

TI-Nspire™ CX CAS Sovelluksen käsikirja

Tärkeitä tietoja

Texas Instruments ei myönnä minkäänlaista nimenomaista tai oletettua takuuta mukaan lukien rajoituksetta oletetut takuut soveltuvuudesta kaupankäynnin kohteeksi tai sopivuudesta tiettyyn tarkoitukseen koskien ohjelmia tai kirjallista aineistoa, jotka annetaan saataville "sellaisina kuin ne ovat". Texas Instruments ei ole missään tapauksessa vastuussa näiden aineistojen hankinnasta tai käytöstä aiheutuvista erityisistä, rinnakkaisista, satunnaisista tai välillisistä vahingoista. Kannemuodosta riippumatta Texas Instrumentsin vastuu rajoittuu vain ja ainoastaan kyseisen tuotteen hankintahintaan. Texas Instruments ei myöskään ole velvoitettu vastaamaan minkäänlaisiin vaatimuksiin johtuen näiden materiaalien käytöstä muiden osapuolten toimesta.

© 2026 Texas Instruments Incorporated

Todelliset tuotteet saattavat erota hieman mukana tulevista kuvista.

Sisältö

Lausekemallit	1
Luettelo aakkosjärjestyksessä	8
A	8
B	17
C	21
D	48
E	61
F	72
G	82
I	93
L	102
M	119
N	128
O	138
P	141
Q	150
R	154
S	169
T	197
U	213
V	214
W	215
X	217
Z	219
Symbolit	228
TI-Nspire™ CX II – Piirtokomennot	256
Kuvaajien ohjelmointi	256
Kuvaajanäyttö	256
Oletusnäkymä ja asetukset	257
Kuvaajanäytön virheviestit	258
Virheelliset komennot kuvaajatilassa	258
C	260
D	261
F	264
G	266
P	267
J	269
I	271

Tyhjät elementit	272
Matemaattisten lausekkeiden syöttäminen pikavalintojen avulla	274
EOS-järjestelmän (yhtälökäyttöjärjestelmä) hierarkia	276
TI-Nspire CX II – TI-Basic-ohjelmointiominaisuudet	278
Automaattinen sisennys Ohjelmointieditorissa	278
TI-Basicin virheviestien parannukset	278
Vakiot ja arvot	281
Virhekoodit ja viestit	282
Varoituskoodit ja -viestit	290
Yleistä	292
Indeksi	293

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 230.

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 231.

Esimerkki:

$$2^3 = 8$$

Neliöjuurimalli

ctrl **x²** painikkeet



Huomaa: Katso myös √() (neliöjuuri), sivu 242.

Esimerkki:

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

N:s juuri -malli

ctrl **^** **painikkeet**

$\sqrt{}$

$\sqrt{}$ **Huomaa:** Katso myös **root()**, sivu 166.

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

e-kantainen eksponenttifunktio korotettuna potenssiin

Huomaa: Katso myös **e^()**, sivu 62.

Esimerkki:

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

Logaritmimalli

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

log $\left(\right)$

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

Huomaa: Katso myös **log()**, sivu 115.

Esimerkki:

$$\log \left(2. \right) \quad 0.5$$
$$\log \left(4 \right)$$

Paloittain määritellyn funktion malli (2-osainen)

Katalogi > $\left| \frac{a}{b} \right|$

$\left\{ \begin{array}{l} \\ \end{array} \right.$

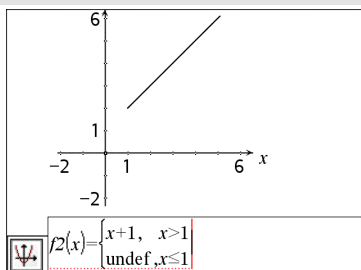
Esimerkki:

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

Huomaa: Katso myös **piecewise()**, sivu 142.

Paloittain määritellyn funktion malli (2-osainen)

Katalogi >

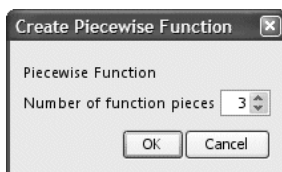


Paloittain määritellyn funktion malli (N-osainen)

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 142.

Yhtälöparin malli

Katalogi >



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

Huomaa: Katso myös `system()`, sivu 197.

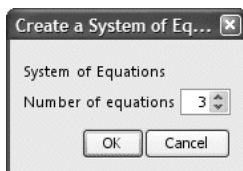
Esimerkki:

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

N-osaisten 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 197.

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 > 



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

Esimerkki:

$$30^{\circ}15'10'' \quad \frac{10891 \cdot \pi}{64800}$$

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 > 

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

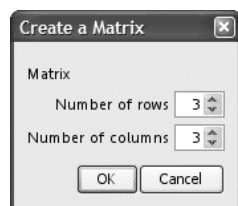
Esimerkki:

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

Matriisimalli (m x n)

Katalogi > 

Malli tulee näkyviin määritettyäsi rivien ja sarakkeiden lukumäärän syöttöruutuun.



Create a Matrix

Matrix

Number of rows

Number of columns

OK Cancel

Esimerkki:

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

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 243.

Tulon malli (Π)

Katalogi > 

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

Huomaa: Katso myös $\Pi()$ (**prodSeq**), sivu 242.

Esimerkki:

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

Ensimmäisen derivaatan malli

Katalogi > 

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

Ensimmäisen derivaatan mallia voi käyttää myös laskettaessa ensimmäinen derivaatta pisteessä.

Huomaa: Katso myös $d()$ (**derivaatta**), sivu 239.

Esimerkki:

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

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

Toisen derivaatan malli

Katalogi > 

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

Toisen derivaatan mallia voi käyttää myös laskettaessa toinen derivaatta pisteessä.

Huomaa: Katso myös $d()$ (**derivaatta**), sivu 239.

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 > 

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

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

Esimerkki:

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

N:nen derivaatan malli

Katalogi > 

Huomaa: Katso myös **d()** (derivaatta), sivu 239.

Määrätyn integraalin malli

Katalogi > 

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

Esimerkki:

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

Huomaa: Katso myös **f()** integraali(), sivu 240.

määrittämättömän integraalin malli

Katalogi > 

$$\int f(x) dx$$

Esimerkki:

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

Huomaa: Katso myös **f()** integral(), sivu 240.

Raja-arvon malli

Katalogi > 

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

Esimerkki:

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

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

Huomaa: Katso myös **limit()**, sivu 104.

Luettelo aakkosjärjestyksessä

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

A

abs()

Katalogi >

abs(*LausI*)⇒*lauseke*

abs(*ListaI*)⇒*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.

$\left\{\left\{\frac{\pi}{2}, \frac{\pi}{3}\right\}\right\}$	$\left\{\frac{\pi}{2}, \frac{\pi}{3}\right\}$
$2-3 \cdot i$	$\sqrt{13}$
$ 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 211.

- 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**.

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	

- 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 244, 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 näky kantalukutilan 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 kantalukutilassa:

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

Tärkeää: Nolla, ei O-kirjain.

Binaarisessa kantalukutilassa:

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

Desimaalisessa kantalukutilassa:

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 syötteenä voi olla korkeintaan 64 numeroa (etuliitettä 0b ei lasketa). Heksadesimaalisessa syötteenä voi olla korkeintaan 16 numeroa.

angle()

angle(LausI) ⇒ lauseke

Laskee argumentin kulman tulkiten argumentin kompleksiluvuksi.

Huomaa: Kaikkia määrittämättömiä muuttujia käsitellään reaali muuttujina.

Astekulmatilassa:

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

Graadikulmatilassa:

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

Radiaanikulmatilassa:

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

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

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

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

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 *List1,Lista2[,Lista3,...,Lista20]*
[,Lippu]

Suorittaa yksisuuntaisen varianssianalyysin 2-20 perusjoukon keskiarvon vertailua varten. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Lippu=0 datalle, *Lippu=1* tilastoille

Tulosmuuttuja	Kuvaus
stat.F	F-tilaston arvo
stat.PVal	Alin merkitsevyytaso, jolla nollahypoteesi voidaan hylätä
stat.df	Ryhmien vapausasteet
stat.SS	Ryhmien neliöiden summa
stat.MS	Ryhmien 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 191.)

TasoRivi=0 lohkolle

TasoRivi=2,3,...,Pit-1, kahdelle tekijälle, jossa $Pit = \text{pituus}(List1) = \text{pituus}(List2) = \dots = \text{pituus}(List10)$ ja $Pit / TasoRivi \in \{2,3,\dots\}$

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.dfInteract	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

approx(1/3)0.333333

Määrittää argumentin sievennetyn arvon lausekkeena, joka sisältää desimaaliarvoja, mikäli mahdollista, riippumatta nykyisestä Automaattinen tai likimääräinen -tilasta.

approx({1/3, 1/9}){0.333333, 0.111111}

Tämä vastaa argumentin syöttämistä ja painikkeen ctrl enter painamista.

approx({sin(π), cos(π)}){0., -1.}

approx([sqrt(2), sqrt(3)])[1.41421 1.73205]

approx([1/3, 1/9])[0.333333 0.111111]

approx({sin(π), cos(π)}){0., -1.}

approx([sqrt(2), sqrt(3)])[1.41421 1.73205]

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.

$$\frac{5}{6}$$

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

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 34.

arccosh()

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

arccot()Katso $\cot^{-1}()$, sivu 36.**arccoth()**Katso $\coth^{-1}()$, sivu 37.**arccsc()**Katso $\csc^{-1}()$, sivu 39.**arccsch()**Katso $\csch^{-1}()$, sivu 40.**arcLen()**Katalogi > **arcLen(LausI,Muutt,Alku,Loppu)**
 $\Rightarrow$ lausekeLaskee *LausI*:n kaaren pituuden alusta
Alku loppuun *Loppu* muuttujan *Muutt*
suhteen.Kaaren pituus lasketaan kokonaislukuna
käyttäen oletuksena funktiotilan
määrittystä.**arcLen(ListaI,Muutt,Alku,Loppu)** $\Rightarrow$ listaLaskee listan jokaisen *Listai*:n elementin
kaaren pituuden alusta *Alku* loppuun
Loppu muuttujan *Muutt* suhteen. $\text{arcLen}(\cos(x),x,0,\pi)$ 3.8202 $\text{arcLen}(f(x),x,a,b)$ $\int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} dx$ $\text{arcLen}(\{\sin(x),\cos(x)\},x,0,\pi)$
 $\{3.8202,3.8202\}$ **arcsec()**Katso $\sec^{-1}()$, sivu 170.**arcsech()**Katso $\text{sech}^{-1}()$, sivu 171.

arcsin()**Katso $\sin^{-1}()$, sivu 182.****arcsinh()****Katso $\sinh^{-1}()$, sivu 183.****arctan()****Katso $\tan^{-1}()$, sivu 198.****arctanh()****Katso $\tanh^{-1}()$, sivu 200.****augment()****Katalogi > ****augment(Lista1, Lista2)⇒lista**
$$\text{augment}(\{1, -3, 2\}, \{5, 4\}) \quad \{1, -3, 2, 5, 4\}$$

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

augment(Matriisi1, Matriisi2)⇒matriisi

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

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

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

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

Lausl 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 keskeiserotusosamäärää.

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, täältä.

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

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

- 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*

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

oletusarvot ovat samat kuin TVM-funktioilla.

pyörArvo määrittää pyöristyksessä käytettävien desimaalien määrän.
Oletusarvo=2.

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 täältä.

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

► 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
0hFFFFFFFFFFFFFFF 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 kantalukutilassa
 0b100...000 (63 zeros) binaarisessa kantalukutilassa

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

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

► 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.

0b10011 ► Base10	19
0h1F ► Base10	31

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

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

►Base16 (►Kantaluku16)

Kokonaisluku1

►Base16⇒*kokonaisluku*

256►Base16	0h100
0b111100001111►Base16	0hF0F

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

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 desimaalikonaisluvun, 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($n,p,alaraja,yläraja$)⇒*luku*, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

binomCdf($n,p,yläraja$)kun $P(0 \leq X \leq yläraja) \Rightarrow$ *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 $P(X \leq yläraja)$, aseta *alaraja*=0

binomPdf()

binomPdf(n,p)⇒*lista*

binomPdf($n,p,XVal$)⇒*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()**

ceiling(*Lausl*)⇒*kokonaisluku*

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

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

Argumentti voi olla reaali- tai kompleksiluku.

