta, sivu 6,

Toinen derivaatta, sivu 6 tai

N:s derivaatta, sivu sivu 6.

f() (integraali)

$f(\text{Laus } l, \text{Muutt}[, Ala, Ylä]) \Rightarrow \text{lauseke}$

$f(\text{Laus } l, \text{Muutt}[, Vakio]) \Rightarrow \text{lauseke}$

Laskee *Laus l*:n integraalin muuttujan *Muutt* suhteen välillä *Ala* - *Ylä* olevista arvoista.

Huomaa: Katso myös **Määrätyn tai Määrittämättömän integraalin malli**, sivu 7.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **integral** (...).

Laskee antiderivaatan, jos *Ala* ja *Ylä* jätetään pois. Integraalin symbolinen vakio jätetään pois, ellet anna argumenttia *Vakio*.

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

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

$$\int (a \cdot x^2, x, c) = \frac{a \cdot x^3}{3} + c$$

Samanarvoisesti kelpaavat antiderivaatat voivat erota numeerisen vakion suhteen. Tällainen vakio on olla naamioitunut—erityisesti, kun antiderivaatta sisältää logaritmeja tai käänteisiä trigonometrisia funktioita. Lisäksi joskus lisätään paloittain määritettyjä vakiolausekkeita, jotta antiderivaatasta saadaan kelvollinen suuremmalle välille kuin tavanomainen kaava.

∫() antaa vastauksena itsensä niille *Laus I:n* paloille, joita se ei pysty määrittämään sisäänrakennettujen funktioidensa ja operaattoreidensa eksplisiittisenä äärellisenä kombinaationa.

Kun annat argumentit *Ala* ja *Ylä*, laskin yrittää paikallistaa mahdolliset epäjatkuvuudet tai epäjatkovat derivaatat välillä *Ala* < *Muutt* < *Ylä* ja jakaa välin näissä kohdissa.

Auto or Approximate (Automaattinen tai likimääräinen) -tilan Auto (Automaattinen) -asetuksessa käytetään numeerista integrointia, mikäli mahdollista, kun antiderivaattaa tai raja-arvoa ei voida määrittää.

Approximate (Likimääräinen) -asetuksessa yritetään ensin numeerista integrointia, mikäli mahdollista. Antiderivaattoja etsitään vain, kun tällainen numeerinen integrointi ei ole mahdollista tai epäonnistuu.

$$\int b \cdot e^{-x^2} + \frac{a}{x^2 + a^2} dx \quad b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$$

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina  .

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

iPad®: Pidä **enter** ja valitse .

$$\int_{-1}^1 e^{-x^2} dx \quad 1.49365$$

∫() (integraali)

Katalogi > 

∫()-komentoa voidaan sijoittaa sisäkkäin usean integraalin suorittamiseksi. Integroinnin raja-arvot voivat riippua niiden ulkopuolella olevista integrointimuuttujista.

Huomaa: Katso myös **nInt()**, sivu 128.

$$\int_0^a \int_0^x \ln(x+y) \, dy \, dx$$

$$\frac{a^2 \cdot \ln(a)}{2} + \frac{a^2 \cdot (4 \cdot \ln(2) - 3)}{4}$$

√() (neliöjuuri)

  **painikkeet**

√(*LausI*) ⇒ *lauseke*

$$\sqrt{4} \quad 2$$

√(*Listal*) ⇒ *lista*

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

Laskee argumentin neliöjuuren.

Kun kyseessä on lista, laskee kaikkien *Listal*:n elementtien neliöjuuret.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **sqrt (...)**

Huomaa: Katso myös **Neliöjuurimalli**, sivu 1.

Π() (tulo)

Katalogi > 

Π(*LausI*, *Muutt*, *Matala*, *Korkea*)
⇒ *lauseke*

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

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **prodSeq (...)**.

Sieventää *LausI*:n jokaisen *Muutt*:n arvon suhteen väliltä *Matala* - *Korkea* ja laskee vastausten tulon.

$$\prod_{k=1}^n (k^2) \quad (n!)^2$$

Huomaa: Katso myös **Kertolaskumalli (Π)**, sivu 6.

$$\prod_{n=1}^5 \left(\left\{ \frac{1}{n}, n, 2 \right\} \right) \quad \left\{ \frac{1}{120}, 120, 32 \right\}$$

Π(*LausI*, *Muutt*, *Matala*, *Matala*-1) ⇒ 1

$$\prod_{k=4}^3 (k) \quad 1$$

Π(*LausI*, *Muutt*, *Matala*, *Korkea*)
⇒ 1/Π(*LausI*, *Muutt*, *Korkea*+1, *Matala*-1), jos *Korkea* < *Matala*-1

Esimerkkien kertolaskukaavat on otettu seuraavasta viitteestä:

Π() (tulo)

Katalogi > 

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) = 6$$

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) \cdot \prod_{k=2}^4 \left(\frac{1}{k} \right) = \frac{1}{4}$$

Σ() (summa)

Katalogi > 

Σ(Laus I, Muutt, Matala, Korkea)
⇒ lauseke

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **sumSeq (...)**.

Sieventää *Laus I*:n jokaisen muuttujan *Muutt* arvon suhteen väliltä *Matala* - *Korkea* ja laskee vastausten summan.

Huomaa: Katso myös **Summamalli**, sivu 5.

$$\sum_{n=1}^5 \left(\frac{1}{n} \right) = \frac{137}{60}$$

$$\sum_{k=1}^n (k^2) = \frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$$

$$\sum_{n=1}^{\infty} \left(\frac{1}{n^2} \right) = \frac{\pi^2}{6}$$

Σ(Laus I, Muutt, Matala, Matala-I) ⇒ 0

$$\sum_{k=4}^3 (k) = 0$$

Σ(Laus I, Muutt, Matala, Korkea)
⇒ Σ(Laus I, Muutt, Korkea+I, Matala-1),
jos Korkea < Matala-I

$$\sum_{k=4}^1 (k) = -5$$

Esimerkkien yhteenlaskukaavat on otettu seuraavasta viitteestä:

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) = 4$$

ΣInt()

Katalogi > 

ΣInt(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [pyör.Arvo]) ⇒ arvo

$$\Sigma \text{Int}(1, 3, 12, 4.75, 20000, , 12, 12) = -213.48$$

ΣInt(*NPmt1*,*NPmt2*,*amortTable*)⇒*arvo*

Lyhennystoiminto, joka laskee koron summan määritetyn maksueräjakson ajalta.

NPmt1 ja *NPmt2* määrittävät maksujakson alku- ja loppurajat.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 205.

- Jos jätät argumentin *Pmt* pois, sen oletusarvoksi tulee ***Pmt=tvmpmt*** (*N*,*I*,*PV*,*FV*,*PpY*,*CpY*,*PmtAt*).
- Jos jätät argumentin *FV* pois, sen oletusarvoksi tulee *FV*=0.
- Argumenttien *PpY*, *CpY* ja *PmtAt* oletusarvot ovat samat kuin TVM-funktioilla.

pyörArvo määrittää pyöristyksessä käytettävien desimaalien määrän.
Oletusarvo=2.

ΣInt(*NPmt1*,*NPmt2*,*amortTable*) laskee koron summan lyhennystaulukon *amortTable* mukaisesti. *amortTable*-argumentin on oltava matriisi, joka on kohdassa **amortTbl()** kuvatun muotoinen, katso sivu 8.

Huomaa: Katso myös ΣPrn(), jäljempänä, sekä **Bal()**, sivu 17.

<i>tbl:=amortTbl(12,12,4.75,20000,,12,12)</i>			
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
ΣInt(1,3, <i>tbl</i>)			-213.48

ΣPrn(*NPmt1*, *NPmt2*, *N*, *I*, *PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*pyörArvo*])
⇒*arvo*

ΣPrn(*NPmt1*,*NPmt2*,*amortTable*)⇒*arvo*

Lyhennystoiminto, joka laskee pääoman summan määritetyn maksujakson ajalta.

NPmt1 ja *NPmt2* määrittävät maksujakson alku- ja loppurajat.

ΣPrn(1,3,12,4.75,20000,,12,12)	-4916.28
--------------------------------	----------

$N, I, PV, Pmt, FV, PpY, CpY$ ja $PmtAt$ on kuvattu TVM-argumenttien taulukossa, sivu 205.

- Jos jätät argumentin Pmt pois, sen oletusarvoksi tulee $Pmt = \mathbf{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.
- Jos jätät argumentin FV pois, sen oletusarvoksi tulee $FV = 0$.
- Argumenttien PpY, CpY ja $PmtAt$ oletusarvot ovat samat kuin TVM-funktioilla.

$pyörArvo$ määrittää pyöristyksessä käytettävien desimaalien määrän.
Oletusarvo=2.

$\Sigma Prn(NPmt1, NPmt2, amortTable)$ laskee pääoman summan lyhennystaulukon $amortTable$ perusteella. $amortTable$ -argumentin on oltava matriisi, joka on kohdassa **amortTbl()** kuvatun muotoinen, katso sivu 8.

Huomaa: Katso myös $\Sigma Int()$, edellä, sekä **Bal()**, sivu 17.

$tbl := \mathbf{amortTbl}(12, 12, 4.75, 20000., 12, 12)$

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

$\Sigma Prn(1, 3, tbl)$ -4916.28

(epäsuora operaattori)

  **painikkeet**

$\# \mathbf{muuttNimiMerkkijono}$

Viittaa muuttujaan, jonka nimi on $\mathbf{muuttNimiMerkkijono}$. Tällä operaattorilla voit luoda muuttujanimiä funktion sisältä merkkijonojen avulla.

$\#("x" \& "y" \& "z")$ xyz

Luo muuttujan xyz tai viittaa siihen.

$10 \rightarrow r$	10
$"r" \rightarrow sI$	"r"
$\#sI$	10

Laskee arvon muuttujalle (r), jonka nimi on tallennettu muuttujaan sI .

E (kymmenpotenssimuoto)

EE painike

<i>mantissa</i> E <i>eksponentti</i>	23000.	23000.
Syöttää luvun kymmenpotenssimuodossa. Luku tulkitaan seuraavasti: <i>mantissa</i> × <i>10</i> ^{<i>eksponentti</i>} .	2300000000.+4.1E15	4.1E15
	3·10 ⁴	30000

Vinkki: Jos haluat syöttää 10-potenssin ilman, että vastauksena on desimaaliluku, käytä komentoa 10^{kokonaisluku}.

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @E. Kirjoita esimerkiksi 2.3@E4, kun haluat syöttää 2.3E4.

g (graadi)

TT painike

<i>Laus I</i> g⇒ <i>lauseke</i>	Aste-, graadi- tai radiaanikulmatilassa.	
<i>Lista I</i> g⇒ <i>lista</i>	cos(50 ^g)	$\frac{\sqrt{2}}{2}$
<i>Matriisi I</i> g⇒ <i>matriisi</i>	cos({0,100 ^g ,200 ^g })	{1,0,-1}

Tämän funktion avulla voit määrittää graadikulman ollessasi aste- tai radiaanikulmatilassa.

Kun laskin on radiaanikulmatilassa, kertoo *Laus I*:n arvolla π/200.

Kun laskin on astekulmatilassa, kertoo *Laus I*:n arvolla g/100.

Graadikulmatilassa antaa vastauksena lausekkeen *Laus I* muuttumattomana.

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla @g.

r (radiaani)

TT painike

<i>Laus I</i> r⇒ <i>lauseke</i>	Aste-, graadi- tai radiaanikulmatilassa:	
<i>Lista I</i> r⇒ <i>lista</i>		
<i>Matriisi I</i> r⇒ <i>matriisi</i>		

ʀ (radiaani)

 painike

Tämän funktion avulla voit määrittää radiaanikulman ollessasi aste- tai graadikulmatilassa.

Kun laskin on astekulmatilassa, kertoo argumentin arvolla $180/\pi$.

Radiaanikulmatilassa antaa vastauksena argumentin muuttumattomana.