Huomaa: Katso myös **floor()**.

ceiling(Lista1) $\Rightarrow$ lista

ceiling(Matriisi1) $\Rightarrow$ matriisi

Laskee listan tai matriisin jokaisen elementin ylärajasta.

$\text{ceiling}(\{-3.1, 1, 2.5\})$	$\{-3., 1, 3.\}$
$\text{ceiling}\left(\begin{bmatrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{bmatrix}\right)$	$\begin{bmatrix} 0 & -3. \cdot i \\ 2. & 4 \end{bmatrix}$

centralDiff()

centralDiff(Laus1, Muutt [=Arvo] [,Askel]) $\Rightarrow$ lauseke

centralDiff(Laus1, Muutt [,Askel]) | **Muutt=Arvo** $\Rightarrow$ lauseke

centralDiff(Laus1, Muutt [=Arvo] [,Lista]) $\Rightarrow$ lista

centralDiff(Lista1, Muutt [=Arvo] [,Askel]) $\Rightarrow$ lista

centralDiff(Matriisi1, Muutt [=Arvo] [,Askel]) $\Rightarrow$ matriisi

Laskee numeerisen derivaatan käyttäen keskeiserotusosamäärän kaavaa.

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

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

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

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

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

cFactor(*LausI* [, *Muutt*]) $\Rightarrow$ lauseke

cFactor(*Listal* [, *Muutt*]) $\Rightarrow$ lista

cFactor(*MatriisiI* [, *Muutt*]) $\Rightarrow$ 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.

cFactor(*LausI* [, *Muutt*]) jakaa *LausI*:n tekijöihin muuttujan *Muutt* suhteen.

LausI: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.

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

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

cFactor()Katalogi > 

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()**.

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

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

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

$$(x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x + \dots)$$

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()Katalogi > 

charPoly
(*neliömatrissi*, *Muutt*) ⇒
polynomilauseke

charPoly
(*neliömatrissi*, *Laus*) ⇒ *polynomilauseke*

charPoly
(*neliömatrissi1*, *Matrissi2*) ⇒
polynomilauseke

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

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

jossa $\mathbf{I}$ tarkoittaa identtistä matriisia $n \times n$.

neliömatrissi1:n ja *neliömatrissi2*:n on oltava samankokoiset.

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

χ^2 2way ObsMatriisi

chi22way ObsMatriisi

Laskee χ^2 -testin tarkasteltavan matriisin ObsMatriisi sisältämän kaksisuuntaisen lukemataulukon arvojen välisestä assosiaatiosta. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Lisätietoja matriisissa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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

 χ^2 Cdf()

χ^2 Cdf(alaraja,yläraja,df) $\Rightarrow$ luku, jos alaraja ja yläraja ovat lukuja, lista, jos alaraja ja yläraja ovat listoja

chi2Cdf(alaraja,yläraja,df) $\Rightarrow$ 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 tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

χ^2 GOF *obsLista,expLista,df*chi2GOF *obsLista,expLista,df*

Suorittaa testin, jolla varmistetaan, että otoksen data on tiettyä jakaumaa vastaavasta perusjoukosta. *obsList* on lukemalista, ja sen tulee sisältää kokonaislukuja. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Lisätietoja listassa olevien tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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

 χ^2 Pdf()

χ^2 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 *XArvon* arvolla määritetylle vapausasteelle *df*.

Lisätietoja listassa olevien tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

clearAZ

Katalogi > 

clearAZ

$5 \rightarrow b$	5
b	5
ClearAZ	Done
b	b

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 214.

ClrErr

Katalogi > 

ClrErr

Esimerkki **ClrErr**-komennosta, katso esimerkki 2 **Try**-komennon kohdalla, sivu 207.

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 142, ja **Try**, sivu 207.

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

colAugment()

Katalogi > 

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

Luo uuden matriisin, joka on *Matriisi2* liitettynä *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)**⇒*lauseke*

Laskee *Matriisin* sisältämien sarakkeiden lukumäärän.

Huomaa: Katso myös **rowDim()**.

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

colNorm()Katalogi > **colNorm(Matriisi)**⇒*lauseke*

Laskee maksimiarvon *Matriisin* sarakkeissa olevien elementtien itseisarvojen summista.

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

$$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix} \rightarrow \text{mat} \quad \begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$$

$$\text{colNorm}(\text{mat}) \quad 9$$

comDenom()Katalogi > **comDenom(LausI[,Muutt])**⇒*lauseke***comDenom(ListaI[,Muutt])**⇒*lista*

comDenom(MatriisiI[,Muutt])⇒*matriisi*

comDenom(LausI) supistaa täydellisesti lavennetun osoittajan täydellisesti lavennetulla nimittäjällä.

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

$$\frac{x^2 \cdot y^2 + x^2 \cdot y + 2 \cdot x \cdot y^2 + 2 \cdot x \cdot y + 2 \cdot y^2 + 2 \cdot y}{x^2 + 2 \cdot x + 1}$$

comDenom(*Laus1*, *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 *Laus1*:ssä, **comDenom**(*Laus1*, *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 määrittäminen ja kokeile sitä rutiininomaisesti vaihtoehtona funktioille **comDenom()** ja **factor()**.

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

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

$$\text{Define } \text{comden}(\text{exprn}) = \text{comDenom}(\text{exprn}, \text{abc}) \\ \text{Done}$$

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

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

completeSquare ()

completeSquare(*ExprOrEqn*, *Var*) ⇒ lauseke tai yhtälö

completeSquare(*ExprOrEqn*, *Var*^{Power}) ⇒ lauseke tai yhtälö

completeSquare(*ExprOrEqn*, *Var1*, *Var2* [...]) ⇒ lauseke tai yhtälö

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

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

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

completeSquare ()

Katalogi > 

completeSquare(*ExprOrEqn*, {*Var1*, *Var2* [...]}) $\Rightarrow$ *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 vakiomuodossa 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.

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

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

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

conj()

Katalogi > 

conj(*Laus1*) $\Rightarrow$ *lauseke*

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

conj(*Matriisi1*) $\Rightarrow$ *matriisi*

Laskee argumentin liittokompleksiluvun.

Huomaa: Kaikkia määrittämättömiä muuttujia käsitellään reaali muuttujina.

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

constructMat()

Katalogi > 

constructMat

(*Laus*, *Muutt1*, *Muutt2*, *numRivit*, *numSarakkeet*)

$\Rightarrow$ *matriisi*

Laskee matriisin argumentteihin perustuen.

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

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 - numSarakeet**.

CopyVar

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.

MuuttI: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

Muutt1. tulee olla olemassa olevan muuttujaryhmän nimi, kuten tilastollinen *stat.nn* 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.

```
getVarInfo()
aa.a "NUM" "0" 0
aa.b "NUM" "0" 0
bb.a "NUM" "0" 0
bb.b "NUM" "0" 0
```

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.

$$\left(\sin(x)\right)^2 \rightarrow \cos \qquad 1 - \left(\cos(x)\right)^2$$

cos()

cos(*Laus1*) ⇒ lauseke

cos(*Lista1*) ⇒ lista

cos(*Laus1*) määrittää argumentin kosinin lausekkeena.

cos(*Lista1*) määrittää listan kaikkien *Lista1*:n sisältämien elementtien kosineista.

Astekulmatilassa:

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

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

Graadikulmatilassa:

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

Radiaanikulmatilassa:

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

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

cos(neliömatriisi) ⇒ neliömatriisi

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

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

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

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

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:

$\cos(B) =$

Radiaanikulmatilassa:

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

cos() **painike**

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

Kaikki laskut suoritetaan liukulukuaritmetiikalla.

cos⁻¹() **painike**

cos⁻¹(LausI) ⇒ lauseke

Astekulmatilassa:

cos⁻¹(ListaI) ⇒ lista

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

cos⁻¹(LausI) määrittää lausekkeena kulman, jonka kosini on *LausI*.

Graadikulmatilassa:

cos⁻¹(ListaI) laskee listan *ListaI*: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** (...).

$$\cos^{-1}(\{0,0,2,0,5\}) \quad \left\{ \frac{\pi}{2}, 1.36944, 1.0472 \right\}$$

cos⁻¹(neliömatrissiI) ⇒ neliömatrissi

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

$$\cos^{-1} \left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \right) \quad \begin{bmatrix} 1.73485+0.064606 \cdot i & -1.49086+2.10514 \\ -0.725533+1.51594 \cdot i & 0.623491+0.778369 \\ -2.08316+2.63205 \cdot i & 1.79018-1.27182 \cdot i \end{bmatrix}$$

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

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

cosh()**Katalogi** > 

cosh(LausI) ⇒ lauseke

Astekulmatilassa:

cosh(ListaI) ⇒ lista

cosh()Katalogi > 

cosh(*LausI*) määrittää argumentin hyperbolisen kosinin lausekkeena.

cosh(*ListaI*) määrittää listan *ListaI*:n kunkin elementin hyperbolisista kosineista.

cosh(*neliömatriisiI*) $\Rightarrow$ *neliömatriisi*

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

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

$$\cosh\left(\left(\frac{\pi}{4}\right)_r\right) \qquad \cosh(45)$$

Radiaanikulmatilassa:

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

cosh⁻¹()Katalogi > 

cosh⁻¹(*LausI*) $\Rightarrow$ *lauseke*

cosh⁻¹(*ListaI*) $\Rightarrow$ *lista*

cosh⁻¹(*LausI*) määrittää argumentin käänteisen hyperbolisen kosinin lausekkeena.

cosh⁻¹(*ListaI*) määrittää listan *ListaI*:n kunkin elementin käänteisistä hyperbolisista kosineista.

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

cosh⁻¹(*neliömatriisiI*) $\Rightarrow$ *neliömatriisi*

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

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

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

$$\cosh^{-1}(\{1,2,1,3\}) \qquad \{0,1.37286,\cosh^{-1}(3)\}$$

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

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

cot() **painike****cot**(*Laus1*) ⇒ *lauseke***cot**(*Listal*) ⇒ *lista*

Laskee *Laus1*:n kotangentin tai määrittää listan *Listal*:n kaikkien elementtien kotangenteista.

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

Astekulmatilassa:

$\cot(45)$	1
------------	---

Graadikulmatilassa:

$\cot(50)$	1
------------	---

Radiaanikulmatilassa:

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

cot⁻¹() **painike****cot⁻¹**(*Laus1*) ⇒ *lauseke***cot⁻¹**(*Listal*) ⇒ *lista*

Laskee kulman, jonka kotangenti on *Laus1*, tai määrittää listan, joka sisältää *Listal*:n kunkin elementin käänteiskotangentit.

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

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

Astekulmatilassa:

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

Graadikulmatilassa:

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

Radiaanikulmatilassa:

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

coth()**Katalogi** > **coth**(*Laus1*) ⇒ *lauseke***coth**(*Listal*) ⇒ *lista*

Laskee *Laus1*:n hyperbolisen kotangentin tai määrittää listan *Listal*:n kaikkien elementtien hyperbolisista kotangenteista.

$\coth(1.2)$	1.19954
--------------	---------

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

coth⁻¹()**Katalogi** > **coth⁻¹(*Laus1*)**⇒*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 *Laus1*:n käänteisen hyperbolisen kotangentin tai määrittää *listan*, joka sisältää *Listal*:n kaikkien elementtien käänteiset hyperboliset kotangentit.

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

count()**Katalogi** > **count(*Arvo1taiListal* [,*Arvo2taiListal2* [...]])**⇒*arvo***count(2,4,6)** 3

Laskee elementtien kokonaismäärän argumenteille, jotka sievennyvät numeroarvoiksi.

count({2,4,6}) 3

Argumentit voivat olla lausekkeita, arvoja, listoja tai matriiseja.

Argumenttien datatyypit voivat olla erilaisia, ja argumentit voivat olla erikokoisia.

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

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

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

Listat & Taulukot -sovelluksessa voit käyttää solualueita argumenttien tilalla.

Tyhjiä elementtejä ei huomioida.

Lisätietoja tyhjistä elementeistä, katso sivu 272.

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

countif()**Katalogi** > **countif(*Listal*,*Kriteerit*)**⇒*arvo***countif({1,3,"abc",undef,3,1},3)** 2

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

Laskee niiden elementtien lukumäärän, jotka ovat yhtä kuin 3.

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 tyhjästä elementistä, katso sivu 272.