Kun laskin on graadikulmatilassa, kertoo argumentin arvolla $200/\pi$.

Vinkki: Käytä komentoa ʀ , jos haluat pakottaa funktion määritelmän yksiköksi radiaanit riippumatta tilasta, joka on käytössä funktion käytön aikana.

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla @r .

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

$$\cos\left(\left\{0^r, \frac{\pi}{12}, \frac{\pi}{r}, -(\pi)^r\right\}\right) \quad \left\{1, \frac{(\sqrt{3+1}) \cdot \sqrt{2}}{4}, -1\right\}$$

° (aste)

 painike

$\text{Laus } l^\circ \Rightarrow \text{lauseke}$

$\text{Listal } l^\circ \Rightarrow \text{lista}$

$\text{Matriisi } l^\circ \Rightarrow \text{matriisi}$

Tämän funktion avulla voit määrittää astekulman ollessasi graadi- tai radiaanikulmatilassa.

Kun laskin on radiaanikulmatilassa, kertoo argumentin arvolla $\pi/180$.

Astekulmatilassa antaa vastauksena argumentin muuttumattomana.

Kun laskin on graadikulmatilassa, kertoo argumentin arvolla $10/9$.

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla @d .

Aste-, graadi- tai radiaanikulmatilassa:

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

Radiaanikulmatilassa:

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina  .

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

iPad®: Pidä **enter** ja valitse .

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) \quad \{1., 0.707107, 0., 0.864976\}$$

$dd^{\circ}mm'ss.ss'' \Rightarrow \text{lauseke}$

Astekulmatilassa:

 dd Positiivinen tai negatiivinen luku $25^{\circ}13'17.5''$ 25.2215 mm Ei-negatiivinen luku $25^{\circ}30'$ $\frac{51}{2}$ $ss.ss$ Ei-negatiivinen lukuLaskee $dd+(mm/60)+(ss.ss/3600)$.

Tässä kantaluku-60:n syötemuodossa voit:

- Syöttää kulman asteina/minuutteina/sekunteina nykyisestä kulmatilasta riippumatta.
- Syöttää kellonajan tunteina/minuutteina/sekunteina.

Huomaa: Merkitse sekuntien $ss.ss$ perään kaksi heittomerkkiä ("), ei lainausmerkkiä (').

∠ (kulma)

[Säde, ∠θ_Kulma] ⇒ vektori
(polaarinen syöte)

Radiaanikulmatilassa ja kun vektori muoto on asetettu valintaan: suorakulma

[Säde, ∠θ_Kulma, Z_Koordinaatti] ⇒ vektori
(lieriömäinen syöte)

$$\left[5 \quad \angle 60^{\circ} \quad \angle 45^{\circ} \right] \left[\frac{5 \cdot \sqrt{2}}{4} \quad \frac{5 \cdot \sqrt{6}}{4} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

[Säde, ∠θ_Kulma, ∠θ_Kulma] ⇒ vektori
(pallonmuotoinen syöte)

sylinteri

Laskee koordinaatit vektorina riippuen vektorimuotoilan asetuksesta: suorakulma, sylinteri tai pallo.

$$\left[5 \quad \angle 60^{\circ} \quad \angle 45^{\circ} \right] \left[\frac{5 \cdot \sqrt{2}}{2} \quad \angle \frac{\pi}{3} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla @<.

pallo

$$\left[5 \quad \angle 60^{\circ} \quad \angle 45^{\circ} \right] \left[5 \quad \angle \frac{\pi}{3} \quad \angle \frac{\pi}{4} \right]$$

(Magnitudi ∠ Kulma) ⇒ kompleksiarvo
(polaarinen syöte)

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

Syöttää kompleksilukuarvon ($r\angle\theta$) polaarisisä muoossa. Kulma tulkitaan nykyisen kulmatila-asetuksen mukaisesti.

$$5+3 \cdot i \left(10 \quad \angle \frac{\pi}{4} \right) \quad 5-5 \cdot \sqrt{2} + (3-5 \cdot \sqrt{2}) \cdot i$$

∠ (kulma)

  painikkeet

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina  .

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

iPad®: Pidä **enter** ja valitse .

$$5+3\cdot i\left(10\angle\frac{\pi}{4}\right) \quad -2.07107-4.07107\cdot i$$

' (jaoton)

 painike

muuttuja'
muuttuja''

Syöttää jaottoman symbolin differentiaaliyhtälöön. Yksi jaottoman symboli tarkoittaa 1. asteen differentiaaliyhtälöä, kaksi jaottoman symbolia tarkoittaa 2. asteen differentiaaliyhtälöä jne.

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

_ (alaviiva edustaa tyhjää elementtiä)

KatsoTyhjät elementit, sivu 249.

_ (alaviiva edustaa yksikön nimeä)

  painikkeet

Laus_Yksikkö

3·_m▶_ft 9.84252·_ft

Määrittää lausekkeen *Laus* yksiköt. Kaikkien yksiköiden nimien alussa on oltava alaviiva.

Huomaa: Muunnoksen symboli ▶ löytyy

Voit käyttää valmiiksi määritettyjä yksiköitä tai luoda omia yksiköitä.

Katalogista. Napsauta ensin  ja sen jälkeen **Matemaattiset operaattorit**.

Valmiiksi määritetyt yksiköt on esitetty vakioiden ja mittausyksiköiden moduulissa. Voit valita yksiköiden nimet Katalogista tai kirjoittaa ne suoraan.

Muuttuja_

Olettaen, että z on määrittämätön:

_ (alaviiva edustaa yksikön nimeä)

  painikkeet

Kun *Muuttujalla* ei ole arvoa, sitä käsitellään ikään kuin se edustaisi kompleksilukua. Oletusarvoisesti, ilman merkkiä `_`, muuttujaa käsitellään reaali lukuna.

$\text{real}(z)$	z
$\text{real}(z_)$	$\text{real}(z_)$
$\text{imag}(z)$	0
$\text{imag}(z_)$	$\text{imag}(z_)$

Jos *Muuttujalla* on arvo, merkkiä `_` ei huomioida, ja *Muuttuja* säilyttää alkuperäisen datatyypinsä.

Huomaa: Voit tallentaa kompleksiluvun muuttujaan käyttämättä alaviivaa `_`. Parhaan tuloksen aikaansaamiseksi alaviivaa `_` on kuitenkin suositeltavaa käyttää laskutoimituksissa, kuten `cSolve()` ja `cZeros()`.

► (muunna)

  painikkeet

Laus_Unit1 ► *_Yksikkö2* ⇒ *Laus_Yksikkö2*

3·_m►_ft 9.84252·_ft

Muuntaa lausekkeen yksiköstä toiseen.

Alaviivan merkki `_` määrittää yksiköt. Yksiköiden tulee kuulua samaan luokkaan, esimerkiksi Pituus tai Pinta-ala.

Saat valmiiksi määritettyjen yksiköiden luettelon näkyviin avaamalla Katalogin ja näyttämällä Yksikkömuunnokset-välilehden:

- Voit valita yksikön nimen luettelosta.
- Voit valita muunnosoperaattorin, ►, luettelon yläosasta.

Voit myös kirjoittaa yksiköiden nimet käsin. Voit syöttää merkin “_” kirjoittaessasi yksiköiden nimiä kämmenlaitteesta painamalla painiketta  .

Huomaa: Kun muunnat lämpötilan yksiköitä, käytä funktioita `tmpCnv()` ja `ΔtmpCnv()`. Muunnosoperaattori ► ei käsittele lämpötilan yksiköitä.

10^()Katalogi > **10^** (*Laus l*) $\Rightarrow$ *lauseke*

$$10^{1.5} \quad 31.6228$$

10^ (*Listal*) $\Rightarrow$ *lista*

$$10^{\{0, -2.2, a\}} \quad \left\{ 1, \frac{1}{100}, 100, 10^a \right\}$$

Laskee luvun 10 korotettuna argumentin potenssiin.

Jos kyseessä on lista, laskee luvun 10 korotettuna *Listal*:n elementtien potenssiin.

10^(*neliömatriisi l*) $\Rightarrow$ *neliömatriisi*

Laskee luvun 10 korotettuna *neliömatriisi l*:n potenssiin. Tämä ei ole sama kuin kunkin elementin 10-potenssiin korottamisen laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} \quad \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

neliömatriisi l:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

^-1(käänteisluku)Katalogi > *Laus l* ^-1 $\Rightarrow$ *lauseke*

$$(3.1)^{-1} \quad 0.322581$$

Listal ^-1 $\Rightarrow$ *lista*

$$\{a, 4, -0.1, x, -2\}^{-1} \quad \left\{ \frac{1}{a}, \frac{1}{4}, -10, \frac{1}{x}, \frac{-1}{2} \right\}$$

Laskee argumentin käänteisluvun.

Jos kyseessä on lista, laskee *Listal*:n elementtien käänteisluvut.

neliömatriisi l ^-1 $\Rightarrow$ *neliömatriisi*

Laskee *neliömatriisi l*:n käänteisluvun.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

neliömatriisi l:n oltava ei-singulaarinen neliömatriisi.

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} \frac{-2}{a-2} & \frac{1}{a-2} \\ \frac{a}{2 \cdot (a-2)} & \frac{-1}{2 \cdot (a-2)} \end{bmatrix}$$

Laus | *Booleanlaus1* [**and***Booleanlaus2*]...

Laus | *Booleanlaus1* [**or***Booleanlaus2*]...

Rajoittava ("|")-symboli toimii binaarisena operaattorina. Operaattorin | vasemmalla puolella oleva operandi on lauseke. Operaattorin | oikealla puolella oleva operandi määrittää yhden tai useampia suhteita, joiden tarkoitus on vaikuttaa lausekkeen sieventämiseen. Operaattorin | jäljessä olevat useat suhteet on yhdistettävä loogisilla operaattoreilla "and" tai "or".

Rajoittava operaattori tarjoaa kolme perustoimintoa:

- Sijoitukset
- Välin rajoitusehdot
- Pois rajaaminen

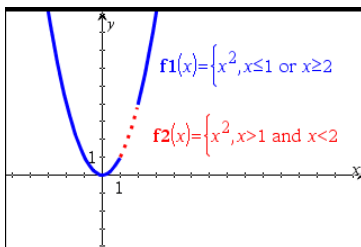
Sijoitukset ovat yhtälön muodossa, kuten $x=3$ tai $y=\sin(x)$. Tehokkainta on, kun vasen puoli on yksinkertainen muuttuja. *Laus* | *Muuttuja* = *arvo* korvaa *arvon* jokaisessa *Muuttujan* esiintymiskohdassa lausekkeessa *Laus*.

Välien rajoitusten muoto on yksi tai useampia epäyhtälöitä, jotka on yhdistetty loogisilla operaattoreilla "and" or "or". Välien rajoitusehdot sallivat myös sievennyksen, joka saattaisi muuten olla kelpaamaton tai ei laskettavissa.

$x+1 x=3$	4
$x+y x=\sin(y)$	$\sin(y)+y$
$x+y \sin(y)=x$	$x+y$

$x^3-2\cdot x+7\rightarrow f(x)$	<i>Done</i>
$f(x) x=\sqrt{3}$	$\sqrt{3}+7$
$(\sin(x))^2+2\cdot\sin(x)-6 \sin(x)=d$	$d^2+2\cdot d-6$

$\text{solve}(x^2-1=0,x) x>0 \text{ and } x<2$	$x=1$
$\sqrt{x}\cdot\sqrt{\frac{1}{x}} x>0$	1
$\sqrt{x}\cdot\sqrt{\frac{1}{x}}$	$\sqrt{\frac{1}{x}}\cdot\sqrt{x}$



| (rajoittava operaattori)

ctrl  **-näppäimet**

Pois sulkemisessa käytetään suhdeoperaattoria "ei ole yhtä kuin" ($\neq$ tai $\neq$), jonka avulla jokin tietty arvo suljetaan pois. Näitä käytetään pääasiallisesti täsmällisen ratkaisun poissulkemiseen käytettäessä funktioita **cSolve()**, **cZeros()**, **fMax()**, **fMin()**, **solve()**, **zeros()**, jne.

$$\text{solve}(x^2-1=0, x) | x \neq 1 \quad x = -1$$

→ (tallenna)

ctrl **var** **painike**

Laus → *Muutt*

Lista → *Muutt*

Matriisi → *Muutt*

Laus → *Funktio*(*Param1*,...)

Lista → *Funktio*(*Param1*,...) *Matriisi* → *Funktio*(*Param1*,...)

$\frac{\pi}{4} \rightarrow \text{myvar}$	$\frac{\pi}{4}$
$2 \cdot \cos(x) \rightarrow y1(x)$	Done
$\{1, 2, 3, 4\} \rightarrow \text{lst5}$	$\{1, 2, 3, 4\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \text{matg}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
"Hello" → <i>str1</i>	"Hello"

Jos muuttujaa *Muutt* ei ole, laskin luo sen ja alustaa sen muotoon *Laus*, *Lista* tai *Matriisi*.

Jos muuttuja *Muutt* on jo olemassa eikä se ole lukittu tai suojattu, laskin korvaa sen sisällön lausekkeella *Laus*, listalla *Lista* tai matriisilla *Matriisi*.

Vinkki: Jos aiot suorittaa symbolisia laskutoimituksia käyttäen määrittämättömiä muuttujia, vältä tallentamasta mitään yleisesti käytettyihin, yksikirjaimisiin muuttujiin, kuten a, b, c, x, y, z ja niin edelleen.

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla =: pikavalintana. Kirjoita esimerkiksi **pi/4 =: myvar**.

:= (määrittä)

  **painikkeet**

Muutt := Laus

Muutt := Lista

Muutt := Matriisi

Funktio(Param1,...) := Laus

Funktio(Param1,...) := Lista *Funktio*
(Param1,...) := Matriisi

Jos muuttujaa *Muutt* ei ole, laskin luo muuttujan *Muutt* ja alustaa sen muotoon *Laus*, *Lista* tai *Matriisi*.

Jos *Muutt* on jo olemassa eikä se ole lukittu tai suojattu, laskin korvaa sen sisällön lausekkeella *Laus*, listalla *List* tai matriisilla *Matriisi*.

Vinkki: Jos aiot suorittaa symbolisia laskutoimituksia käyttäen määrittämättömiä muuttujia, vältä tallentamasta mitään yleisesti käytettyihin, yksikirjaimisiin muuttujiin, kuten a, b, c, x, y, z ja niin edelleen.

$myvar := \frac{\pi}{4}$	$\frac{\pi}{4}$
$yI(x) := 2 \cdot \cos(x)$	<i>Done</i>
$lst5 := \{1, 2, 3, 4\}$	$\{1, 2, 3, 4\}$
$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
$str1 := \text{"Hello"}$	"Hello"

© (kommentti)

  **painikkeet**

© [*teksti*]

© käsittelee *tekstin* kommenttirivinä, minkä avulla voit lisätä merkintöjä luomiisi funktioihin ja ohjelmiin.

© voi olla rivin alussa tai missä tahansa rivin kohdassa. Kaikki merkin © jäljessä oleva, aina rivin loppuun saakka, on kommenttia.

Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Define $g(n) = \text{Func}$	
© <i>Declare variables</i>	
Local $i, result$	
$result := 0$	
For $i, 1, n, 1$ © <i>Loop n times</i>	
$result := result + i^2$	
EndFor	
Return $result$	
EndFunc	
	<i>Done</i>
$g(3)$	14

Ob, Oh

Ob *binaariluku*
Oh *heksadesimaaliluku*

Määrittää binaari- (Ob) tai heksadesimaaliluvun (Oh). Syöttääksesi binaari- tai heksadesimaaliluvun sinun on syötettävä etuliite Ob tai Oh riippumatta kantalukutilasta. Ilman etuliitettä lukua käsitellään desimaalilukuna (kantaluku 10).

Tulokset näytetään kantalukutilan mukaisesti.

Desimaalisessa kantalukutilassa:

Ob10+OhF+10	27
-------------	----

Binaarisessa kantalukutilassa:

Ob10+OhF+10	Ob11011
-------------	---------

Heksadesimaalisessa kantalukutilassa:

Ob10+OhF+10	Oh1B
-------------	------

Tyhjät elementit

Reaalimaailman dataa analysoitaessa käytössäsi ei aina välttämättä ole täydellistä datasarjaa. TI-Nspire™ CAS sallii käyttää tyhjiä dataelementtejä, jolloin voit jatkaa toimintaa lähes täydellisellä datalla tarvitsematta aloittaa alusta tai hylätä epätäydellisiä tapauksia.

Esimerkki tyhjiä elementtejä sisältävästä datasta löytyy luvusta Listat & Taulukot, kohdasta Kuvaajien piirtäminen taulukkodatasta.

Funktion **delVoid()** avulla voit poistaa tyhjät elementit listasta. Funktion **isVoid()** avulla voit testata tyhjää elementtiä. Lisätietoja, katso **delVoid()**, sivu 51, ja **isVoid()**, sivu 99.

Huomaa: Voit syöttää tyhjän elementin manuaalisesti matemaattiseen lausekkeeseen syöttämällä merkin “_” tai avainsanan **void**. Avainsana **void** muuntuu automaattisesti merkiksi “_”, kun lauseke sievennetään. Voit syöttää merkin “_” kämmenlaitteesta painamalla painiketta  .

Tyhjiä elementtejä sisältävät laskutoimitukset

Suurin osa tyhjän syötteen sisältävistä laskutoimituksista tuottaa tyhjän tuloksen. Katso erikoistapaukset jäljempänä.	$\frac{ _ }{_ }$	-
	$\gcd(100, _)$	-
	$3+_$	-
	$\{5,_,10\}-\{3,6,9\}$	$\{2,_,1\}$

Tyhjiä elementtejä sisältävät lista-argumentit

Seuraavat funktiot ja komennot eivät huomioi (ohittavat ne) lista-argumenteista löytyviä tyhjiä elementtejä.	$\text{sum}(\{2,_,3,5,6,6\})$	16.6
	$\text{median}(\{1,2,_,_,3\})$	2
	$\text{cumulativeSum}(\{1,2,_,4,5\})$	$\{1,3,_,7,12\}$
count , countIf , cumulativeSum , freqTable ▶ lista, frekvenssi , max , mean , median , tulo , stDevPop , stDevSamp , sum , sumIf , varPop ja varSamp sekä regressiolaskutoimitukset yhden muuttujan (OneVar), kahden muuttujan (TwoVar) ja FiveNumSummary -tilastot, luottamusvälit ja tilastotestit	$\text{cumulativeSum}\left(\begin{bmatrix} 1 & 2 \\ 3 & - \\ 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 2 \\ 4 & - \\ 9 & 8 \end{bmatrix}$

Tyhjiä elementtejä sisältävät lista-argumentit

SortA ja **SortD** siirtävät kaikki ensimmäisen argumentin sisältämät tyhjät elementit viimeiseksi.

$\{5,4,3,_,1\} \rightarrow list1$	$\{5,4,3,_,1\}$
$\{5,4,3,2,1\} \rightarrow list2$	$\{5,4,3,2,1\}$
SortA list1,list2	Done
list1	$\{1,3,4,5,_\}$
list2	$\{1,3,4,5,2\}$

$\{1,2,3,_,5\} \rightarrow list1$	$\{1,2,3,_,5\}$
$\{1,2,3,4,5\} \rightarrow list2$	$\{1,2,3,4,5\}$
SortD list1,list2	Done
list1	$\{5,3,2,1,_\}$
list2	$\{5,3,2,1,4\}$

Regressioissa X- tai Y-listassa oleva tyhjä elementti aiheuttaa tyhjän elementin jäännöksen vastaavalle elementille.

$l1:=\{1,2,3,4,5\}; l2:=\{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
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	$\{1,_,1,1,1,1\}$

Poisjätetty luokka regressioissa aiheuttaa tyhjän elementin jäännöksen vastaavalle elementille.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
$cat:=\{"M", "M", "F", "F"\}; incl:=\{"F"\}$	$\{"F"\}$
LinRegMx l1,l2,1,cat,incl	Done
stat.Resid	$\{_,_,0,0,0\}$
stat.XReg	$\{_,_,4,5\}$
stat.YReg	$\{_,_,5,6,6\}$
stat.FreqReg	$\{_,_,1,1,1\}$

Frekvenssi 0 regressioissa aiheuttaa tyhjän elementin jäännöksen vastaavalle elementille.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
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	$\{1,_,1,1,1\}$

Matemaattisten lausekkeiden syöttäminen pikavalintojen avulla

Pikavalintojen avulla voit syöttää matemaattisten lausekkeiden elementtejä kirjoittamalla ne näppäimistöltä sen sijaan, että käyttäisit katalogia tai symbolipalettia. Kun haluat esimerkiksi syöttää lausekkeen $\sqrt{6}$, voit kirjoittaa syöteriville `sqrt(6)`. Kun painat painiketta `[enter]`, lauseke `sqrt(6)` muuttuu muotoon $\sqrt{6}$. Jotkin pikavalinnat ovat näppäriä sekä kämmenlaitteesta että tietokoneen näppäimistöltä syötettyinä. Toiset taas ovat käteviä pääasiassa tietokoneen näppäimistöltä syötettyinä.

Kämmenlaitteesta tai tietokoneen näppäimistöltä

Syötettävä kohde:	Pikavalinta:
π	<code>pi</code>
θ	<code>theta</code>
∞	<code>infinity</code>
$\leq$	<code><=</code>
$\geq$	<code>>=</code>
$\neq$	<code>/=</code>
$\Rightarrow$ (looginen seuraus)	<code>=></code>
$\Leftrightarrow$ (looginen kaksoisseuraus, XNOR)	<code><=></code>
$\rightarrow$ (tallennusoperaattori)	<code>=:</code>
$ $ (itseisarvo)	<code>abs(...)</code>
$\sqrt{()}$	<code>sqrt(...)</code>
$d()$	<code>derivative(...)</code>
$\int()$	<code>integral(...)</code>
$\Sigma()$ (yhteenlaskumalli)	<code>sumSeq(...)</code>
$\Pi()$ (tulon malli)	<code>prodSeq(...)</code>
$\sin^{-1}()$, $\cos^{-1}()$, ...	<code>arcsin(...)</code> , <code>arccos(...)</code> , ...
$\Delta\text{List}()$	<code>deltaList(...)</code>
$\Delta\text{tmpCnv}()$	<code>deltaTmpCnv(...)</code>

Tietokoneen näppäimistöltä

Syötettävä kohde:	Pikavalinta:
<code>c1, c2, ...</code> (vakiot)	<code>@c1, @c2, ...</code>
<code>n1, n2, ...</code> (kokonaislukuvakiot)	<code>@n1, @n2, ...</code>
i (imaginaarinen vakio)	<code>@i</code>
e (luonnollisen logaritmin kantaluku e)	<code>@e</code>
E (kymmenpotenssimuoto)	<code>@E</code>
T (transponointi)	<code>@t</code>
r (radiaanit)	<code>@r</code>

Syötettävä kohde:	Pikavalinta:
$^{\circ}$ (asteet)	@d
$^{\circ}$ (graadit)	@g
$\angle$ (kulma)	@<
$\triangleright$ (muunnos)	@>
$\triangleright$Decimal, $\triangleright$approxFraction	@>Decimal, @>approxFraction(), jne.
(), jne.	

EOS-järjestelmän (yhtälökäyttäjärjestelmä) hierarkia

Tässä kappaleessa kuvataan yhtälökäyttäjärjestelmä (EOS™), jota sovelletaan matematiikan ja luonnontieteiden TI-Nspire™ CAS -oppimisteknologiassa. Luvut, muuttujat ja funktiot syötetään yksinkertaisena ja suoraviivaisena sekvenssinä. EOS™-ohjelmisto sieventää lausekkeet ja yhtälöt käyttäen sulkuryhmitäyksiä sekä alla kuvattuja prioriteetteja.