Huomaa: Katso myös **sumlf()**, sivu 196, ja **frequency()**, sivu 80.

$$\text{countIf}(\{ \text{"abc"}, \text{"def"}, \text{"abc"}, 3, \text{"def"} \}) \quad 1$$

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

$$\text{countIf}(\{ \{ x^{-2}, x^{-1}, 1, x, x^2 \}, x \}) \quad 1$$

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

$$\text{countIf}(\{ \{ 1, 3, 5, 7, 9 \}, ? < 5 \}) \quad 2$$

Laskee lukumäärään 1:n ja 3:n.

$$\text{countIf}(\{ \{ 1, 3, 5, 7, 9 \}, 2 < ? < 8 \}) \quad 3$$

Laskee lukumäärään 3:n, 5:n ja 7:n.

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

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

cPolyRoots()

cPolyRoots(Poly, Muutt) ⇒ lista

cPolyRoots(Kertoinlista) ⇒ lista

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

Poly on oltava polynomi yhdessä muuttujassa.

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

Huomaa: Katso myös **polyRoots()**, sivu 147.

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

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

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

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

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

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

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

crossP(Vektori1, Vektori2) ⇒ vektori

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

Sekä *Vektori1*:n että *Vektori2*:n on oltava rivivektoreita tai sarakevektoreita. Vektoreiden on oltava samankokoiset, ja koon tulee olla joko 2 tai 3.

$$\begin{array}{l} \text{crossP}(\{a1, b1\}, \{a2, b2\}) \\ \hline \{0, 0, a1 \cdot b2 - a2 \cdot b1\} \\ \text{crossP}(\{0.1, 2.2, -5\}, \{1, -0.5, 0\}) \\ \hline \{-2.5, -5., -2.25\} \end{array}$$

$$\begin{array}{l} \text{crossP}([1 \ 2 \ 3], [4 \ 5 \ 6]) \\ \hline [-3 \ 6 \ -3] \\ \text{crossP}([1 \ 2], [3 \ 4]) \\ \hline [0 \ 0 \ -2] \end{array}$$

csc() **painike****csc(Laus1) ⇒ lauseke****csc(Lista1) ⇒ lista**

Laskee *Laus1*:n kosekantin tai määrittää listan, joka sisältää *List1*:n kaikkien elementtien kosekantit.

Astekulmatilassa:

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

Graadikulmatilassa:

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

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(Laus1) ⇒ lauseke****csc⁻¹(Lista1) ⇒ lista**

Astekulmatilassa:

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

Graadikulmatilassa:

csc⁻¹()**trig** painike

Laskee kulman, jonka kosekanti on *Laus1*, tai määrittää listan, joka sisältää *Listal*:n kunkin elementin käänteiskosekantit.

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

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

$$\text{csc}^{-1}(1)$$

100.

Radiaanikulmatilassa:

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

csch()**Katalogi** >

csch(*Laus1*) ⇒ lauseke

$$\text{csch}(3)$$

$$\frac{1}{\sinh(3)}$$

csch(*Listal*) ⇒ lista

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

$$\text{csch}\left(\left\{1, 2, 1, 4\right\}\right)$$

$$\left\{\frac{1}{\sinh(1)}, 0.248641, \frac{1}{\sinh(4)}\right\}$$

csch⁻¹()**Katalogi** >

csch⁻¹(*Laus1*) ⇒ lauseke

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

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

csch⁻¹(*Listal*) ⇒ lista

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

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

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

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

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 reaalitytuloksen kompleksilukumuodossa.

cSolve() asettaa määrittäjäjoukon väliaikaisesti kompleksilukumuotoon yhtälön ratkaisemisen ajaksi, vaikka nykyinen määrittäjäjoukko olisi reaalinen. Kompleksilukujen määrittäjäalueella murtolukueksponenteissa, joiden nimittäjä on pariton luku, käytetään perus- eikä reaalitylukualueella. 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()**.

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

cSolve(Yhtälöryhmä, MuuttTaiArvaus1,

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

Desimaalien näyttötilassa Kiinteä 2:

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

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

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 määrittä 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.

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.

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

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

Polynomisarjoissa laskun suoritus aika tai muistin käyttö voivat riippua merkittävästi ratkaisumuuttujien järjestyksestä. Jos ensimmäinen valintasi kuluttaa muistia, tai et jaksaa 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 ratkaisumuuttujiltaan, **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.

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

$$u=\frac{e^w+i}{2} \text{ and } v=\frac{e^w-i}{2}$$

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

$$w=0.494866 \text{ and } z=0.703467$$

$$\text{cSolve}\left(e^z=w \text{ and } w=z^2, \{w,z=1+i\}\right)$$

$$w=0.149606+4.8919 \cdot i \text{ and } z=1.58805+1.5402 \cdot i$$

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

CubicReg

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

Laskee 3. asteen polynomiregression $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$ listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttuun. (Katso sivu 191.)

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

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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	Muokatun 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	Muokatun 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()

cumulativeSum(Lista1) ⇒ lista

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

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

cumulativeSum()

Katalogi > 

cumulativeSum(Matriisi1)⇒matriisi

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

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

Tyhjä elementti listassa *Listal* tai matriisissa *Matriisi1* tuottaa tyhjän elementin tuloksena olevaan listaan tai matriisiin. Lisätietoja tyhjästä elementistä, katso sivu 272.

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 määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

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

Define <i>g()</i> =Func	Done
Local <i>temp,i</i>	
0 → <i>temp</i>	
For <i>i</i> ,1,100,1	
If <i>i</i> =50	
Cycle	
<i>temp</i> + <i>i</i> → <i>temp</i>	
EndFor	
Return <i>temp</i>	
EndFunc	
<i>g()</i>	5000

►Cylind

Katalogi > 

Vektori ►**Cylind**

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

Näyttää rivi- tai sarakevektorin sylinterin muodossa [*r*,*∠θ*,*z*].

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

$\begin{bmatrix} 2 & 2 & 3 \end{bmatrix} \rightarrow \text{Cylind}$	$\begin{bmatrix} 2\sqrt{2} & \angle \frac{\pi}{4} & 3 \end{bmatrix}$
---	--

cZeros(Laus, Muutt) ⇒ lista

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

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

Desimaalien näyttötilassa Kiinteä 3:

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

$$\{-1.114+1.073\cdot i, -1.114-1.073\cdot i, -2.125, -0.612, 0\}$$

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

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

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

Määritä rivin 2 juuri:

Jokainen tulosmatriisin 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 c_k , 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 jaksa 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.

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

$$\text{cZeros} \left(\left\{ u \cdot v - u - c \cdot v^2, v^2 + u \right\}, \{u, v\} \right) \quad \begin{bmatrix} 0 & 0 \\ -(c-1)^2 & -(c-1) \end{bmatrix}$$

$$\text{cZeros} \left(\left\{ u \cdot v - u - v \cdot v^2 + u \right\}, \{u, v, w\} \right) \quad \text{cZero} \left(\left\{ u \cdot (v-1) - v \cdot u + v^2 \right\}, \{u, v, w\} \right) \quad \begin{bmatrix} 0 & 0 & c4 \\ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & c4 \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & c4 \end{bmatrix}$$

$$\text{cZeros} \left(\left\{ u + v - e^w, u - v - i \right\}, \{u, v\} \right) \quad \begin{bmatrix} e^w + i & e^w - i \\ 2 & 2 \end{bmatrix}$$

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.

$$\left| \text{cZeros} \left(\left\{ e^z - w, w - z^2 \right\}, \left\{ w, z \right\} \right) \right|$$

$$\left[0.494866 \quad -0.703467 \right]$$

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

$$\left| \text{cZeros} \left(\left\{ e^{-z} - w, w - z^2 \right\}, \left\{ w, z = 1 + i \right\} \right) \right|$$

$$\left[0.149606 + 4.8919 \cdot i \quad 1.58805 + 1.54022 \cdot i \right]$$

D

dbd()

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°) ►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°
-------------	-----------------

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 *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 lausekkeitä ja ohjeita (kuten **If**, **Then**, **Else** ja **For**).

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

Huomaa: Katso myös **Define LibPriv**, sivu 51, ja **Define LibPub**, sivu 51.

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 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 49, ja **Define LibPub**, sivu 51.

```
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 49, ja **Define LibPriv**, sivu 51.

DelVar Katalogi >

DelVar <i>Muutt1</i> [, <i>Muutt2</i>] [, <i>Muutt3</i>] ...	$2 \rightarrow a$	2
DelVar <i>Muutt.</i>	$(a+2)^2$	16
Poistaa määritetyn muuttujan tai muuttujaryhmän muistista.	DelVar <i>a</i>	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 214.

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.*

<i>aa.a:=45</i>	45									
<i>aa.b:=5.67</i>	5.67									
<i>aa.c:=78.9</i>	78.9									
getVarInfo()	<table><tr><td><i>aa.a</i></td><td>"NUM"</td><td>"[]"</td></tr><tr><td><i>aa.b</i></td><td>"NUM"</td><td>"[]"</td></tr><tr><td><i>aa.c</i></td><td>"NUM"</td><td>"[]"</td></tr></table>	<i>aa.a</i>	"NUM"	"[]"	<i>aa.b</i>	"NUM"	"[]"	<i>aa.c</i>	"NUM"	"[]"
<i>aa.a</i>	"NUM"	"[]"								
<i>aa.b</i>	"NUM"	"[]"								
<i>aa.c</i>	"NUM"	"[]"								
DelVar <i>aa.</i>	<i>Done</i>									
getVarInfo()	"NONE"									

delVoid() Katalogi >

delVoid (<i>Listal</i>)⇒ <i>lista</i>	delVoid({1,void,3})	{1,3}
--	---------------------	-------

Antaa tuloksena listan, jossa on listan *Listal* sisältö, ja kaikki tyhjät elementit on poistettu.

Lisätietoja tyhjästä elementeistä, katso sivu 272.

deSolve()

Katalogi >

deSolve(*1.astTai2.astODE*, *Muutt*,
riippuvaMuutt) $\Rightarrow$ 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 muuttuajaan 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 ck , jossa k on kokonaisluku 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.

$$\text{deSolve}(y''+2\cdot y'+y=x^2, x, y)$$

$$y=(c3\cdot x+c4)\cdot e^{-x}+x^2-4\cdot x+6$$

$$\text{right}(Ans)\rightarrow temp \quad (c3\cdot x+c4)\cdot e^{-x}+x^2-4\cdot x+6$$

$$\frac{d^2}{dx^2}(temp)+2\cdot\frac{d}{dx}(temp)+temp-x^2 \quad 0$$

$$\text{DelVar } temp \quad Done$$

$$\text{deSolve}(y'=(\cos(y))^2, x, y) \quad \tan(y)=\frac{x^2}{2}+c4$$

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

$$Ans|c4=c-1 \text{ and } n3=0 \quad y=\tan^{-1}\left(\frac{x^2+2\cdot(c-1)}{2}\right)$$

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.

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) ⇒ 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) ⇒ tietty ratkaisu

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

$$\sin(y) = (y \cdot e^x + \cos(y)) \cdot y' \rightarrow ode$$

$$\sin(y) = (e^x \cdot y + \cos(y)) \cdot y'$$

$$deSolve(ode \text{ and } y(0)=0, x, y) \rightarrow soln$$

$$\frac{-(2 \cdot \sin(y) + y^2)}{2} = (e^x - 1) \cdot e^{-x} \cdot \sin(y)$$

$$soln|x=0 \text{ and } y=0 \quad true$$

$$ode|y'=\text{impDif}(soln, x, y) \quad true$$

$$DeIVar \text{ ode, soln} \quad Done$$

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

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

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

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

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

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

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

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

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

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

det()

det(*neliömatrissi* [, *Toleranssi*]) ⇒ lauseke

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

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

$$\det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = a \cdot d - b \cdot c$$

$$\det\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = -2$$

$$\det\left(\text{identity}(3) \cdot x \cdot \begin{pmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ -6 & -2 & 7 \end{pmatrix}\right) = -(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$$

$$\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \text{matI} \quad \begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\det(\text{matI}) = 0$$

$$\det(\text{matI}, 1) = 1. \text{E}20$$

diag()

diag(*Lista*) ⇒ *matriisi*

diag(*rivimatriisi*) ⇒ *matriisi*

diag(*sarakematriisi*) ⇒ *matriisi*

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

$$\text{diag}([2 \ 4 \ 6]) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

diag()Katalogi > **diag(neliömatriisi)⇒rivimatriisi**

Laskee rivimatriisin, joka sisältää elementit *neliömatriisin* päälävistäjästä.

neliömatriisi:n on oltava neliö.

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

dim()Katalogi > **dim(Lista)⇒kokonaisluku**

Laskee *Listan* mitat.

dim(Matriisi)⇒lista

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

dim(Merkkijono)⇒kokonaisluku

Laskee merkkijonon *Merkkijono* sisältämien merkkien lukumäärän.

dim({ 0,1,2 })	3
dim($\begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & 5 \end{pmatrix}$)	{ 3,2 }
dim("Hello ")	5
dim("Hello "&"there ")	11

DispKatalogi > 

Disp lausTaiMerkkijono1 [, lausTaiMerkkijono2] ...

Näyttää *Laskin*-sovelluksen historiatietojen sisältämät argumentit. Argumentit näytetään peräkkäin, ja erotinmerkkeinä 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(<i>start,end</i>)=Prgm	
For <i>i,start,end</i>	
Disp <i>i</i> , " ",char(<i>i</i>)	
EndFor	
EndPrgm	
	<i>Done</i>
chars(240,243)	
	240 ð
	241 ñ
	242 ò
	243 ó
	<i>Done</i>

DispAtKatalogi > **DispAt int,lauseke1 [,lauseke2 ...] ...**

DispAt

Esimerkki

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.

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

Komentoja **DispAtja 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 lausekkeita. 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	Kuvaus
DisplAt-rivinumeron on oltava lukujen 1 ja 8 välillä	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.

Laus ►DMS

Astekulmatilassa:

Arvo ►DMS

$$\begin{aligned} & \{45.371\} \text{►DMS} && 45^{\circ}22'15.6'' \\ & \{\{45.371,60\}\} \text{►DMS} && \{45^{\circ}22'15.6'',60^{\circ}\} \end{aligned}$$

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 248 .

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

domain()

Luettelo >

domain(Laus1, Muut)⇒lauseke

Antaa vastauksena *Laus1* 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 $f()$ on kielletty funktio.

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

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

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

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

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

dominantTerm(LausI, Muutt [, Piste]) ⇒ lauseke

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

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

Laskee dominanttitermin *LausI*: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 $-\infty$.

$\text{dominantTerm}(\tan(\sin(x)) - \sin(\tan(x)), x)$	$\frac{x^7}{30}$
$\text{dominantTerm}\left(\frac{1 - \cos(x-1)}{(x-1)^3}, x, 1\right)$	$\frac{1}{2 \cdot (x-1)}$
$\text{dominantTerm}\left(x^{-2} \cdot \tan\left(\frac{1}{3}\right), x\right)$	$\frac{1}{x^3}$
$\text{dominantTerm}(\ln(x^x - 1) \cdot x^{-2}, x)$	$\frac{\ln(x \cdot \ln(x))}{x^2}$

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

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

dominantTerm() jakautuu 1. argumentin listoihin ja matriiseihin.

dominantTerm() on hyödyllinen funktio, kun haluat selvittää mahdollisimman yksinkertaisen lausekkeen, joka on asymptoottinen toisen lausekkeen suhteen, esim. *Muutt* → *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 174.

dotP()

dotP(Lista1, Lista2) ⇒ 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) ⇒ 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

Kummannkin on oltava rivivektoreita, tai kummannkin on oltava sarakevektoreita.

E

$e^{\wedge}()$

e^x painike

$e^{\wedge}(LausI) \Rightarrow$ lauseke

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

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

Huomaa: Painikkeen e^x painaminen, jotta näkyviin saadaan $e^{\wedge}$, on eri asia kuin näppäimistön merkin E painaminen.

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

$e^{\wedge}(ListaI) \Rightarrow$ lista

Laskee $e:n$ arvon korotettuna $ListaI:n$ jokaisen elementin potenssiin.

$e^{\wedge}(neliömatrissiI) \Rightarrow$ neliömatrissi

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

$neliömatrissiI:n$ on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

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

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

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

$eff()$

Katalogi > 

$eff(nimelliskorko, CpY) \Rightarrow$ arvo

Talouseläintaloustoiminto, joka muuntaa nimelliskorkokannan $nimelliskorko$ efektiiviseksi vuosikoroksi, kun CpY määritetään korkojaksojen lukumääräksi vuodessa.

$nimelliskoron$ on oltava reaalityyppinen, ja $CpY:n$ on oltava reaalityyppinen > 0 .

$eff(5.75,12)$	5.90398
----------------	---------

Huomaa: Katso myös **nom()**, sivu 132.

eigVc()

eigVc(neliömatriisi) ⇒ *matriisi*

Suorakulmakompleksimuodossa:

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

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

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

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

$$\text{eigVc}(mI) \begin{bmatrix} -0.800906 & 0.767947 & (\\ 0.484029 & 0.573804+0.052258 \cdot i & 0.5738 \cdot \\ 0.352512 & 0.262687+0.096286 \cdot i & 0.2626 \end{bmatrix}$$

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

eigVl()

eigVl(neliömatriisi) ⇒ *lista*

Suorakulmakompleksimuodossa:

Laskee listan reaalisien tai kompleksisen *neliömatriisin* ominaisarvoista.

neliömatriisia tasapainotetaan ensin similaarimuunnoksilla, kunnes rivi- ja sarakenormit ovat mahdollisimman lähellä samaa arvoa. Sen jälkeen *neliömatriisi* sievennetään Hessenbergin ylämatriisimuotoon ja ominaisarvot lasketaan Hessenbergin ylämatriisista.

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

$$\text{eigVl}(mI) \{ -4.40941, 2.20471+0.763006 \cdot i, 2.20471-0. \cdot i \}$$

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

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 $g(x)$ = FuncIf $x \leq -5$ Then

Return 5

ElseIf $x > -5$ and $x < 0$ ThenReturn $-x$ ElseIf $x \geq 0$ and $x \neq 10$ ThenReturn x ElseIf $x = 10$ Then

Return 3

EndIf

EndFunc

Done

EndFor

Katso For, sivu 78.

EndFunc

Katso Func, sivu 82.

EndIf

Katso If, sivu 94.

EndLoop

Katso Loop, sivu 118.

EndPrgm

Katso Prgm, sivu 148.

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*,
{*Var0*, *VarMax*},
ListOfDepVars0, *VarStep* [,
eulerStep]) ⇒ matriisi

Käyttää Eulerin menetelmää
järjestelmän ratkaisuun

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

muuttujalla *depVar*(*Var0*)=*depVar0*
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*).

Differentiaaliyhtälö:

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

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

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

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 deSolve
()- ja seqGEN()-funktioita:

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

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

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

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

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.

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 alkuehtojen 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).

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 reaalityllyvukuksi.

Aseta RGB-ledin sininen väri puolelle intensiteetille.

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

Palauta sininen väri OFF-tilaan.

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

eval()-argumentin on sievennyttävä reaalityllyvukuksi.

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

Ohjelmoi punainen väri voimistumaan