Laskemisjärjestys

Taso Operaattori

- 1 Kaarisulkeet (), hakasulkeet [], aaltosulkeet { }
 - 2 Epäsuora operaattori (#)
 - 3 Funktioiden komennot
 - 4 Jälkioperaattorit: astetta-minuuttia-sekuntia (°, ', "), kertoma (!), prosentti (%), radiaani ($\times^r$), alaindeksi ([]), transponointi ($\top$)
 - 5 Potenssiin korotus, potenssioperaattori ($\wedge$)
 - 6 Negaatio (-)
 - 7 Merkkijonojen ketjutus (&)
 - 8 Kertolasku ($\bullet$), jakolasku (/)
 - 9 Yhteenlasku (+), vähennyslasku (-)
 - 10 Yhtäläisyyssuhteet: on yhtä kuin (=), ei ole yhtä kuin ($\neq$ tai $\neq$), pienempi kuin (<), pienempi tai yhtä suuri kuin ($\leq$ tai $\leq$), suurempi tai yhtä suuri kuin (>), suurempi tai yhtä suuri kuin ($\geq$ tai $\geq$)
 - 11 Looginen **not**
 - 12 Looginen **and**
 - 13 Looginen **or**
 - 14 **xor**, **nor**, **ei-ja**
 - 15 Looginen seuraus ($\Rightarrow$)
 - 16 Looginen kaksoisseuraus, XNOR ($\Leftrightarrow$)
 - 17 Rajoittava operaattori ("|")
 - 18 Tallenna ($\rightarrow$)
-

Kaarisulkeet, hakasulkeet ja aaltosulkeet

Kaikki kaarisulkeiden, hakasulkeiden tai aaltosulkeiden sisällä olevat laskutoimitukset sievennetään ensin. Esimerkiksi lausekkeesta $4(1+2)$ EOS sieventää ensin kaarisulkeiden sisällä olevan osan, $1+2$, ja kertoo tuloksen, 3, sen jälkeen luvulla 4.

Kaikki kaarisulkeiden, hakasulkeiden tai aaltosulkeiden sisällä olevat laskutoimitukset sievennetään ensin. Esimerkiksi lausekkeesta $4(1+2)$ EOS™-ohjelmisto sieventää ensin kaarisulkeiden sisällä olevan osan, $1+2$, ja kertoo sen jälkeen tuloksen, 3, luvulla 4.

Huomaa: Koska käyttäjä voi määrittää omia funktioita TI-Nspire™ CAS -ohjelmistossa, ohjelmisto tulkitsee muuttujan nimen, jonka perässä on sulkeissa oleva lauseke, funktion komennoksi eikä implisiittiseksi kertolaskuksi. Esimerkiksi $a(b+c)$ on funktio a , jota sievennetään lausekkeella $b+c$. Kun haluat kertoa lausekkeen $b+c$ muuttujalla a , käytä eksplisiittistä kertolaskua: $a*(b+c)$.

Epäsuora operaattori

Epäsuora operaattori (#) muuttaa merkkijonon muuttujaksi tai funktion nimeksi. Esimerkiksi lausekkeesta #("x"&"y"&"z") luodaan muuttujanimi xyz. Epäsuora operaattori sallii myös muuttujien luomisen ja muokkaamisen ohjelman sisällä. Esimerkiksi, jos $10 \rightarrow r$ ja $"r" \rightarrow s1$, niin $\#s1=10$.

Jälkioperaattorit

Jälkioperaattorit ovat operaattoreita, jotka tulevat suoraan argumentin perään, kuten 5!, 25% tai $60^\circ 15' 45''$. Argumentit, joiden perässä on jälkioperaattori, sievennetään neljännellä prioriteettitasolla. Esimerkiksi lausekkeesta $4^3!$ sievennetään ensimmäiseksi 3! Vastauksesta, 6, tulee luvun 4 eksponentti, josta saadaan 4096.

Eksponenttilausekkeet

Eksponenttilausekkeet (^) ja potenssiin korotus elementti elementiltä (.^) sievennetään oikealta vasemmalle. Esimerkiksi lauseke 2^3^2 sievennetään samalla tavalla kuin $2^{(3^2)}$, josta saadaan 512. Tämä on eri asia kuin $(2^3)^2$, josta saadaan 64.

Negaatio

Negatiivinen luku syötetään painamalla ensin näppäintä $\boxed{-}$ ja syöttämällä luku sen perään. Jälkioperaattorit ja eksponenttilausekkeet suoritetaan ennen negatiota. Esimerkiksi lausekkeen $-x^2$ tulos on negatiivinen luku, ja $-9^2 = -81$. Käytä sulkeita, kun haluat määrittää neliön negatiivisesta luvusta, kuten $(-9)^2$, josta saadaan 81.

Rajoittavan operaattorin ("|") jälkeen seuraava argumentti tuottaa sarjan ehtoja, jotka vaikuttavat operaattoria edeltävän argumentin laskemiseen.

Vakiot ja arvot

Seuraavassa taulukossa on lueteltu vakiot ja niiden arvot, jotka ovat käytettävissä suoritettaessa yksikköjen muunnoksia. Ne voidaan kirjoittaa manuaalisesti tai valita **Vakiot**-luettelosta **Utilities> Yksikkömuunnokset** (Kannettava laite: Paina  **3**).

Vakio	Nimi	Arvo
_c	valon nopeus	299792458 _m/_s
_Cc	Coulombin vakio	8987551787.3682 _m/_F
_Fc	Faradayn vakio	96485.33289 _coul/_mol
_g	Painovoiman kiihtyvyys	9.80665 _m/_s ²
_Gc	Gravitaatiovakio	6.67408E-11 _m ³ /_kg/_s ²
_h	Planckin vakio	6.626070040E-34 _J_s
_k	Boltzmannin vakio	1.38064852E-23 _J/_°K
_μ0	Tyhjiön permeabiliteetti	1.2566370614359E-6 _N/_A ²
_μb	Bohr magnetoni	9.274009994E-24 _J_m ² /_Wb
_Me	Elektronin lepomassa	9.10938356E-31 _kg
_Mμ	Myonin massa	1.883531594E-28 _kg
_Mn	Neutronin lepomassa	1.674927471E-27 _kg
_Mp	Protonin lepomassa	1.672621898E-27 _kg
_Na	Avogadron vakio	6.022140857E23 /_mol
_q	Elektronin varaus	1.6021766208E-19 _coul
_Rb	Bohrin säde	5.2917721067E-11 _m
_Rc	Moolinen kaasuvakio	8.3144598 _J/_mol/_°K
_Rdb	Rydbergin vakio	10973731.568508/_m
_Re	Elektronin säde	2.8179403227E-15 _m
_u	Atomimassa	1.660539040E-27 _kg
_Vm	moolinen tilavuus	2.2413962E-2 _m ³ /_mol
_ε0	Tyhjiön permittiivisyys (vakio)	8.8541878176204E-12 _F/_m
_σ	Stefan-Boltzmannin vakio	5.670367E-8 _W/_m ² /_°K ⁴
_φ0	Magneettivuon kvantti	2.067833831E-15 _Wb

Virhekoodit ja viestit

Kun tapahtuu virhe, virhekoodi määritetään muuttujaan *errCode*. Käyttäjän määrittämällä ohjelmilla ja funktioilla voidaan tutkia *errCode*-muuttujaa virheen syyn määrittämiseksi. Esimerkki muuttujan *errCode* käytöstä on esitetty Esimerkissä 2 **Try**-komennon kohdalla, sivu 201.

Huomaa: Jotkin virhetilanteet koskevat vain *TI-Nspire™ CAS* -tuotteita ja toiset taas vain *TI-Nspire™*-tuotteita.

Virhekoodi Kuvaus

10	Funktio ei laskenut arvoa.
20	Kokeen tulos ei ollut TOSI eikä EPÄTOSI. Määrittämättömiä muuttujia ei yleensä voi verrata. Esimerkiksi testi $If\ a < b$ aiheuttaa virheen, jos joko a tai b on määrittämätön suoritettaessa If -lauseketta.
30	Argumentti ei voi olla kansion nimi.
40	Argumenttinvirhe
50	Argumentin yhteensopivuusvirhe Kahden tai useamman argumentin on oltava samaa tyyppiä.
60	Argumentin on oltava Boolean lauseke tai kokonaisluku.
70	Argumentin on oltava desimaaliluku.
90	Argumentin on oltava lista.
100	Argumentin on oltava matriisi.
130	Argumentin on oltava merkkijono.
140	Argumentin on oltava muuttujan nimi. Tarkista, että nimi: • ei ala numerolla • ei sisällä välilyöntejä tai erikoismerkkejä • ei käytä alaviivaa tai pistettä väärällä tavalla • ei ylitä pituusrajoituksia Lisätietoja löytyy ohjekirjan Laskin-sovellusta käsittelevästä osasta.
160	Argumentin on oltava lauseke.
165	Paristovirta ei riitä lähettämiseen tai vastaanottamiseen. Asenna uudet paristot ennen lähettämistä tai vastaanottoa.
170	Raja Alarajan on oltava ylärajaa pienempi hakuvälillä määritettäessä.
180	Keskeytys Näppäintä  tai  on painettu pitkän laskutoimituksen tai ohjelman suorituksen aikana.

Virhekoodi Kuvaus

190	Kehämääritys
	Tämä viesti tulee näkyviin muistin loppumisen välttämiseksi muuttujien arvojen äärettömän korvauksen aikana sievennettäessä lausekkeita. Virheen aiheuttaa esimerkiksi lauseke $a+1 \rightarrow a$, jossa a on määrittämätön muuttuja.
200	Määrittelyalueen lauseke ei kelpaa.
	Tämän virheilmoituksen voi aiheuttaa esimerkiksi lauseke $\text{solve}(3x^2-4=0,x) \mid x<0 \text{ tai } x>5$, koska ehto on erotettu operaattorilla "or" eikä operaattorilla "and".
210	Datatyyppi ei kelpaa.
	Argumentin datatyyppi on väärä.
220	Riippuvainen raja-arvo
230	Dimensio
	Lista tai matriisi-indeksi ei kelpaa. Jos esimerkiksi lista $\{1,2,3,4\}$ on tallennettu $L1$:een, tällöin $L1[5]$ aiheuttaa dimensiovirheen, koska $L1$ sisältää vain neljä elementtiä.
235	Dimensiovirhe Listoissa ei ole riittävästi elementtejä.
240	Dimensioiden yhteensopivuusvirhe
	Kahdella tai useammalla argumentilla on oltava sama dimensio. Esimerkiksi lauseke $[1,2]+[1,2,3]$ aiheuttaa dimensioiden yhteensopivuusvirheen, koska matriisit sisältävät eri määrän elementtejä.
250	Jako nolalla
260	Määrittelyjoukkovirhe
	Argumentin on kuuluttava tiettyyn määrittelyjoukkoon. Esimerkiksi rand(0) ei kelpaa.
270	Muuttujan nimi jo käytössä
280	Else ja Elseif eivät kelpaa lohkon If..EndIf ulkopuolella
290	EndTry-komennoista puuttuu sopiva Else-lauseke
295	Liian monta iteraatiota
300	Odotettavissa 2- tai 3-elementtinen lista tai matriisi
310	Funktion nSolve ensimmäisen argumentin on oltava yhden muuttujan yhtälö. Se ei voi sisältää muuta arvotonta muuttujaa kuin käsittelyn kohteena olevan muuttujan.
320	solve - tai cSolve -funktion ensimmäisen argumentin on oltava yhtälö tai epäyhtälö.
	Esimerkiksi $\text{solve}(3x^2-4,x)$ ei kelpaa, koska ensimmäinen argumentti ei ole yhtälö.
345	Ristiriitaiset yksiköt
350	Indeksi alueen ulkopuolella
360	Epäsuora merkijono ei kelpaa muuttujan nimeksi.
380	Määrittämätön Ans
	Edellinen laskutoimitus ei luonut Ans:n arvoa, tai aikaisempaa laskutoimitusta ei ole syötetty.