```

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

```

Suorita ohjelma.

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

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

iostr.SendAns
iostr.GetAns
iostr.GetStrAns

Huomio: Katso myös **Get** (sivu 84), **GetStr** (sivu 91), ja **Send** (sivu 171).

exact()

Katalogi >

exact(*Lausl* [, *Toleranssi*]) ⇒ *lauseke*
exact(*Listal* [, *Toleranssi*]) ⇒ *lista*
exact(*Matriisi1* [, *Toleranssi*]) ⇒ *matriisi*

Laskee täsmällisen tilan aritmetiikalla argumentin vastaavan rationaaliluvun, mikäli mahdollista.

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

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

Exit

Katalogi >

Exit

Funktion listaus:

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

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

exp	Katalogi > 	
Laus exp		
Näyttää Laus:n $e:n$ luonnollisen eksponentin arvolla. Tämä on näytön muunnosoperaattori. Sitä voidaan käyttää vain syöterivin lopussa.	$\frac{d}{dx}(e^x + e^{-x})$	$2 \cdot \sinh(x)$
	$2 \cdot \sinh(x)$ exp	$e^x - e^{-x}$
Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @>exp .		

exp()	 painike	
exp(Laus1) ⇒ <i>lauseke</i>	e^1	e
Laskee e:n arvon korotettuna Laus1:n potenssiin.	$e^{1.}$	2.71828
Huomaa: Katso myös e eksponenttimalli, sivu 2.	e^{3^2}	e^9
Voit syöttää kompleksiluvun $rei\theta$ polaarisisä muodossa. Käytä tätä tuotoa kuitenkin vain radiaanikulmatilassa; aste- tai graadikulmatilassa se aiheuttaa määrittelyjoukkovirheen (Domain).		
exp(Lista1) ⇒ <i>lista</i>	$e^{\{1,1,0.5\}}$	$\{e,2.71828,1.64872\}$
Laskee e:n arvon korotettuna Listal:n jokaisen elementin potenssiin.		

exp()

e^x painike

exp(neliömatriisiI)⇒*neliömatriisi*

Laskee *neliömatriisiI*:n matriisieksponentin. 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.

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

exp►list()

Katalogi >

exp►list(Laus,Muutt)⇒*lista*

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 (...)**.

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

expand()

Katalogi >

expand(LausI [, Muutt])⇒*lauseke*

expand(ListaI [,Muutt])⇒*lista*

expand(MatriisiI [,Muutt])⇒*matriisi*

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

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

Funktion **expand()** tehtävä on muuntaa *Laus1* yksinkertaisten termien summaksi ja/tai erotukseksi. Funktion **factor()** tehtävä sen sijaan on muuntaa *Laus1* yksinkertaisten tekijöiden tuloksi ja/tai osamääräksi.

expand(Laus1, Muutt) laskee *Laus1*: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.

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(Laus1, [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(Laus1, [Muutt]) jakaa myös itseisarvot, **sign()**, ja eksponentit muuttujasta *Muutt* riippumatta.

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

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

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

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

expr()

expr(Merkkijono) ⇒ lauseke

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

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

ExpReg

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

Laskee eksponentiaalisen regressiоны $y = a \cdot (b)^x$ listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 272.

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 sovittukseen liittyvät jäännökset
stat.XReg	Muokatun <i>X Lista</i> :n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä luokat</i> 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 luokat</i> rajoitusten mukaisesti
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava frekvenssilista

F

factor()

Katalogi > 

factor(LausI[, Muutt]) ⇒ lauseke

factor(ListaI[, Muutt]) ⇒ lista

factor(MatriisiI[, Muutt]) ⇒ matriisi

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

LausI :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(LausI, Muutt) laskee *LausI*:n jaettuna tekijöihin muuttujan *Muutt* suhteen.

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

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

Laus 1 :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.

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.

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

$$\frac{\text{factor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3, x)}{(x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x^2$$

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.

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

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.

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

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

Fill

Fill *Laus*, *matriisiMuutt* $\Rightarrow$ *matriisi*

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

*matriisiMuutt*ujan on oltava valmiiksi olemassa.

Fill *Laus*, *listaMuutt* $\Rightarrow$ *lista*

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

*listaMuutt*ujan on oltava valmiiksi olemassa.

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

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

FiveNumSummary

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

Antaa lyhennetyn version 1 muuttujan tilastoista listalle *X*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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.

Sisällyttä 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 272.

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(LausI)⇒kokonaisluku

$$\text{floor}(-2.14) \qquad -3.$$

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

Argumentti voi olla reaali- tai kompleksiluku.

floor(ListaI)⇒lista

floor(MatriisiI)⇒matriisi

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

Määrittää listan tai matriisin jokaisen elementin alarajasta.

$$\text{floor}\left(\begin{bmatrix} 1.2 & 3.4 \\ 2.5 & 4.8 \end{bmatrix}\right) \qquad \begin{bmatrix} 1. & 3. \\ 2. & 4. \end{bmatrix}$$

Huomaa: Katso myös **ceiling()** ja **int()**.

fMax()

fMax(Laus, Muutt)⇒Boolean lauseke

fMax(Laus, Muutt, alaraja)

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

fMax(Laus, Muutt, alaraja, ylärajaja)

$$\text{fMax}\left(.5 \cdot x^3 - x - 2, x\right) \qquad x=\infty$$

fMax(Laus, Muutt) | alaraja≤Muutt

$\leq$ yläraja

Laskee Boolean lausekkeen, joka määrittää mahdollisia arvoja muuttujalle *Muutt*, joilla saadaan suurin lausekkeen *Laus* arvo tai jotka määrittävät sen pienimmän ylärajan.

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

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

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(Laus, Muutt) $\Rightarrow$ Boolean lauseke

fMin(Laus, Muutt, alaraja)

fMin(Laus, Muutt, alaraja, yläraja)

fMin(Laus, Muutt) | alaraja $\leq$ Muutt $\leq$ yläraja

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.

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

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

Katalogi >

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

Suorittaa *Lohkon* sisältämät lausekkeet iteratiivisesti jokaiselle muuttujan *Muutt* arvolle, *Matalasta Korkeaan* kohdassa *Askel* määritetyin 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 <i>g()</i> =Func	Done
Local <i>tempsum,step,i</i>	
0 → <i>tempsum</i>	
1 → <i>step</i>	
For <i>i</i> ,1,100, <i>step</i>	
<i>tempsum</i> + <i>i</i> → <i>tempsum</i>	
EndFor	
EndFunc	
<i>g()</i>	5050

format()

Katalogi >

format(*Laus*[,
muotoMerkkijono])⇒*merkkijono*

Määrittää lausekkeen *Laus* merkkijonona muotoilumallin perusteella.

Laus on voitava sieventää luvuksi.

format(1.234567,"f3")	"1.235"
format(1.234567,"s2")	"1.23E0"
format(1.234567,"e3")	"1.235E0"
format(1.234567,"g3")	"1.235"
format(1234.567,"g3")	"1,234.567"
format(1.234567,"g3,r:")	"1:235"

muoto Merkkijono 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 kerrokselliseksi, 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.

fPart(*Lausl*) $\Rightarrow$ *lauseke*

fPart(*Listal*) $\Rightarrow$ *lista*

fPart(*Matriisil*) $\Rightarrow$ *matriisi*

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

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

Laskee argumentin murtolukuosan.

Kun kyseessä on lista tai matriisi, laskee elementtien murtolukuosat.

Argumentti voi olla reaali- tai kompleksiluku.

$Fpdf(XArvo, dfOsoitt, dfNimitt) \Rightarrow luku$, jos $XArvo$ on luku, *lista*, jos $XArvo$ on lista

Laskee F-jakauman todennäköisyyden $XArvon$ kohdalle määritetyille $dfOsoittajalle$ (vapausasteet) ja $dfNimittäjälle$.

freqTable►list()

freqTable►list

$(Lista1, frekvKokonaislukuLista) \Rightarrow lista$

Laskee listan, joka sisältää *Lista1*:n elementit lavennettuina *frekvKokonaislukuListan* määrittämien frekvenssien mukaisesti. Tätä funktiota voidaan käyttää laadittaessa frekvenssitaulukkoa Data & Tilastot -sovelluksessa.

Lista1 voi olla mikä tahansa kelvollinen lista.

frekvKokonaislukuListan on oltava samankokoinen kuin *Lista1* ja sen tulee sisältää ainoastaan ei-negatiivisia kokonaislukuelementtejä. Jokainen elementti määrittää kuinka monta kertaa *Lista1*-elementti toistetaan tuloslistassa. Nolla-arvo sulkee pois vastaavan *Lista1*-elementin.

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

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 272.

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

frequency()

$frequency(Lista1, lokerotLista) \Rightarrow lista$

Luo listan, joka sisältää *Lista1*:n elementtien lukumäärät. Lukumäärät perustuvat alueisiin (lokeroihin), jotka määritetään kohtaan *lokerotLista*.

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

Jos *lokerotLista* on $\{b(1), b(2), \dots, b(n)\}$, määritetyt alueet ovat $\{? \leq b(1), b(1) < ? \leq b(2), \dots, b(n-1) < ? \leq b(n), b(n) > ?\}$. Tuloksena oleva lista on yhden elementin pitempi kuin *lokerotLista*.

Jokainen vastauksen elementti vastaa niiden *Lista1*:n elementtien lukumäärää, jotka ovat kyseisen lokeron alueella. Funktion **countlf()** termeillä ilmaistuna vastaus on $\{\text{countlf}(\text{list}, ? \leq b(1)), \text{countlf}(\text{list}, b(1) < ? \leq b(2)), \dots, \text{countlf}(\text{list}, b(n-1) < ? \leq b(n)), \text{countlf}(\text{list}, b(n) > ?)\}$.

Niitä *Lista1*:n elementtejä, joita ei voi "lokeroida", ei huomioida. Tyhjiä elementtejä ei myöskään huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 272.

Listat & Taulukot -sovelluksessa voit käyttää solualueita kummankin argumentin tilalla.

Huomaa: Katso myös **countlf()**, sivu 37.

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 *Lista1, Lista2[, Frekv1[, Frekv2[, Hypot]]]*

FTest_2Samp *Lista1, Lista2[, 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 yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Kun $H_a: \sigma_1 > \sigma_2$, aseta *Hypot* > 0

Kun $H_a: \sigma_1 \neq \sigma_2$ (oletus), aseta *Hypot* = 0