Virhekoodi Kuvaus

390	Epäkelpo tehtävä
400	Epäkelpo tehtävän arvo
410	Virheellinen komento
430	Virheellinen nykyisille tila-asetuksille
435	Virheellinen arvaus
440	Virheellinen kertomerkitön kertolasku
	Esimerkiksi lauseke $x(x+1)$ ei kelpaa, mutta lausekkeen $x*(x+1)$ syntaksi on oikein. Tarkoituksena on välttää sekaannusta kertomerkitömiä kertolaskujen ja funktioiden komentojen kanssa.
450	Virhe funktiossa tai nykyisessä lausekkeessa
	Vain tietyt komennot kelpaavat käyttäjän määrittämässä funktiossa.
490	Virhe Try..EndTry-lohkossa
510	Virheellinen lista tai matriisi
550	Virheellinen ulkopuolinen funktio tai ohjelma
	Jotkin komennot eivät kelpaa funktion tai ohjelman ulkopuolella. Esimerkiksi Local -komentoa ei voi käyttää, ellei se ole funktion tai ohjelman sisällä.
560	Virhe lohkojen Loop..EndLoop, For..EndFor tai While..EndWhile ulkopuolella
	Esimerkiksi Exit-komento kelpaa vain näiden silmukkalohkojen sisällä.
565	Virheellinen ulkopuolinen ohjelma
570	Polkunimi ei kelpaa
	Esimerkiksi \muutt ei kelpaa.
575	Virheellinen polaarinen kompleksiluku
580	Virheellinen ohjelmaviittaus
	Ohjelmiin ei voi viitata funktioiden tai lausekkeiden sisällä, kuten $1+p(x)$, jossa p on ohjelma.
600	Kelpaamaton taulukko
605	Yksiköitä käytetty väärin
610	Virheellinen muuttujan nimi Local-lausekkeessa
620	Virheellinen muuttujan tai funktion nimi
630	Virheellinen muuttujan viittaus
640	Virheellinen vektorin syntaksi
650	Linkkisiirto
	Kahden laitteen välistä siirtoa ei ole suoritettu loppuun. Tarkista, että kaapeli on kytketty kunnolla kumpaankin laitteeseen.
665	Matriisia ei voi diagonalisoida
670	Muisti vähissä

Virhekoodi Kuvaus

1. Poista tietoja tästä asiakirjasta

2. Tallenna ja sulje tämä asiakirja

Jos toimenpiteet 1 ja 2 eivät auta, poista ja asenna paristot uudelleen

672 Resource exhaustion (Resurssit lopussa)

673 Resource exhaustion (Resurssit lopussa)

680 Puuttuva (

690 Puuttuva)

700 Puuttuva "

710 Puuttuva]

720 Puuttuva }

730 Syntaksista puuttuu lohkon alku tai loppu

740 Then puuttuu lohko If..EndIf

750 Nimi ei ole funktio eikä ohjelma

765 Yhtään funktiota ei ole valittu

780 Ratkaisua ei löydy

800 Ei-reaalinen tulos

Esimerkiksi, jos ohjelmisto on Real-asetuksessa, $\sqrt{-1}$ ei kelpaa.

Jotta voit saada kompleksilukuvastauksia, muuta reaali- tai kompleksitilan asetus valintaan RECTANGULAR (SUORAKULMA) tai POLAR (POLAARINEN).

830 Laskualueen ylitys

850 Ohjelmaa ei löydy

Toisen ohjelman sisällä olevaa ohjelman viittausta ei löytynyt määritetystä polusta ohjelman suorittamisen aikana.

855 Rand-tyyppiset funktiot eivät ole sallittuja kuvaajan piirtämisen aikana

860 Liian syvä rekursio

870 Varattu nimi tai järjestelmän muuttuja

900 Argumenttinvirhe

Mediaani-mediaani-mallia ei voitu käyttää datasarjaan.

910 Syntaksivirhe

920 Tekstiä ei löydy

930 Liian vähän argumentteja

Funktiosta tai komennosta puuttuu yksi tai useampia argumentteja.

940 Liian monta argumenttia

Virhekoodi Kuvaus

Lauseke tai yhtälö sisältää liian monta argumenttia, eikä sitä voi ratkaista.

950 Liian monta alaindeksiä

955 Liian monta määrittämätöntä muuttujaa

960 Muuttujaa ei ole määritetty

Muuttujalle ei ole määritetty arvoa. Käytä jotakin seuraavista komennoista:

- `sto →`
- `:=`
- **Define**

määrittääksesi muuttujille arvot.

965 Käyttöjärjestelmälle ei ole lisenssiä

970 Muuttuja on käytössä, joten viittaukset tai muutokset eivät ole sallittuja

980 Muuttuja on suojattu

990 Virheellinen muuttujan nimi

Varmista, että nimi ei ylitä pituusrajoituksia

1000 Ikkunamuuttujien määrittelyalue

1010 Zoomaus

1020 Sisäinen virhe

1030 Suojatun muistin rikkomus

1040 Toimintoa ei tueta Tämä toiminto vaatii tietokonealgebrajärjestelmän Kokeile TI-Nspire™ CAS -järjestelmää.

1045 Operaattoria ei tueta. Tämä operaattori vaatii tietokonealgebrajärjestelmän. Kokeile TI-Nspire™ CAS -järjestelmää.

1050 Toimintoa ei tueta. Tämä operaattori vaatii tietokonealgebrajärjestelmän. Kokeile TI-Nspire™ CAS -järjestelmää.

1060 Syötetyn argumentin on oltava numeerinen. Vain numeerisia arvoja sisältävät syötteet sallitaan.

1070 Trig-funktion argumentti on liian suuri täsmälliseen sievennykseen

1080 Ans-muuttujan käyttöä ei tueta. Tämä sovellus ei tue Ans-muuttujan käyttöä.

1090 Funktiota ei ole määritetty. Käytä jotakin seuraavista komennoista:

- **Define**
- `:=`
- `sto →`

määrittääksesi funktion.

1100 Ei-reaalinen laskutoimitus

Esimerkiksi, jos ohjelmisto on Real-asetuksessa, $\sqrt{-1}$ ei kelpaa.

Virhekoodi Kuvaus

	Jotta voit saada kompleksilukuvastauksia, muuta reaali- tai kompleksitilan asetus valintaan RECTANGULAR (SUORAKULMA) tai POLAR (POLAARINEN).
1110	Virheelliset rajat
1120	Ei etumerkin muutosta
1130	Argumentti ei voi olla lista tai matriisi
1140	Argumenttinvirhe Ensimmäisen argumentin on oltava toisen argumentin sisällä oleva polynomi-lauseke. Jos toinen argumentti jätetään pois, ohjelmisto yrittää valita oletusarvon.
1150	Argumenttinvirhe Ensimmäisten kahden argumentin on oltava kolmannen argumentin sisällä olevia polynomi-lausekeita. Jos kolmas argumentti jätetään pois, ohjelmisto yrittää valita oletusarvon.
1160	Virheellinen kirjaston polkunimi Polkunimen on oltava muodossa xxx\yyy, jossa: • xxx-osassa voi olla 1-16 merkkiä. • yyy-osassa voi olla 1–15 merkkiä. Lisätietoja löydät ohjekirjan osasta Kirjasto.
1170	Kirjaston polkunimeä on käytetty väärin • Polkunimelle ei voi määrittää arvoa komennon Define, := tai sto → avulla. • Polkunimeä ei voi määrittää paikalliseksi muuttujaksi tai käyttää funktion tai ohjelman parametrina.
1180	Virheellinen kirjastomuuttujan nimi. Tarkista, että nimi: • ei sisällä pistettä • ei ala alaviivalla • ei ylitä 15 merkkiä Lisätietoja löydät ohjekirjan osasta Kirjasto.
1190	Kirjastodokumenttia ei löydy: • Tarkista, että kirjasto on MyLib-kansiossa. • Päivitä kirjastot. Lisätietoja löydät ohjekirjan osasta Kirjasto.
1200	Kirjastomuuttujaa ei löydy: • Tarkista, että kirjaston ensimmäisessä tehtävässä on kirjastomuuttuja. • Tarkista, että kirjastomuuttujaksi on määritetty LibPub tai LibPriv. • Päivitä kirjastot. Lisätietoja löydät ohjekirjan osasta Kirjasto.

Virhekoodi Kuvaus

1210	Virheellinen kirjaston pikavalintanimi. Tarkista, että nimi: • ei sisällä pistettä • ei ala alaviivalla • ei ylitä 16 merkkiä • ei ole varattu nimi Lisätietoja löydät ohjekirjan osasta Kirjasto.
1220	Määrittelyjoukkovirhe: Funktiot <code>tangentLine</code> ja <code>normalLine</code> tukevat vain reaaliarvoisia funktioita.
1230	Määrittelyjoukkovirhe. Trigonometrysten muunnosten operaattoreita ei tueta aste- eikä graadikulmatilassa.
1250	Argumenttiovirhe Käytä lineeariyhtälöryhmää. Esimerkki kahden lineeariyhtälön ryhmästä, jossa on muuttujat x ja y : $3x+7y=5$ $2y-5x=-1$
1260	Argumenttiovirhe: nfMin:n tai nfMax:n ensimmäisen argumentin on oltava yhden muuttujan sisältämä lauseke. Se ei voi sisältää muuta arvotonta muuttujaa kuin käsittelyn kohteena olevan muuttujan.
1270	Argumenttiovirhe Derivaatan asteen on oltava 1 tai 2.
1280	Argumenttiovirhe Käytä yhden muuttujan sisältämää laajennetussa muodossa olevaa polynomia.
1290	Argumenttiovirhe Käytä yhden muuttujan sisältämää polynomia.
1300	Argumenttiovirhe Polynomin kertoimien tuloksena on oltava numeerinen arvo.
1310	Argumenttiovirhe: Yhtä tai useampaa funktion argumenttia ei voi sieventää.
1380	Argumenttiovirhe: Sisäkkäisiä kutsuja <code>domain()</code> funktioon ei sallita.

Varoituskoodit ja -viestit

Voit käyttää funktiota `warnCodes()` tallentaaksesi lausekkeen laskennan luomat varoituskoodit. Tämä taulukko luettelee jokaisen numeerisen varoituskoodin ja siihen liittyvän viestin.

Esimerkki varoituskoodien tallentamisesta, katso `warnCodes()`, sivu 210.

Varoituskoodi	Viesti
10000	Laskutoimitus voi antaa väärää ratkaisuja.
10001	Yhtälön derivointi voi antaa väärän yhtälön.
10002	Kyseenalainen ratkaisu
10003	Kyseenalainen tarkkuus
10004	Laskutoimituksessa voi hävitä ratkaisuja.
10005	<code>cSolve</code> voi määrittää lisää nollakohtia.
10006	<code>Solve</code> voi määrittää lisää nollia.
10007	Lisää ratkaisuja voi olla olemassa. Yritä määrittämällä oikeat ala- ja ylärajat ja/tai arvaamalla. Esimerkkejä <code>solve()</code> käytöstä: <code>solve(Yhtälö, Muut=Arvaus) alaraja<Muut<yläraja</code> <code>solve(Yhtälö, Muut) alaraja<Muut<yläraja</code> <code>solve(Yhtälö, Muut=Arvaus)</code>
10008	Tuloksen määrittelyjoukko voi olla pienempi kuin syötteen määrittelyjoukko.
10009	Tuloksen määrittelyjoukko voi olla suurempi kuin syötteen määrittelyjoukko.
10012	Ei-reaalinen laskutoimitus
10013	∞^0 tai undef^0 korvattu arvolla 1
10014	undef^0 korvattu arvolla 1
10015	1^∞ tai 1^undef korvattu arvolla 1
10016	1^undef korvattu arvolla 1
10017	Laskualueen ylitys korvattu arvolla ∞ tai $-\infty$
10018	Laskutoimitus vaatii ja laskee 64-bittisen arvon.
10019	Muisti täynnä, sievennystä ei mahdollisesti suoriteta loppuun.
10020	Trig-funktion argumentti on liian suuri täsmälliseen sievennykseen.
10021	Syöte sisältää määrittämättömän parametrin.

Varoituskoodi	Viesti
	Tulos ei ehkä ole voimassa kaikille mahdollisille parametriarvoille.
10022	Ratkaisu on mahdollinen, kun määritetään oikeat ala- ja ylärajat.
10023	Skalaari on kerrottu identtisellä matriisilla.
10024	Tulos saatu käyttämällä likimääräistä aritmetiikkaa.
10025	Ekvivalenssia ei voida varmistaa EXACT-tilassa.
10026	Ehto saatetaan ohittaa. Määrittele ehto muodossa "\" 'Muuttuja MatemaattinenTestiSymboli Vakio' tai näiden muotojen yhdistelmänä, esimerkiksi 'x<3 ja x>-12'

Yleistä

Online-tuki

education.ti.com/eguide

Valitse maasi, niin näet lisää tuotetietoja.

Ota yhteyttä TI-tukeen

education.ti.com/ti-cares

Valitse maasi, niin näet teknisiä tietoja ja muita tukiresursseja.

Huolto- ja takuutiedot

education.ti.com/warranty

Valitse maasi saadaksesi lisätietoja takuun kestosta ja ehdoista tai tuotetuesta.

Rajoitettu takuu. Tämä takuu ei vaikuta lainmukaisiin oikeuksiisi

Indeksi

		$\wedge$, potenssi	224
	–		
–, vähennyslasku[*]	221	– yksikön merkki	242
	!	– yksikön nimi	242
!, kertoma	232		
	"	, (rajoittava operaattori)	245
", sekuntimuoto	241	+	
	#	+, yhteenlasku	221
#, epäsuora operaattori	238, 254	/	
	%	/, jakolasku[*]	223
%, prosentti	227	≠	
	&	≠, ei yhtä kuin[*]	228
&, liitä	232	=	
	*	=, on yhtä kuin	227
*, kertolasku	222	>	
	,	>, suurempi kuin	230
, jaoton	242	Π	
, minuuttimuoto	241	Π, tulo[*]	235
	.	Σ	
., piste-erotus	225	Σ(), summa[*]	236
., pistetulo	226	ΣInt()	236
./, pisteosamäärä	226	ΣPrn()	237
., pistepotenssi	226	√	
., pistesumma	225	√, neliöjuuri[*]	235
	:	∫	
:=, määritä	247	∫, integraali[*]	233
	^		
$\wedge^{-1}$, käänteisluku	244		

$\leq$		$\Leftrightarrow$	
$\leq$, pienempi tai yhtä suuri kuin	229	$\Leftrightarrow$, looginen kaksoisseuraus[*]	231
$\geq$		$\textcircled{C}$	
$\geq$, suurempi tai yhtä suuri kuin	230	$\textcircled{C}$, kommentti	247
$\blacktriangleright$		$\circ$	
$\blacktriangleright$, muunna graadikulmaksi[Grad]	91	$^\circ$, astekulmamuoto[*]	240
$\blacktriangleright$, muunna yksiköt[*]	243	$^\circ$, astetta/minuuttia/sekuntia[*]	241
$\blacktriangleright$, asteen regressio, QuartReg	149	0	
$\blacktriangleright$ approxFraction()	14	0b, binaarinen indikaattori	248
$\blacktriangleright$ cos, näytä kulman kosini[cos]	31	0h, heksadesimaalinen indikaattori	248
$\blacktriangleright$ Cylind, näytä vektorina sylinderikoordinaatistossa [Cylind]	44	1	
$\blacktriangleright$ DD, näytä desimaalikulmana[DD]	48	$10^{\wedge}()$, kymmenen potenssi	244
$\blacktriangleright$ Desimaali, näytä tulos desimaalilukuna[Desimaali]	48	2	
$\blacktriangleright$ DMS, näytä asteina/minuutteina/sekun teina[DMS]	57	2 otoksen F-testi	79
$\blacktriangleright$ exp, näytä e:n arvolla[exp]	66	2. asteen regressio, QuadReg	147
$\blacktriangleright$ Kantaluku10, näytä desimaalikonaislukuna [Kantaluku10]	19	3	
$\blacktriangleright$ Kantaluku16, näytä heksadesimaalilukuna [Kantaluku16]	20	3. asteen regressio, CubicReg	42
$\blacktriangleright$ Kantaluku2, näytä binaarisena [Kantaluku2]	18	A	
$\blacktriangleright$ Polaarinen, näytä polaarisenä vektorina[Polaarinen]	139	abs(), itseisarvo	8
$\blacktriangleright$ Rad, muunna radiaanikulmaan	150	alaraja, floor()	74
$\blacktriangleright$ Rect, näytä suorakulmavektorina	154	alaviiva, _	242
$\blacktriangleright$ sin, näytä kulman sini[sin]	176	alimatriisi, subMat()	189, 191
$\blacktriangleright$ Sphere, näytä pallonmuotoisena vektorina[Sphere]	184	amortTbl(), lyhennystaulukko	8, 17
$\Rightarrow$		and, Boolean operator	9
$\Rightarrow$, looginen seuraus[*]	231, 251	angle(), kulma	10
$\rightarrow$		ANOVA, yksisuuntainen varianssianalyysi	10
$\rightarrow$, tallenna	246	ANOVA2way, kaksisuuntainen varianssianalyysi	11
		ans, viimeinen vastaus	13
		approx(), likimääräinen	13
		approxRational()	14
		arccos()	14
		arccosh()	14

arccot()	14	centralDiff()	22
arccoth()	15	cFactor(), kompleksilukutekijä	22
arccsc()	15	char(), merkkijono	23
arccsch()	15	charPoly()	24
arcLen(), kaaren pituus	15	χ^2 2way	24
arcsec()	15	clearAZ	26
arcsech()	15	ClrErr, poista virhe	26
arcsin()	15	colAugment	27
arsinh()	16	colDim(), matriisin sarakemitat	27
arctan()	16	colNorm(), matriisin sarakenormi	27
arctanh()	16	comDenom(), yhteinen nimittäjä	28
arkuskosini, $\cos^{-1}()$	33	completeSquare(), complete square	29
arkussini, $\sin^{-1}()$	177	conj(), liittokompleksiluku	29
arkustangentti, $\tan^{-1}()$	192	constructMat(), luo matriisi	30
aseta		corrMat(), korrelaatiomatriisi	31
tila, setMode()	171	$\cos^{-1}$, arkuskosini	33
asetukset, hae nykyiset	88	cos(), kosini	32
aste/minuutti/sekunti-muoto	241	cosh $^{-1}()$, hyperbolinen arkuskosini	34
astekulmamuoto, -	240	cosh(), hyperbolinen kosini	34
augment(), lisää/ketjuta	16	cot $^{-1}()$, arkuskotangentti	35
avgRC(), keskimääräinen		cot(), kotangentti	35
muutosnopeus	16	coth $^{-1}()$, hyperbolinen	
		arkuskotangentti	36
B		coth(), hyperbolinen kotangentti	36
binaarinen		count(), laske listan kohtien	
indikaattori, Ob	248	lukumäärä	36
näytä, 4Kantaluku2	18	countif(), laske listan yksiköiden	
binomCdf()	20, 97	lukumäärä ehdollisesti	37
binomPdf()	21	cPolyRoots()	38
Boolean operators		crossP(), ristitulo	38
and	9	csc $^{-1}()$, käänteiskosekantti	39
Boolean operaattorit		csc(), kosekantti	38
$\Rightarrow$	231, 251	csch $^{-1}()$, käänteinen hyperbolinen	
$\Leftrightarrow$	231	kosekantti	39
ei	131	csch(), hyperbolinen kosekantti	39
ei-ja	125	cSolve(), ratkaise kompleksiyhtälö	40
eikä	129	CubicReg, 3. asteen regressio	42
tai	136	cumulativeSum(), kumulatiivinen	
xor	211	summa	44
		cycle, Cycle	44
C		Cycle, cycle	44
Cdf()	73	cZeros(), kompleksiset nollakohdat	45
ceiling(), yläraja	21		

D

d(), ensimmäinen derivaatta	232
dbd(), päivämäärien väliset päivät ..	47
Define (Määritä)	48
Define LibPriv (Määritä LibPriv)	49
Define LibPub (Määritä LibPub)	50
Define, määritä	48
deltaList()	51
deltaTmpCnv()	51
DelVar, poista muuttuja	51
delVoid(), poista tyhjät elementit ..	51
derivaatat	
ensimmäinen derivaatta, d() ..	232
numeerinen derivaatta, nDeriv()	128
numeerinen derivaatta, nDerivative()	127
derivaatta tai n:s derivaatta	
malli	6
desimaali	
kokonaislukunäyttö, 4Kantaluku10	19
kulmanäyttö, ►DD	48
deSolve(), ratkaisu	52
det(), matriisin determinantti	54
diag(), diagonaalimatriisi	54
dim(), mitta	54
Disp, näytä tiedot	55, 167
DispAt	55
dominantTerm(), dominanttitermi ..	58
dominanttitermi, dominantTerm() ..	58
dotP(), pistetulo	60

E

e eksponentti	
malli	2
e korotettuna potenssiin, e^()	60, 67
E, eksponentti	239
e, näytä lauseke e:n arvolla	66
e^(), e korotettuna potenssiin	60
efektiivinen korko, eff()	61
eff(), muunna nimelliskorko efektiiviseksi korkoksi	61
ei-ja, Boolean operaattori	125
ei yhtä kuin, ≠	228

ei, Boolean operaattori	131
eigVc(), ominaisvektori	61
eigVL(), ominaisarvo	62
eikä, Boolean operaattori	129
eksklusio "!"-operaattorilla	245
eksponentiaalinen regressio, ExpReg	69
eksponentit	
malli	1
eksponentti, E	239
else if, Elseif	62
Elseif, else if	62
end	
for, EndFor	76
funktio, EndFunc	80
ohjelma, EndPrgm	144
silmukka, EndLoop	115
try, EndTry	201
while, EndWhile	211
end while, EndWhile	211
EndTry, kokeilun loppu	201
EndWhile, end while	211
ensimmäinen derivaatta	
malli	6
EOS (yhtälökäyttöjärjestelmä)	253
epäsuora operaattori (#)	254
epäsuora operaattori, #	238
etumerkki, sign()	174
euler(), Euler function	63
exact(), täsmällinen	66
Exit, lopeta	66
exp(), e korotettuna potenssiin	67
exp►list(), lauseke listaksi	67
expand(), lavenna	68
expr(), merkkijono lausekkeeksi	69, 113
ExpReg, eksponentiaalinen regressio	69

F

factor(), tekijä	71
Fill, matriisin täyttö	73
FiveNumSummary	73
floor(), alaraja	74
fMax(), funktion maksimi	75
fMin(), funktion minimi	75