Kun $H_a: \sigma_1 < \sigma_2$, aseta *Hypot* < 0

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.F	Laskettu F -tilasto datasekvenssille
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:n</i> ja <i>Lista 2:n</i> sisältämille datasekvensseille
stat.x1_bar stat.x2_bar	Otoksen keskiarvot <i>Lista 1:n</i> ja <i>Lista 2:n</i> sisältämille datasekvensseille
stat.n1, stat.n2	Otosten koko

Func

Func

Lohko

EndFunc

Malli käyttäjän määrittämän funktion luomista varten.

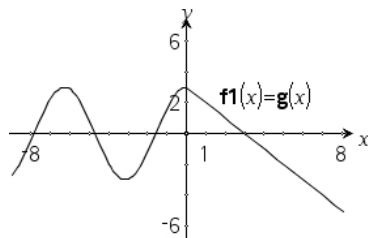
Lohko voi olla yksi lauseke tai sarja lausekkeita, jotka on erotettu toisistaan kaksoispisteellä (:), tai sarja eri riveillä olevia lausekkeita. Funktio voi käyttää **Return**-ohjetta tietyn vastauksen laskemiseen.

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

Määritä paloittain määritelty funktio:

Define $g(x)=$ Func	Done
If $x<0$ Then	
Return $3\cdot\cos(x)$	
Else	
Return $3-x$	
EndIf	
EndFunc	

Funktion $g(x)$ kuvaajan piirtämisen tulos



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(*List1*, *List2*) $\Rightarrow$ *lista* $\text{gcd}(\{12,14,16\},\{9,7,5\})$ { 3,7,1 }

Laskee *List1*:n ja *List2*:n toisiaan vastaavien elementtien suurimmat yhteiset jakajat.

gcd(*Matriisi1*, *Matriisi2*) $\Rightarrow$ *matriisi* $\text{gcd}\left(\begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}, \begin{pmatrix} 4 & 8 \\ 12 & 16 \end{pmatrix}\right)$ $\begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}$

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 $P(X \leq \text{yläraja})$, aseta *alaraja* = 1.

geomPdf()Katalogi > 

geomPdf(*p*,*XArvo*) $\Rightarrow$ *luku*, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

Laskee diskreetin jakauman todennäköisyyden $XArvo:n$, eli ensimmäisen onnistuneen kokeen järjestysnumeron kohdalla, määritetyllä onnistumistodennäköisyyllä p .

Get

Laittevalikko

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.

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:

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

Määrittele $func(arg1, \dots, argn) =$
vastaanotettu merkkijono

Sen jälkeen ohjelma voi käyttää
määritettyä funktiota $func()$.

Huomio: Kommentoita **Get** voi käyttää
käyttäjän määrittelemän ohjelman
sisällä, mutta ei funktion sisällä.

Huomio: Katso myös **GetStr**, sivu 91 ja
Send sivu 171.

getDenom()

Katalogi >

getDenom($Laus1$) $\Rightarrow$ lauseke

Muuttaa argumentin lausekkeeksi, jolla
on sievennetty yhteinen nimittäjä, ja
laskee sen jälkeen lausekkeen nimittäjän.

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

getKey()

Katalogi >

getKey([0|1]) $\Rightarrow$ returnString

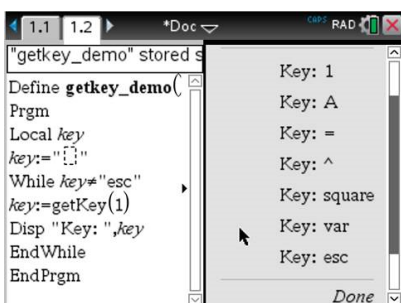
Kuvaus: **getKey()** – mahdollistaa TI-
Basic -ohjelmalle näppäimistön –
kannettava laite, pöytätietokone ja
emulaattori pöytätietokoneella.

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.

getKey()

Esimerkki:



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 – Keskipainallus	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"
=	=	"="
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"

Kannettava laite/Emulaattorinäppäin	Pöytätietokone	Palauta arvo
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
,	,	" , "
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	Erityiset näytön kontrollinäppäimet	Kirjainmerkkejä ei ole palautettu
Ei pääsyä	Muita näytön näppäimiä, jotka eivät ole käytettävissä	Sama kirjainmerkki, jonka saat Notesista (ei

Kannettava laite/Emulaattorinäppäin	Pöytätietokone	Palauta arvo
	laskimessa, kun getKey() odottaa näppäimen painallusta. ({, },, ;, ...)	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()

getLockInfo(*Muutt*)⇒*arvo*

Määrittää muuttujan *Muutt* nykyisen lukittu/lukitsematon-tilan.

arvo =0: *Muutt* on lukitsematon tai sitä ei ole olemassa.

arvo =1: *Muutt* on lukittu eikä sitä voi muuttaa tai poistaa.

Katso **Lock**, sivu 114, ja **unLock**, sivu 214.

<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

getMode()

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 176.

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

getNum()	Katalogi > 	
getNum (<i>Laus1</i>) $\Rightarrow$ <i>lauseke</i>		
Muuttaa argumentin lausekkeeksi, jolla on sievennetty yhteinen nimittäjä, ja laskee sen jälkeen lausekkeen osoittajan.		
	$\text{getNum}\left(\frac{x+2}{y-3}\right)$	$x+2$
	$\text{getNum}\left(\frac{2}{7}\right)$	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 84 ja Send sivu 171.		

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".		
	$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$
	$\text{getType}(temp)$	"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 käytettävyyys 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*) sievennyy 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>volcy12</td><td>"NONE"</td><td>"LibPub"</td><td>0</td></tr></table>	volcy12	"NONE"	"LibPub"	0								
volcy12	"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 tunnukseseen *tunnusnimi*.

tunnusnimi on määritettävä samassa funktiossa käyttäen **Lbl**-ohjetta.

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

Define $g()$ =Func	Done
Local $temp,i$	
$0 \rightarrow temp$	
$1 \rightarrow i$	
Lbl top	
$temp+i \rightarrow temp$	
If $i < 10$ Then	
$i+1 \rightarrow i$	
Goto top	
EndIf	
Return $temp$	
EndFunc	
$g()$	55

►Grad

Laus1 ► **Grad** ⇒ lauseke

Muuttaa *Laus1*:n graadikulmaan.

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

Astekulmatilassa:

$(1.5) \blacktriangleright \text{Grad}$ $(1.66667)^{\circ}$

Radiaanikulmatilassa:

$(1.5) \blacktriangleright \text{Grad}$ $(95.493)^{\circ}$

/

identity()

identity(*kokonaisluku*) ⇒ *matriisi*

Laskee identiteettimatriisin, jonka koko on *kokonaisluku*.

Kokonaisluvun on oltava positiivinen kokonaisluku.

identity(4)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
-------------	--

Jos *BooleanLaus*
Ilmaisut

Jos *BooleanLaus Niin*
Lohko

EndIf

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

Jos BooleanLaus on tosi, suorittaa *Lohko1*:n ja sen jälkeen jättää väliin *Lohko2*:n.

Jos BooleanLaus on epätosi, ohittaa *Lohko1*:n, mutta suorittaa *Lohko2*:n.

Lohko1 ja *Lohko2* voivat olla yksi lauseke.

Define $g(x)$ =Func	Done
If $x < 0$ Then	
Return x^2	
EndIf	
EndFunc	
$g(-2)$	4

Define $g(x)$ =Func	Done
If $x < 0$ Then	
Return $-x$	
Else	
Return x	
EndIf	
EndFunc	
$g(12)$	12
$g(-12)$	12

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 määritelty. Käytetään merkintää EiMäär

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()

imag(Expr1) ⇒ lauseke

Laskee argumentin imaginaarisen osan.

Huomaa: Kaikkia määrittelemättömiä muuttujia käsitellään reaali muuttujina. Katso myös *real()*, page 157

imag(Lista1) ⇒ lista

Laskee listan alkutekijöiden imaginaarisista osista.

imag(Matriisi1) ⇒ matriisi

Laskee matriisin alkutekijöiden imaginaarisista osista.

$$\begin{array}{ll} \text{imag}(1+2 \cdot i) & 2 \\ \text{imag}(z) & 0 \\ \text{imag}(x+i \cdot y) & y \end{array}$$

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

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

impDif()

impDif(Yhtälö, Muutt, riippuvaMuutt [,Aste]) ⇒ lauseke

jossa luokan *Aste* oletusarvo on 1

Laskee implisiittisen derivaatan yhtälöille, joissa yksi muuttuja määritellään implisiittisesti toisen suhteen

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

Epäsuora operaattori

inString()

Luettelo > 

inString(*srcMerkkijono*, *alaMerkkijono* [, *Alku*]) ⇒ kokonaisluku

inString("Hello there", "the")	7
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()

Luettelo > 

int(*Laus*) ⇒ kokonaisluku

int(*Lista1*) ⇒ lista

int(*Matriisi1*) ⇒ matriisi

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

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()

Luettelo > 

intDiv(*Luku1*, *Luku2*) ⇒ kokonaisluku

intDiv(*Lista1*, *Lista2*) ⇒ lista

intDiv(*Matriisi1*, *Matriisi2*) ⇒ matriisi

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

Laskee lausekkeen *Luku1* ÷ *Luku2*) etumerkillisen kokonaislukuosan.

Laskee sarjoille ja matriiseille lausekkeen (argumentti 1 ÷ 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:

$$y' = -3 \cdot y + 6 \cdot t + 5 \text{ and } y(0) = 5$$

$rk := rk23(-3 \cdot y + 6 \cdot t + 5, y, \{0, 10\}, 5, 1)$					
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()-funktiota 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., 8.5, 9., 9.5, 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 annetulle käyrän alapuoliselle alalle Ala .

invF()

Luettelo > 

invF(Ala , $dfOsoitt$, $dfNimitt$)

vakioF(Area,dfOsoitt,dfNimitt)

Laskee käänteisen kumulatiivisen F-jakaumafunktion, jolle on määritelty $dfOsoitt$ ja $dfNimitt$, annetulle käyrän alapuoliselle alueelle *Ala*.

invBinom()**invBinom**

(CumulativeProb, NumTrials, Prob, OutputForm) ⇒ *asteikkumuoto* 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 asteikkumuodossa (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()

invBinomN*(CumulativeProb, Prob, NumSuccess, OutputForm)* ⇒ *asteikkumuoto* 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 asteikkumuodossa (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**(*List1*) $\Rightarrow$ lista**iPart**(*Matriisi*) $\Rightarrow$ matriisi

iPart (-1.234)	-1.
iPart ($\left\{\frac{3}{2}, -2.3, 7.003\right\}$)	$\{1, 2., 7.\}$

Laskee argumentin kokonaisosan.

Laskee sarjoille ja matriiseille kunkin elementin kokonaisosan.

Argumentti voi olla reaali- tai kompleksiluku.

irr()Luettelo > **irr**(*CF0*, *CFList1* [, *CFFrekv*]) $\Rightarrow$ arvo

Talouseläintaloustoiminto, joka laskee investoinnin sisäisen korkokannan.

CF0 on kassavirta alussa aikana 0; arvon on oltava kokonaisluku.

CFList1 on lista kassavirtamääristä alun kassavirran *CF0* jälkeen.

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

CFFreqv on valinnainen lista, jossa kukin elementti määrää esiintymisfrekvenssin ryhmitetylle (peräkkäiselle) kassavirtamäärälle, joka on *CFFreqv*:n vastaava alkutekijä. Oletusarvo on 1; jos syötät arvoja, niiden on oltava positiivisia kokonaislukuja ja < 10 000.

Huomaa: Katso myös *mirr()*, sivu 124.

isPrime()

isPrime(Luku) ⇒ *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 kommentoa **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 määrittysten 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) ⇒ *Boolean vakiolauseke*

isVoid(Laus) ⇒ *Boolean vakiolauseke*

isVoid(Lista) ⇒ *lista Boolean vakiolausekkeista*

Laskee totuusarvon tosi tai epätosi ilmaisten, onko argumentti tyhjä datatyyppi.

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

Lisätietoja tyhjästä elementeistä, katso sivu 272.

L

Lbl

Katalogi > **Lbl** *tunnusnimi*

Määrittää funktion sisällä tunnuksen, jonka nimi on *tunnusnimi*.

Ohjeella **Siirry** *tunnusnimi* voit siirtää ohjauksen kyseistä tunnusta välittömästi seuraavaan ohjaukseen.

tunnusnimellä on samat nimeämissäännöt kuin muuttujan nimellä.

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