For	76		
for, For	76		
For, for	76		
format(), merkkijonon muoto	77		
fpart(), funktion osa	77		
freqTable()	78		
frequency()	79		
Frobeniusin normi, norm()	130		
Func, funktio	80		
Func, ohjelmafunktio	80		
funktion loppu, EndFunc	80		
funktiot			
käyttäjän määrittämät	48		
maksimi, fMax()	75		
minimi, fMin()	75		
ohjelmafunktio, Func	80		
osa, fpart()	77		
funktiot ja muuttujat			
kopioiminen	30		
G			
g, graadit	239		
gcd(), suurin yhteinen jakaja	81		
geomCdf()	81		
geomPdf()	82		
Get	82		
getDenom(), hae/laske nimittäjä ..	83		
getKey()	83		
getLangInfo(), hae/anna kielitiedot ..	87		
getLockInfo(), testaa muuttujan tai			
muuttujaryhmän			
lukitustilan	87		
getMode(), hae tila-asetukset	88		
getNum(), hae/laske luku	89		
GetStr	89		
getType(), get type of variable	89		
getVarInfo(), hae/laske muuttujien			
tiedot	90		
Goto, siirry	91		
graadimuoto, g	239		
		H	
		hae/laske	
		luku, getNum()	89
		muuttujien tiedot, getVarInfo() ..	87, 90
		nimittäjä, getDenom()	83
		heksadesimaali	
		näytä, 4Kantaluku16	20
		heksadesimaalinen	
		indikaattori, Oh	248
		hyperbolinen	
		areatangenti, tanh ⁻¹ ()	194
		arkuskosini, cosh ⁻¹ ()	34
		arkussini, sinh ⁻¹ ()	178
		kosini, cosh()	34
		sini, sinh()	178
		tangenti, tanh()	193
		I	
		identity(), identtinen matriisi	91
		identtinen matriisi, identity()	91
		If, jos	92
		ifFn()	93
		imag(), imaginaarinen osa	94
		imaginaarinen osa, imag()	94
		ImpDif(), implisiittinen derivaatta ..	94
		implisiittinen derivaatta, Impdif() ..	94
		Input, syöte	94
		inString(), merkkijonon sisällä	94
		int(), kokonaisluku	95
		intDiv(), kokonaisluvun jakaminen ..	95
		integraali, %o	233
		interpolate(), interpoloi	96
		invF()	96
		invNorm(), käänteinen	
		kumulatiivinen	
		normaalijakauma)	98
		invt()	98
		Invχ ² ()	96
		iPart(), kokonaisosa	98
		irr(), sisäinen palautuksen määrä	
		sisäinen palautuksen määrä, irr(	
		)	98
		isPrime(), jaottoman testi	99

isVoid(), tyhjän testi	99	kierrä, rotate()	161-162
itseisarvo		kirjasto	
malli	4	luo pikavalinnat objekteihin ...	101
J		kokonaisluku, int()	95
jakakaumafunktiot		kokonaisluvun jakaminen, intDiv() ..	95
binomCdf()	97	kokonaisosa, iPart()	98
invNorm()	98	kombinaatiot, nCr()	126
invt()	98	kommentti, ©	247
jakaumafunktiot		kompleksi	
binomCdf()	20	liitto, conj()	29
binomPdf()	21	nollakohdat, cZeros()	45
Inv χ^2 ()	96	ratkaise, cSolve()	40
normCdf()	131	tekijä, cFactor()	22
normPdf()	131	kopioi muuttuja tai funktio, CopyVar	30
poissCdf()	139	korkoerien summa	236
poissPdf()	139	korrelaatiomatriisi, corrMat()	31
tCdf()	195	korvaaminen "I"-operaattorilla	245
tPdf()	200	kosini	
χ^2 2way()	24	näytä kulman	31
χ^2 Cdf()	25	kosini, cos()	32
χ^2 GOF()	25	kotangentti, cot()	35
χ^2 Pdf()	26	kulma, angle()	10
jakolasku, P	223	kumulatiivinen summa,	
jaoton,	242	cumulativeSum()	44
jaottoman luvun testi, isPrime()	99	kun, when()	210
jos käytettävissäsi on 	245	kymmenen potenssi, 10^()	244
jos, EndIf	92	käyttäjän määrittämät funktiot	48
jos, If	92	käyttäjän määrittämät funktiot ja	
jäännös, remain()	157	ohjelmat	49-50
K		käänteinen kumulatiivinen	
kaaren pituus, arcLen()	15	normaalijakauma (invNorm(	
kahden muuttujan tulokset, TwoVar	205	)	98
kertolasku, *	222	käänteinen, $\wedge^{-1}$	244
kertoma, !	232	käänteisluku, $\wedge^{-1}$	244
keskiarvo, mean()	117	L	
keskihajonta, stdDev()	187-188, 208	lajitteleminen	
keskimäinen merkkijono, mid() ..	120	laskeva järjestys, SortD	184
keskimääräinen muutosnopeus,		nouseva järjestys, SortA	184
avgRC()	16	laske listan kohtien lukumäärä,	
kieli		count()	36
hae kielitiedot	87	laske listan yksiköiden lukumäärä	
		ehdollisesti, countIf()	37
		laske polynomi, polyEval()	141
		laske päivämäärien väliset päivät,	47

dbd()			minimi, min()		120
lausekkeet			pistetulo, dotP()		60
lauseke listaksi, exp►list()		67	ristitulo, crossP()		38
merkkijono lausekkeeksi, expr()	69, 113		tulo, product()		145
lavenna, expand()		68	tyhjät elementit		249
Lbl, tunnus		100	uusi, newList()		127
lcm, pienin yhteinen jaettava		100	yhteenlasku, sum()		190
left(), vasen		100	lisää/ketjuta, augment()		16
LibPriv		49	ln(), luonnollinen logaritmi		109
LibPub		50	LnReg, logaritminen regressio		110
libShortcut(), luo pikavalinnat			Local, paikallinen muuttuja		111
kirjasto-objekteihin		101	Lock, lukitse muuttuja tai		
liitä, &		232	muuttujaryhmä		111
likimääräinen, approx()		13	Logaritmi		
limit() tai lim(), raja-arvo		102	malli		2
lineaarinen regressio, LinRegAx		104	logaritminen regressio, LnReg		110
lineaarinen regressio, LinRegBx		103, 105	logaritmit		109
LinRegBx, lineaarinen regressio		103	Logistic, logistinen regressio		113
LinRegMx, lineaarinen regressio		104	LogisticD, logistinen regressio		114
LinRegtIntervals, lineaarinen			logistinen regressio, Logistic		113
regressio		105	logistinen regressio, LogisticD		114
LinRegtTest		106	looginen kaksoisseuraus, ⇔		231
linSolve()		107	looginen seuraus, ⇒		231, 251
Δlist(), listaerotus		108	Loop, silmukka		115
list►mat(), listasta matriisiksi		108	lopetta, Exit		66
lista, laske kohtien lukumäärä		36	loppu		92
lista, laske yksiköiden lukumäärä			loppu jos, EndIf		92
ehdollisesti		37	LU, matriisin ala-ylä-dekomponointi		116
listasta matriisiksi, list►mat()		108	lukujono		
listat			kierrä, rotate()		161-162
erotukset listassa, @list()		108	oikea, right()		159-160
erotus, @list()		108	lukujonot		
keskimmäinen merkkijono, mid()		120	oikea, right()		96
kumulatiivinen summa,			luo matriisi, constructMat()		30
cumulativeSum()		44	luonnollinen logaritmi, ln()		109
lajittele laskevaan järjestykseen,			lyhennystaulukko, amortTbl()		8, 17
SortD		184			
lajittele nousevaan					
järjestykseen, SortA		184			
lauseke listaksi, exp►list()		67			
listasta matriisiksi, list►mat()		108			
lisää/ketjuta, augment()		16			
maksimi, max()		117			
matriisi listaksi, mat►list()		116			

M

maksimi, max()		117
mallit		
derivaatta tai n:s derivaatta		6
eksponentti		2
eksponentti		1

ensimmäinen derivaatta	6	mitta, dim()	54
itseisarvo	4	ominaisarvo, eigVl()	62
Logaritmi	2	ominaisvektori, eigVc()	61
matriisi (1×2)	4	piste-erotus, -	225
matriisi (2×1)	5	pisteosamäärä, ./	226
matriisi (2×2)	4	pistepotenssi, .^	226
matriisi ($m \times n$)	5	pistesumma, .+	225
murtoluku	1	pistetulo, .*	226
määrittämätön integraali	7	QR-hajottaminen, QR	147
määrätty integraali	7	rivi-echelon-muoto, ref()	154
n:s juuri	2	rivien kertominen ja yhteenlasku, mRowAdd ()	122
nelöjuuri	1	rivien yhteenlasku, rowAdd() ..	164
paloittain määritelty funktio (2- osainen)	2	rivikoko, rowDim()	164
paloittain määritelty funktio (N- osainen)	3	rivinormi, rowNorm()	164
raja-arvo	7	rivinvaihto, rowSwap()	164
sum (G)	5	rivioperaatio, mRow()	122
toinen derivaatta	6	sarakemitat, colDim()	27
tulo (P)	6	sarakenormi, colNorm()	27
yhtälöpari (2 yhtälöä)	3	satunnainen, randMat()	152
yhtälöryhmä (N yhtälöä)	4	transponoi, T	191
mat►list(), matriisi listaksi	116	tulo, product()	145
matriisi (1×2) malli	4	täyttäminen, Fill	73
matriisi (2×1) malli	5	uusi, newMat()	127
matriisi (2×2) malli	4	yhteenlasku, sum()	190
matriisi ($m \times n$) malli	5	max(), maksimi	117
matriisi listaksi, mat►list()	116	mean(), keskiarvo	117
matriisit ala-ylä-dekomponointi, LU	116	mediaani-mediaani-suoran regressio, MedMed	118
alimatriisi, subMat()	189, 191	mediaani, median()	118
determinantti, det()	54	median(), mediaani	118
diagonaali, diag()	54	MedMed, mediaani-mediaani-suoran regressio	118
identtinen, identity()	91	merkit merkkijono, char()	23
kumulatiivinen summa, cumulativeSum()	44	numeerinen koodi, ord()	137
listasta matriisiksi, list►mat() ...	108	merkkijono mitta, dim()	54
lisää/ketjuta, augment()	16	pituus	54
maksimi, max()	117	merkkijono, char()	23
matriisi listaksi, mat►list()	116	merkkijonon muoto, format()	77
minimi, min()	120	merkkijonon pituus	54
		merkkijonon sisällä, inString()	94

merkkijonot		muuttuja	
epäsuora operaattori, #	238	nimen luominen merkkijonosta	254
keskimmäinen merkkijono, mid(		muuttujat	
)	120	paikallinen, Local	111
käyttö muuttujanimien		poista kaikki yksikirjaimiset	26
luomisessa	254	poista, DelVar	51
lauseke merkkijonoksi, string()	189	muuttujat ja funktiot	
liitä, &	232	kopioiminen	30
merkkijono lausekkeeksi, expr()	69, 113	muuttujat, lukitseminen ja	
merkkijono, char()	23	vapauttaminen	87, 111, 208
merkkikoodi, ord()	137	muuttujien ja muuttujaryhmien	
muoto, format()	77	lukitseminen	111
muotoilu	77	muuttujien ja muuttujaryhmien	
siirrä, shift()	173	vapauttaminen	208
sisällä, inString()	94	määrittäminen	
vasen, left()	100	julkinen funktio tai ohjelma	50
mid(), keskimmäinen merkkijono ..	120	yksityinen funktio tai ohjelma ..	49
min(), minimi	120	määrittämätön integraali	
minimi, min()	120	malli	7
minuuttimuoto,	241	määritä, Define	48
mirr(), modifioitu sisäinen		määrätty integraali	
korkokanta	121	malli	7
mitta, dim()	54		
mod(), modulo	121	N	
modifioitu sisäinen korkokanta, mirr(		n:s juuri	
)	121	malli	2
modulo, mod()	121	nCr(), kombinaatiot	126
Moninkertaisen lineaarisen		nDerivative(), numeerinen	
regression t-testi	124	derivaatta	127
mRow(), matriisin rivioperaatio	122	negaatio, negatiivisten lukujen	
mRowAdd(), matriisin rivien		syöttäminen	254
kertominen ja yhteenlasku	122	neliöjuuri	
MultReg	122	malli	1
MultRegIntervals()	123	neliöjuuri, #()	185, 235
MultRegTests()	124	nettonykyarvo, npv()	133
murtoluvut		newList(), uusi lista	127
malli	1	newMat(), uusi matriisi	127
propFrac	146	nfMax(), numeerisen funktion	
muunna		maksimi	128
4Grad	91	nfMin(), numeerisen funktion	
yksiköt	243	minimi	128
muuntaminen		nimelliskorko, nom()	129
►Rad	150	nimittäjä	28
muutoin, Else	92	nInt(), numeerinen integraali	128
		nollakohdat, zeroes()	212