```
Define g()=Func
    Local temp,i
    0→temp
    1→i
    Lbl top
    temp+i→temp
    If i<10 Then
        i+1→i
        Goto top
    EndIf
    Return temp
EndFunc
```

Done

$g()$	55
-------	----

lcm()

Katalogi >

lcm(*Luku1*, *Luku2*)⇒*lauseke*

lcm(*Lista1*, *Lista2*)⇒*lista*

lcm(*Matriisi1*, *Matriisi2*)⇒*matriisi*

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.

$\text{lcm}(6,9)$	18
$\text{lcm}\left(\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}\right)$	$\left\{\frac{2}{3}, 14, 80\right\}$

left()

Katalogi >

left(*lähdemerkkijono*[, *Num*])⇒*merkkijono*

$\text{left}(\text{"Hello"}, 2)$	"He"
----------------------------------	------

Määrittää vasemmanpuoleisimmat *Num*-merkit, jotka sisältyvät merkkijonoon *lähdemerkkijono*.

Jos jätät pois komennon *Num*, määrittää kaiken merkkijonosta *lähdemerkkijono*.

left(*Listal*, *Num*) $\Rightarrow$ *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*) $\Rightarrow$ *lauseke*

$\text{left}(x<3)$ x

Laskee yhtälön tai epäyhtälön vasemman puolen.

libShortcut()

libShortcut(*KirjNimiMerkkijono*, *PikavalNimiMerkkijono* [, *KirjYksitLippu*]) $\Rightarrow$ *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 kommentoa *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 31.
Muuttujaryhmän poistaminen, katso **DelVar** sivulla sivu 52.

Tässä esimerkissä edellytetään asianmukaisesti tallennettua ja päivitettyä kirjastoasiakirjaa nimeltä **linalg2**, joka sisältää objektit *clearmat*, *gauss1* ja *gauss2*.

$\text{getVarInfo}(\text{"linalg2"})$

<i>clearmat</i>	"FUNC"	"LibPub "
<i>gauss1</i>	"PRGM"	"LibPriv "
<i>gauss2</i>	"FUNC"	"LibPub "

$\text{libShortcut}(\text{"linalg2"}, \text{"la"})$

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

$\text{libShortcut}(\text{"linalg2"}, \text{"la"}, 1)$

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

limit(*Laus1*, *Muutt*, *Piste*[,*Suunta*])⇒*lauseke***raja-arvo**(*Listal*, *Muutt*, *Piste* [,*Suunta*])⇒*lista***raja-arvo**(*Matriisil*, *Muutt*, *Piste* [,*Suunta*])⇒*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 *Laus1* sisältää muita määrittämättömiä muuttujia kuin *Muutt*, joudut mahdollisesti rajoittamaan niitä saadaksesi suppeamman tuloksen.

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

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

$$\lim_{x \rightarrow 0} \left(\frac{\sin(x)}{x} \right) \quad 1$$

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

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

$$\lim_{x \rightarrow \infty} (a^x) \quad \text{undef}$$

$$\lim_{x \rightarrow \infty} (a^x) | a > 1 \quad \infty$$

$$\lim_{x \rightarrow \infty} (a^x) | a > 0 \text{ and } a < 1 \quad 0$$

Raja-arvot voivat olla hyvin herkkiä pyöristysvirheille. Mikäli mahdollista, vältä **Auto or Approximate (Automaattinen tai Likimääräinen)** -tilan Approximate (Likimääräinen) -asetusta ja likiarvoista luvut laskiessasi 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.

LinRegBx

LinRegBx $X, Y, [Frequ]$, $[Luokka, Sisällytä]$

Laskee lineaarisen regression $y = a + b \cdot x$ listoista X ja Y frekvenssillä $Frequ$. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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

X ja Y ovat riippumattomien ja riippuvien muuttujien listoja.

Frequ on valinnainen frekvenssiarvojen lista. Jokainen *Frequ*: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 elementtisivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a+b \cdot x$
stat.a, stat.b	Regressiokertoimet
stat.r ²	Määrittyskerroin
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 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

LinRegMx

Katalogi > 

LinRegMx *X*,*Y*, [*Frekv*], [*Luokka*, *Sisällytä*]]

Laskee lineaarisen regression $y = m \cdot x + b$ listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $m \cdot x + b$
stat.m, stat.b	Regressiokertoimet
stat.r ²	Määrittyskerroin
stat.r	Korrelaatiokerroin
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

LinRegtIntervals

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 191.)

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 tyhjiä elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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
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

Tulosmuuttuja	Kuvaus
stat.SEPred	Ennusteen keskivirhe
stat.ŷ	$a + b \cdot X_{\text{Arvo}}$

LinRegtTest

Katalogi > 

LinRegtTest $X, Y[, \text{Frekv}[, \text{Hypot}]]$

Laskee lineaarisen regression X - ja Y -listoista ja suorittaa t -testin kulmakertoimen β ja korrelaatiokerroimen 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 $H_a: \beta \neq 0$ ja $\rho \neq 0$ (oletus), aseta *Hypot*=0
 Kun $H_a: \beta < 0$ ja $\rho < 0$, aseta *Hypot*<0
 Kun $H_a: \beta > 0$ ja $\rho > 0$, aseta *Hypot*>0

Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu sivu 191.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjä elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a + b \cdot x$
stat.t	t -tilasto merkitsevyydestille
stat.PVal	Alin merkitsevyytaso, jolla nollahypoteesi voidaan hylätä

Tulosmuuttuja	Kuvaus
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()

Katalogi > 

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 ..., {Muutt1, Muutt2, ...})⇒lista

linSolve({Lineaariyht1, Lineaariyht2, ...}, {Muutt1, Muutt2, ...})⇒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.

$\text{linSolve}\left(\left\{\begin{array}{l} 2 \cdot x + 4 \cdot y = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{array}\right\}, \{x, y\}\right)$	$\left\{\frac{37}{26}, \frac{1}{26}\right\}$
$\text{linSolve}\left(\left\{\begin{array}{l} 2 \cdot x = 3 \\ 5 \cdot x - 3 \cdot y = 7 \end{array}\right\}, \{x, y\}\right)$	$\left\{\frac{3}{2}, \frac{1}{6}\right\}$
$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple} + 4 \cdot \text{pear} = 23 \\ 5 \cdot \text{apple} - \text{pear} = 17 \end{array}\right\}, \{\text{apple}, \text{pear}\}\right)$	$\left\{\frac{13}{3}, \frac{14}{3}\right\}$
$\text{linSolve}\left(\left\{\begin{array}{l} \text{apple} \cdot 4 + \frac{\text{pear}}{3} = 14 \\ -\text{apple} + \text{pear} = 6 \end{array}\right\}, \{\text{apple}, \text{pear}\}\right)$	$\left\{\frac{36}{13}, \frac{114}{13}\right\}$

$\Delta\text{List}()$ **Katalogi >**  **$\Delta\text{List}(\text{Lista}) \Rightarrow \text{lista}$**

$\Delta\text{List}(\{20,30,45,70\})$	$\{10,15,25\}$
--------------------------------------	----------------

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **$\Delta\text{List}(\dots)$** .

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*.

 $\text{list} \blacktriangleright \text{mat}()$ **Katalogi >**  **$\text{list} \blacktriangleright \text{mat}(\text{Lista } [, \text{elementtiäRivillä}]) \Rightarrow \text{matriisi}$**

$\text{list} \blacktriangleright \text{mat}(\{1,2,3\})$	$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
$\text{list} \blacktriangleright \text{mat}(\{1,2,3,4,5\},2)$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$

Laskee matriisin, joka on täytetty rivi riviltä *Listan* elementeillä.

elementtiäRivillä, mikäli sisällytetään, määrittää elementtien määrän rivillä. Oletusarvo on *Listan* elementtien määrä (yksi rivi).

Jos *Listal* ei täytä tulomatriisia, siihen lisätään nollia.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla **$\text{list} @ \blacktriangleright \text{mat}(\dots)$** .

 $\blacktriangleright \ln$ **Katalogi >**  **$\text{Laus} \blacktriangleright \ln \Rightarrow \text{lauseke}$**

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

Aiheuttaa sen, että lausekkeen *Laus* syöte muunnetaan luonnollisia logaritmeja (*ln*) sisältäväksi lausekkeeksi.

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla **$@ \blacktriangleright \ln$** .

ln(Lausl)⇒*lauseke***ln(Lista1)**⇒*lista*

ln(2.)

0.693147

Määrittää argumentin luonnollisen logaritmin.

Jos kompleksilukumuodon tila on Reaali:

Jos kyseessä on lista, määrittää elementtien luonnolliset logaritmit.

$$\ln(\{-3, 1.2, 5\})$$

"Error: Non-real calculation"

Jos kompleksilukumuodon tila on Suorakulma:

$$\ln(\{-3, 1.2, 5\}) \quad \{\ln(3) + \pi \cdot i, 0.182322, \ln(5)\}$$
ln(neliömatriisi1)⇒*neliömatriisi*

Määrittää *neliömatriisi1*:n matriisin luonnollisen logaritmin. Tämä ei ole sama kuin kunkin elementin luonnollisen logaritmin laskeminen.

Laskentamenetelmä on kuvattu kohdassa **cos()**.

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

neliömatriisi1: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 logaritmisien regression $y = a + b \cdot \ln(x)$ listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 272.

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
stat.ResidTrans	Muunnettujen tietojen lineaariseen sovitukseen liittyvät 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

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	
	<i>Done</i>
<i>rollcount</i> ()	16
<i>rollcount</i> ()	3

LockKatalogi > **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*.

Huomaa: Lukitse-komento (**Lock**) tyhjentää toimintojen Kumoa/Tee uudelleen historian, kun sitä käytetään lukitsemattomiin muuttujiin.

Katso **unLock**, sivu 214, ja **getLockInfo()**, sivu 89.

<i>a</i> :=65	65
Lock <i>a</i>	<i>Done</i>
getLockInfo(<i>a</i>)	1
<i>a</i> :=75	"Error: Variable is locked."
DelVar <i>a</i>	"Error: Variable is locked."
Unlock <i>a</i>	<i>Done</i>
<i>a</i> :=75	75
DelVar <i>a</i>	<i>Done</i>

log()

ctrl **10^x** **painikkeet**

log(Laus1[,Laus2])⇒lauseke**log(Lista1[,Laus2])**⇒lista

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, kantalukuna käytetään lukua 10.

log(neliömatriisi1[,Laus])⇒neliömatriisi

Laskee matriisin kantaluku--Laus:n logaritmin *neliömatriisi1*:lle. Tämä ei ole sama kuin kunkin elementin kantaluku--Laus-logaritmin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisi1:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

Jos kantalukuargumentti jätetään pois, kantalukuna käytetään lukua 10.

$$\begin{array}{r} \log_{10} (2.) \\ \hline \log_4 (2.) \\ \hline \log_3 (10) - \log_3 (5) \end{array} \qquad \begin{array}{r} 0.30103 \\ \\ \log_3 (2) \end{array}$$

Jos kompleksilukumuodon tila on Reaali:

$$\log_{10} (\{-3, 1.2, 5\}) \quad \text{Error: Non-real result}$$

Jos kompleksilukumuodon tila on Suorakulma:

$$\log_{10} (\{-3, 1.2, 5\}) \\ \left\{ \log_{10} (3) + 1.36438 \cdot i, 0.079181, \log_{10} (5) \right\}$$

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

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

logbase

Katalogi >

Laus ▶logbase(Laus1)⇒lauseke

Aiheuttaa sen, että syötteenä oleva lauseke sievennetään lausekkeeksi, joka käyttää kantaluku--Laus1:ä.

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla **@>logbase (...)**.

$$\log_3 (10) - \log_5 (5) \text{ ▶logbase}(5) \qquad \frac{\log_5 \left(\frac{10}{3} \right)}{\log_5 (3)}$$

Logistic X , Y , [$Frekv$] [, $Luokka$, $Sisällytä$]]

Laskee logistisen regressioyhtälön $c/(1+a \cdot e^{-bx})$ listaista X ja Y frekvenssillä $Frekv$. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 272.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $c/(1+a \cdot e^{-bx})$
stat.a, stat.b, stat.c	Regressiokertoimet
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

LogisticD X, Y [, [*Iteraatiot*] , [*Frekv*] [, *Luokka*, *Sisällytä*]]

Laskee logistisen regression $y = (c/(1+a \cdot e^{-bx})+d)$ listoista X ja Y frekvenssillä *Frekv* käyttäen tiettyä *Iteraatioiden* määrää. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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

Tulosmuuttuja	Kuvaus
stat.XReg	Muokatun <i>X Listan</i> sisältämä datapisteiden lista, jota käytetään todellisesti regressiossa komentojen <i>Frekv</i> , <i>Luokka Lista</i> ja <i>Sisällyttä Luokat rajoitusten mukaisesti</i>
stat.YReg	Muokatun <i>Y Listan</i> sisältämä datapisteiden lista, jota käytetään todellisesti regressiossa komentojen <i>Frekv</i> , <i>Luokka Lista</i> ja <i>Sisällyttä Luokat rajoitusten mukaisesti</i>
stat.FreqReg	Komentoja <i>stat.XReg</i> ja <i>stat.YReg</i> vastaava <i>frekvenssilista</i>

Loop (Silmukka)
Katalogi >

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 <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	
	<i>Done</i>
<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 **ctrl** **enter** 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).

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

LU *m1*, *lower*, *upper*, *perm* Done

$$\begin{array}{l} \text{lower} \\ \text{upper} \end{array} \quad \begin{array}{l} \begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} \\ \begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$\text{perm} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

LU *m1*, *lower*, *upper*, *perm* Done

$$\begin{array}{l} \text{lower} \\ \text{upper} \\ \text{perm} \end{array} \quad \begin{array}{l} \begin{bmatrix} 1 & 0 \\ m & 1 \\ o & \end{bmatrix} \\ \begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array}$$

M

mat►list()

mat►list(*Matriisi*) $\Rightarrow$ *lista*

Luo listan, joka on täytetty *Matriisin* elementeillä. Elementit kopioidaan *Matriisista* rivi riviltä.

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

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$\text{mat} \blacktriangleright \text{list}(m1) \quad \{1, 2, 3, 4, 5, 6\}$$

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla `mat@>list(...)`.

max()

Katalogi >

max(Laus1, Laus2) ⇒ lauseke

max(Lista1, Lista2) ⇒ lista

max(Matriisi1, Matriisi2) ⇒ matriisi

Laskee kahden argumentin maksimiaron. Jos argumentteina on kaksi listaa tai matriisia, laskee listan tai matriisin, joka sisältää vastaavien elementtien kunkin parin maksimiaron.

max(Lista) ⇒ lauseke

Laskee *lista*:n maksimielementin.

max(Matriisi1) ⇒ matriisi

Luo rivivektorin, joka sisältää *Matriisi1*:n jokaisen sarakkeen maksimielementin.

Tyhjiä elementtejä ei huomioida.

Lisätietoja tyhjiä elementeistä, katso sivu 272.

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

$\max(2.3, 1.4)$	2.3
$\max(\{1, 2\}, \{-4, 3\})$	$\{1, 3\}$

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

$\max\left(\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 & 7 \end{bmatrix}$
---	---

mean()

Katalogi >

mean(Lista[, frekvLista]) ⇒ lauseke

Laskee *Listan* sisältämien elementtien keskiarvon.