nom), muuta efektiivinen korko	
nimelliskoroksi	129
norm(), Frobeniusin normi	130
normaalijakauman todennäköisyys,	
normCdf()	131
normaalisuora, normalLine()	130
normalLine()	130
normCdf()	131
normPdf()	131
nPr(), permutaatiot	132
npv(), nettonykyarvo	133
nSolve(), numeerinen ratkaisu	133
numeerinen	
derivaatta, nDeriv()	128
derivaatta, nDerivative()	127
integraali, nInt()	128
ratkaisu, nSolve()	133
näppäimistön pikavalinnat	251
näyttö	
asteina/minuutteina/sekun	
teina, ►DMS	57
näyttö pallonmuotoisena vektorina,	
►Sphere	184
näyttö vektorina	
sylinterikoordinaatistossa,	
►Cylind	44
näytä	
asteina/minuutteina/sekunteina,	
►DMS	57
binaarisena, 4Kantaluku2	18
desimaalikonaislukuna,	
4Kantaluku10	19
desimaalikulmana, ►DD	48
heksadesimaalilukuna,	
4Kantaluku16	20
pallonmuotoinen vektori,	
►Sphere	184
polaarinen vektori, 4Polaarinen	139
suorakulmavektorina, ►Rect ...	154
vektorina	
sylinterikoordinaatistos	
sa, 4Cylind	44
näytä tiedot, Disp	55, 167

O

objektit	
luo pikavalinnat kirjastoon	101
Odota-komento	209
ohita virhe, PassErr	138
ohjelmat	
julkisen kirjaston määrittäminen	50
yksityisen kirjaston	
määrittäminen	49
ohjelmat ja ohjelmointi	
kokeilun loppu, EndTry	201
lopeta ohjelma, EndPrgm	144
näytä I/O-näyttö, Disp	55, 167
poista virhe, ClrErr	26
try, Try	201
ohjelmointi	
määritä ohjelma, Prgm	144
näytä tiedot, Disp	55, 167
ohita virhe, PassErr	138
oikea, right()	96, 159-160
ominaisarvo, eigVl()	62
ominaisvektori, eigVc()	61
on yhtä kuin, =	227
OneVar, yhden muuttujan tilastot .	134
operaattorit	
sievennysjärjestys	253
ord(), numeerinen merkkikoodi ...	137

P

P►Rx(), x:n suorakulmakoordinaatti	137
P►Ry(), y:n suorakulmakoordinaatti	137
paikallinen muuttuja, Local	111
paikallinen, Local	111
Palaa, return	159
paloittain määritelty funktio (2-	
osainen)	
malli	2
paloittain määritelty funktio (N-	
osainen)	
malli	3
PassErr, ohita virhe	138
Pdf()	78
permutaatiot, nPr()	132
piecewise()	139

pienempi tai yhtä suuri kuin, {	229
pienin yhteinen jaettava, lcm	100
pikavalinnat, näppäimistö	251
piste	
erotus, -	225
osamäärä, ./	226
potenssi, .^	226
summa, .+	225
tulo, .*	226
tulo, dotP()	60
poissCdf()	139
poissPdf()	139
poista	
poista, ClrErr	26
tyhjät elementit listasta	51
poistaminen	
muuttuja, DelVar	51
polaarinen	
koordinaatti, R►Pr()	150
koordinaatti, R►Pθ()	150
vektorinäyttö, 4Polaarinen	139
polyCoeff()	140
polyDegree()	141
polyEval(), laske polynomi	141
polyGcd()	142
polynomit	
laske, polyEval()	141
satunnainen, randPoly()	153
PolyRoots()	143
potenssi, ^	224
potenssiregressio,	
PowerReg	143, 157-158, 196
PowerReg, potenssiregressio	143
Prgm, määritä ohjelma	144
prodSeq()	145
product(), tulo	145
propFrac, varsinainen murtoluku ..	146
prosentti, %	227
PyydäMerkkij	158
Pyyntö	157
pyöristä, round()	163
päivämäärien väliset päivät, dbd() ..	47
pääoman maksuerien summa	237

Q

QR-hajottaminen, QR	147
QR, QR-hajottaminen (tekijöihin jako)	147
QuadReg, 2. asteen regressio	147
QuartReg, 4. asteen regressio	149

R

R, radiaani	239
R►Pr(), polaarinen koordinaatti	150
R►Pθ(), polaarinen koordinaatti ...	150
radiaani, R	239
rahan aika-arvo, Korko	204
rahan aika-arvo, maksuerien lukumäärä	204
rahan aika-arvo, maksuerä	204
rahan aika-arvo, nykyarvo	205
rahan aika-arvo, Tuleva arvo	203
raja-arvo	
lim()	102
limit()	102
malli	7
rajoitettava operaattori "I"	245
rajoitettava operaattori, laskemisjärjestys	253
rand(), satunnaisluku	151
randBin, satunnaisluku	151
randInt(), satunnainen kokonaisluku	151
randMat(), satunnaismatriisi	152
randNorm(), satunnainen normaalijakauma	152
randPoly(), satunnaispolynomi	153
randSamp()	153
RandSeed, satunnaisluvun siemenluku	153
ratkaise, solve()	180
ratkaisu, deSolve()	52
reaali, real()	153
real(), reaali	153
ref(), rivi-echelon-muoto	154
RefreshProbeVars	156
regressiot	
2. asteen, QuadReg	147

3. asteen, CubicReg	42	sec(), sekantti	165
4. asteen, QuartReg	149	sech ⁻¹ (), käänteinen hyperbolinen sekantti	166
eksponentiaalinen, ExpReg	69	sech(), hyperbolinen sekantti	166
lineaarinen regressio, LinRegAx .	104	sekaluvut, käyttäen funktiota propFrac()	146
lineaarinen regressio, LinRegBx	103, 105	sekuntimuoto, "	241
logaritminen, LnReg	110	sekvenssi, seq()	167
Logistinen	113	seq(), sekvenssi	167
logistinen, Logistic	114	seqGen()	168
mediaani-mediaani-suora, MedMed	118	seqn()	169
MultReg	122	sequence, seq()	168-169
potenssiregressio, PowerReg	143, 157-158, 196	series(), sarja	169
sinimuotoinen, SinReg	179	setMode(), aseta tila	171
remain(), jäänös	157	shift(), siirrä	173
return, Palaa	159	sievennetty rivi-echelon-muoto, rref()	164
right(), oikea	159	sievennysjärjestys	253
right, right()	29, 63, 210	sign(), etumerkki	174
ristitulo, crossP()	38	siirry, Goto	91
rivi-echelon-muoto, rref()	154	silmukan loppu, EndLoop	115
rk23(), Runge Kutta -funktio	160	silmukka, Loop	115
rotate(), kierrä	161-162	simult(), samanaikaiset yhtälöt	175
round(), pyöristä	163	sin ⁻¹ (), arkussini	177
rowAdd(), matriisin rivien yhteenlasku	164	sin(), sini	176
rowDim(), matriisin riviakko	164	sinh ⁻¹ (), hyperbolinen arkussini	178
rowNorm(), matriisin ravinormi	164	sinh(), hyperbolinen sini	178
rowSwap(), matriisin ravinvaihto	164	sini näytä kulman	176
rref(), sievennetty rivi-echelon- muoto	164	sini, sin()	176
ryhmät, lukitseminen ja vapauttaminen	111, 208	siniregressio, SinReg	179
ryhmät, lukitustilan testaaminen	87	SinReg, siniregressio	179
		solve(), ratkaise	180
S		SortA, lajittele nousevaan järjestykseen	184
samanaikaiset yhtälöt, simult()	175	SortD, lajittele laskevaan järjestykseen	184
sarja, series()	169	sqrt(), neliöjuuri	185
satunnainen matriisi, randMat()	152	stat.results	186
normaalijakauma, randNorm() .	152	stat.values	187
polynomi, randPoly()	153	stdDevPop(), perusjoukon keskihajonta	187
siemenluku, RandSeed	153	stdDevSamp(), otoksen keskihajonta	188
satunnaisotos	153	Stop-komento	189
sec ⁻¹ (), käänteissekantti	165		

string(), lauseke merkkijonoksi	189	tekijä, factor()	71
strings		Teksti-komento	196
right, right()	29, 63, 210	Test_2S, 2 otoksen F-testi	79
studentint-todennäköisyysjakauma,		tExpand(), trigonometrinen	
tCdf()	195	laajennus	196
studentint-todennäköisyystiheys,		tila-asetukset, getMode()	88
tPdf()	200	tilastot	
subMat(), alimatriisi	189, 191	kahden muuttujan tulokset,	
sum(), yhteenlasku	190	TwoVar	205
sumIf()	190	kertoma, !	232
summa (G)		keskiarvo, mean()	117
malli	5	keskihajonta, stdDev() ...	187-188, 208
summa, S()	236	kombinaatiot, nCr()	126
sumSeq()	191	mediaani, median()	118
suorakulmavektorinäyttö, ►Rect ...	154	permutaatiot, nPr()	132
suurempi kuin, >	230	satunnainen normaali-jakauma,	
suurempi tai yhtä suuri kuin, 	230	randNorm()	152
suurin yhteinen jakaja, gcd()	81	satunnainen siemenluku,	
syöte, Input	94	RandSeed	153
		varianssi, variance()	208
		yhden muuttujan tilastot,	
		OneVar	134
T		tilat	
t-testi, tTest	202	asetus, setMode()	171
T, transponoi	191	TInterval, t-luottamusväli	197
tai (Boolean), tai	136	TInterval_2Samp, kahden otoksen t-	
tai, Boolean operaattori	136	luottamusvälin	198
tallentaminen		ΔtmpCnv() [tmpCnv]	199
symboli, &	246-247	tmpCnv()	199
talouslaskentafunktiot, tvnFV() ...	203	todennäköisyystiheys, normPdf() ..	131
talouslaskentafunktiot, tvnI()	204	toinen derivaatta	
talouslaskentafunktiot, tvnN()	204	malli	6
talouslaskentafunktiot, tvnPmt() ..	204	tPdf(), studentint-	
talouslaskentafunktiot, tvnPv() ...	205	todennäköisyystiheys	200
tan ⁻¹ () , arkustangentti	192	trace()	200
tan(), tangentti	192	transponoi, T	191
tangentLine()	193	trigonometrinen keruu, tCollect() ..	195
tangentti, tan()	192	trigonometrinen laajennus, tExpand	
tangenttisuora, tangentLine()	193	()	196
tanh ⁻¹ () , hyperbolinen areatangentti	194	Try, virheenkäsittelykomento	201
tanh(), hyperbolinen tangentti	193	tTest, t-testi	202
taylor(), Taylorin polynomi	195	tTest_2Samp, kahden otoksen t-testi	202
Taylorin polynomi, taylor()	195	tulo (P)	
tCdf(), studentint-		malli	6
todennäköisyysjakauma ..	195	tulo, P()	235
tCollect(), trigonometrinen keruu ..	195		

Z

zeroes(), nollakohdat	212
zInterval, z-luottamusväli	215
zInterval_1Prop, yhden osuuden z- luottamusväli	216
zInterval_2Prop, kahden osuuden z- luottamusväli	216
zInterval_2Samp, kahden näytteen z- luottamusväli	217
zTest	218
zTest_1Prop, yhden osuuden z-testi	218
zTest_2Prop, kahden osuuden z-testi	219
zTest_2Samp, kahden otoksen z-testi	220

X

χ^2 Cdf()	25
χ^2 GOF	25
χ^2 Pdf()	26