Jokainen *frekvListan* elementti näyttää *Listan* vastaavien elementtien peräkkäisten esiintymien lukumäärän.

**mean(Matriisi1[,
frekvMatriisi]) ⇒ matriisi**

Luo rivivektorin kaikkien *Matriisi1*:n sarakkeiden keskiarvoista.

$\text{mean}(\{0.2, 0.1, -0.3, 0.4\})$	0.26
$\text{mean}(\{1, 2, 3\}, \{3, 2, 1\})$	$\frac{5}{3}$

Suorakulmavektorimuodossa:

mean()Katalogi > 

Jokainen *frekvMatriisin* elementti näyttää *Matriisi1*:n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjiä elementeistä, katso sivu 272.

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	$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}, \begin{pmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{pmatrix}$	$\begin{bmatrix} \frac{47}{15} & \frac{11}{3} \end{bmatrix}$

median()Katalogi > 

median(Lista[, frekvLista])⇒*lauseke*

Laskee *Listan* elementtien mediaanin.

Jokainen *frekvListan* elementti näyttää *Listan* vastaavien elementtien peräkkäisten esiintymien lukumäärän.

median(Matriisi1[, frekvMatriisi])⇒*matriisi*

Luo rivivektorin, joka sisältää *Matriisi1*:n sarakkeiden mediaanit.

Jokainen *frekvMatriisin* elementti näyttää *Matriisi1*: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 272.

median	$\{0.2, 0.1, -0.3, 0.4\}$	0.2
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, Sisällytä]

Laskee mediaani-mediaani-suorany = $(m \cdot x + b)$ listoista X ja Y frekvenssillä *Frekv*. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 elementtisivulla sivu 272.

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	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

mid()Katalogi > 

mid(lähdemerkkijono, Alku[, Count])⇒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])⇒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])⇒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)⇒lauseke

min(Lista1, Lista2)⇒lista

min(Matriisi1, Matriisi2)⇒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)⇒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(MatriisiI)⇒*matriisi*

Luo rivivektorin, joka sisältää *MatriisiI*:n jokaisen sarakkeen minimielementin.

$$\min \begin{pmatrix} 1 & -3 & 7 \\ -4 & 0 & 0.3 \end{pmatrix} \quad \begin{bmatrix} -4 & -3 & 0.3 \end{bmatrix}$$

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

mirr

(*tal.arvo,uud.invest.arvo,CF0,CFLista*
[,*CFFrekv*])

Talouselaskentatoiminto, joka laskee investoinnin modifioidun sisäisen korkokannan.

$$\begin{aligned} list1 &:= \{6000, -8000, 2000, -3000\} \\ &\quad \{6000, -8000, 2000, -3000\} \\ list2 &:= \{2, 2, 2, 1\} \quad \{2, 2, 2, 1\} \\ \min(4.65, 12, 5000, list1, list2) &\quad 13.41608607 \end{aligned}$$

tal.arvo on kassavirtamääristä maksettava korkoprosentti.

uud.invest.arvo on korkokanta, jolla kassavirrat investoidaan uudelleen.

CF0 on alkuperäinen kassavirta aikana 0; arvon on oltava kokonaisluku.

CFLista on lista kassavirtamääristä ensimmäisen kassavirran *CF0* jälkeen.

CFFrekv on valinnainen 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.

Huomaa: Katso myös **irr()**, sivu 100.

mod()Katalogi > **mod**(*Laus1*, *Laus2*) $\Rightarrow$ lauseke**mod**(*Lista1*, *Lista2*) $\Rightarrow$ lista**mod**(*Matriisi1*, *Matriisi2*) $\Rightarrow$ matriisi

Laskee ensimmäisen argumentin modulo toinen argumentti identtisten yhtälöiden määrittelyn mukaisesti:

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

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

Kun toinen argumentti on ei-nolla, vastaus on periodinen tässä argumentissa. Vastaus on joko nolla tai samanmerkinen 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 161

$\text{mod}(7, 0)$	7
$\text{mod}(7, 3)$	1
$\text{mod}(-7, 3)$	2
$\text{mod}(7, -3)$	-2
$\text{mod}(-7, -3)$	-1
$\text{mod}(\{12, -14, 16\}, \{9, 7, -5\})$	$\{3, 0, -4\}$

mRow()Katalogi > **mRow**(*Laus*, *Matriisi1*, *Indeksi*) $\Rightarrow$ matriisi

Luo kopion *Matriisi1*:stä, jossa jokainen rivin *Indeksi* elementti *Matriisi1*:ssä on kerrottu lausekkeella *Laus*.

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

mRowAdd()Katalogi > **mRowAdd**(*Laus*, *Matriisi1*, *Indeksi1*, *Indeksi2*) $\Rightarrow$ matriisi

Luo kopion *Matriisi1*:stä, jossa jokainen rivin *Indeksi2* elementti *Matriisi1*:ssä on korvattu seuraavasti:

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

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

$$\text{Laus} \cdot \text{rivi Indeksi1} + \text{rivi Indeksi2}$$

MultRegKatalogi > **MultReg** *Y*, *X1*[, *X2*[, *X3*, ..., [, *X10*]]

Laskee listan Y moninkertaisen lineaarisen regression listojen $X1, X2, \dots, X10$ suhteen. Tulosten yhteenveto tallentuu *stat.results*-muuttuun. (Katso sivu 191.)

Kaikkien listojen on oltava samankokoisia.

Lisätietoja listassa olevien tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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 määrittelyn kerroin
stat. $\hat{y}$ Lista	$\hat{y}$ Lista = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Regressioyhtälön jäännökset

MultRegIntervals

MultRegIntervals $Y, X1[,X2[,X3, \dots, X10]]], XArvoLista[, CTaso]$

Laskee ennustetun y :n arvon, tason C ennustevälin yhdelle havainnolle ja tason C luottamusvälin keskiarvovasteelle.

Tulosten yhteenveto tallentuu *stat.results*-muuttuun. (Katso sivu 191.)

Kaikkien listojen on oltava samankokoisia.

Lisätietoja listassa olevien tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat. $\hat{y}$	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

Tulosmuuttuja	Kuvaus
stat.ME	Luottamusvälin virhemarginaali
stat.SE	Keskiarvovasteen keskivirhe
stat.LowerPred, stat.UpperrPred	Yhden havainnon ennusteväli
stat.MEPred	Ennustevälin virhemarginaali
stat.SEPred	Ennusteen keskivirhe
stat.bList	Regressiokertoimien lista, {b0,b1,b3,...}
stat.Resid	Regressioyhtälön jäännökset

MultRegTests

Katalogi > 

MultRegTests $Y, X1[,X2[,X3,...[,XI0]]]$

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 191.)

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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

Tulosmuuttuja	Kuvaus
stat.SSReg	Regression neliöiden summa
stat.MSReg	Regression keskineliö
stat.dfError	Virheen vapausasteet
stat.SSError	Virheen neliöiden summa
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.yList	$\hat{y}$ 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

BooleanLaus1 **nand** *BooleanLaus2* antaa vastauksena *Boolean* lausekkeen*BooleanList1*

nand *BooleanList2* antaa vastauksena *Boolean* listan*BooleanMatriisi1*

nand *BooleanMatriisi2* antaa vastauksena *Boolean* 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.

$x \geq 3$ and $x \geq 4$	$x \geq 4$
$x \geq 3$ nand $x \geq 4$	$x < 4$

Kokonaisluku1 **nand**

Kokonaisluku2 $\Rightarrow$ *kokonaisluku*

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 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 and 4	0
3 nand 4	-1
$\{1,2,3\}$ and $\{3,2,1\}$	$\{1,2,1\}$
$\{1,2,3\}$ nand $\{3,2,1\}$	$\{-2,-3,-2\}$

nCr()

Katalogi > 

nCr(Laus1, Laus2) $\Rightarrow$ *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) $\Rightarrow$ 1

nCr(Laus, negKokonaisluku) $\Rightarrow$ 0

nCr(Laus, posKokonaisluku) $\Rightarrow$ *Laus* · (*Laus*-1) ... (*Laus*-posKokonaisluku+1)/posKokonaisluku!

nCr(Laus, eiKokonaisluku) $\Rightarrow$ *lauseke*! / ((*Laus*-eiKokonaisluku)! · eiKokonaisluku!)

nCr(Lista1, Lista2) $\Rightarrow$ *lista*

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

nCr()Katalogi > 

Laskee listan kombinaatioista kahden listan sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia listoja.

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

Laskee matriisin kombinaatioista kahden matriisin sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia matriiseja.

$$\text{nCr}\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right) \quad \begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$$

nDerivative()Katalogi > 

nDerivative(*Laus1*, *Muutt=Arvo* [, *Aste*]) $\Rightarrow$ *arvo*

nDerivative(*Laus1*, *Muutt* [, *Aste*]) | *Muutt=Arvo* $\Rightarrow$ *arvo*

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**.

$\text{nDerivative}(x , x=1)$	1
$\text{nDerivative}(x , x) _{x=0}$	undef
$\text{nDerivative}(\sqrt{x-1}, x) _{x=1}$	undef

newList()Katalogi > 

newList(*numElementit*) $\Rightarrow$ *lista*

Antaa tuloksena listan, jonka koko on *numElementit*. Jokainen elementti on nolla.

$$\text{newList}(4) \quad \{0, 0, 0, 0\}$$

newMat()Katalogi > 

newMat(*numRivit*, *numSarakkeet*) $\Rightarrow$ *matriisi*

Antaa tuloksena nollamatriisin, jonka koko on *numRivit* ja *numSarakkeet*.

$$\text{newMat}(2, 3) \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

nfMax()Katalogi > 

nfMax(Laus, Muutt)⇒arvo
nfMax(Laus, Muutt, alaraja)⇒arvo
nfMax(Laus, Muutt, alaraja, yläraja)⇒arvo
nfMax(Laus, Muutt) alaraja≤**Muutt**
 ≤**yläraja**⇒arvo

$\text{nfMax}(x^2 - 2 \cdot x - 1, x)$	-1.
$\text{nfMax}(0.5 \cdot x^3 - x - 2, x, -5, 5)$	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)⇒arvo
nfMin(Laus, Muutt, alaraja)⇒arvo
nfMin(Laus, Muutt, alaraja, yläraja)⇒arvo
nfMin(Laus, Muutt) alaraja≤**Muutt**
 ≤**yläraja**⇒arvo

$\text{nfMin}(x^2 + 2 \cdot x + 5, x)$	-1.
$\text{nfMin}(0.5 \cdot x^3 - x - 2, x, -5, 5)$	-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(Laus1, Muutt, Ala, Ylä)⇒lauseke

$\text{nInt}(e^{-x^2}, x, -1, 1)$	1.49365
-----------------------------------	---------

Jos integroitava funktio *LausI* ei sisällä muita muuttujia kuin *Muutt*, ja jos *Ala* ja *Ylä* ovat vakioita, positiivinen ∞ tai negatiivinen $-\infty$, tällöin **nInt()** laskee likiarvon lausekkeesta $\int(LausI, Var, Ala, Ylä)$. Tämä likiarvo on integrandin välillä $Ala < Muutt < Ylä$ olevien joidenkin otosarvojen painotettu keskiarvo.

Tavoitteena on kuusi merkitsevää numeroa. Adaptiivinen algoritmi päättyy, kun näyttää todennäköiseltä, että tavoite on saavutettu, tai kun näyttää epätodennäköiseltä, että lisäotokset tuottaisivat merkittävää parannusta.

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.

Huomaa: Katso myös **j()**, sivu 240.

$$\begin{array}{|l|l|} \hline \text{nInt}(\cos(x), x, -\pi, \pi + 1.E-12) & -1.04144E-12 \\ \hline \int_{-\pi}^{\pi+10^{-12}} \cos(x) dx & -\sin\left(\frac{1}{1000000000000}\right) \\ \hline \end{array}$$

$$\begin{array}{|l|l|} \hline \text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x\right), x, 0, 1\right) & 3.30423 \\ \hline \end{array}$$

nom()

nom(efektiivinenKorko, CpY) $\Rightarrow$ arvo

Talouslaskentatoiminto, joka muuntaa efektiivisen vuosikoron *efektiivinenKorko* nimelliskoroksi, kun *CpY* määritetään korkojaksojen lukumääräksi vuodessa.

efektiivinenKorko on oltava reaalityyppi, ja *CpY*:n on oltava reaalityyppi > 0 .

Huomaa: Katso myös **eff()**, sivu 62.

$$\begin{array}{|l|l|} \hline \text{nom}(5.90398, 12) & 5.75 \\ \hline \end{array}$$

BooleanLaus1 **nor** *BooleanLaus2* antaa vastauksena *Boolean*

lausekkeen BooleanList1

nor *BooleanList2* antaa vastauksena

Boolean listan *BooleanMatriisi1*

nor *BooleanMatriisi2* antaa vastauksena

Boolean matriisin

$x \geq 3$ or $x \geq 4$	$x \geq 3$
$x \geq 3$ nor $x \geq 4$	$x < 3$

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* $\Rightarrow$ *kokonaisluku*

3 or 4	7
3 nor 4	-8
$\{1, 2, 3\}$ or $\{3, 2, 1\}$	$\{3, 2, 3\}$
$\{1, 2, 3\}$ nor $\{3, 2, 1\}$	$\{-4, -3, -4\}$

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 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).

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*.

Varmista, että riippumatonta muuttujaa ei ole määritetty. Esimerkiksi, jos $f1(x):=5$ ja $x:=3$, tällöin **normalLine**($f1(x),x,2$) antaa vastauksen "epätosi".

$\text{normalLine}(x^2,x,1)$	$\frac{3}{2} \frac{x}{2}$
$\text{normalLine}((x-3)^2-4,x,3)$	$x=3$
$\text{normalLine}\left(\frac{1}{x^3},x=0\right)$	0
$\text{normalLine}(\sqrt{ x },x=0)$	undef

normCdf()Katalogi > 

normCdf(alaraja,yläraja[,μ[,σ]])⇒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 $P(X \leq \text{yläraja})$, aseta *alaraja* = $-\infty$.

normPdf()Katalogi > 

normPdf(XArvo[,μ,σ])⇒luku, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

Laskee normaalijakauman pistetodennäköisyysfunktion määrittelyssä X arvossa määritetyille μ :lle ja σ :lle.

not (ei)

not *BooleanLaus* $\Rightarrow$ *Boolean lauseke*

Määrittää totuusarvon tosi, epätosi tai argumentin sievennetyn muodon.

not *Kokonaisluku1* $\Rightarrow$ *kokonaisluku*

Laskee reaalikokonaisluvun ykkösen komplementin. Sisäisesti *Kokonaisluku1* muunnetaan etumerkilliseksi, 64-bittiseksi binaariluvuksi. Jokaisen bitin arvo vaihtuu (0:sta tulee 1 ja päin vastoin) ykkösen komplementille. Tulokset näytetään kantalukutilan mukaisesti.

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.

not($2 \geq 3$)	true
not($x < 2$)	$x \geq 2$
not not <i>innocent</i>	<i>innocent</i>

Heksadesimaalisessa kantalukutilassa:

not 0h7AC36	0hFFFFFFFF853C9
-------------	-----------------

Binaarisessa kantalukutilassa:

0b100101 ►Base10	37
not 0b100101	
0b11111111111111111111111111111111 ►	
not 0b100101 ►Base10	-38

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 >****nPr(Laus1, Laus2)⇒lauseke**

Kokonaisluvulle *Laus1* ja *Laus2* sekä $Laus1 \geq Laus2 \geq 0$, **nPr()** on *Laus1*:n asioiden permutaatioiden lukumäärä, joita otetaan *Laus2*:n verran kerrallaan. Kumpikin argumentti voi olla kokonaisluku tai symbolinen lauseke.

$nPr(z, 3)$	$z \cdot (z-2) \cdot (z-1)$
$Ans z=5$	60
$nPr(z, -3)$	$\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$
$nPr(z, c)$	$\frac{z!}{(z-c)!}$
$Ans \cdot nPr(z-c, -c)$	1

nPr(Laus, 0)⇒1

nPr(Laus, negKokonaisluku)⇒ 1/
((Laus+1) · (Laus+2)...
(lauseke–negKokonaisluku))

nPr(Laus, posKokonaisluku)⇒ Laus ·
(Laus–1)... **(Laus–posKokonaisluku+1)**

nPr(Laus, eiKokonaisluku)⇒Laus! /
(Laus–eiKokonaisluku)!

nPr(Lista1, List2)⇒lista

Luo listan permutaatioista kahden listan sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia listoja.

$nPr(\{5, 4, 3\}, \{2, 4, 2\})$	$\{20, 24, 6\}$
---------------------------------	-----------------

nPr(Matriisi1, Matriisi2)⇒matriisi

Luo matriisin permutaatioista kahden matriisin sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia matriiseja.

$nPr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$
--	---

npv()**Katalogi >****npv(Korkoprosentti, CFO, CFLista [, CFFrekvi])**

Talouselaskentatoiminto, 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.

$list1 := \{6000, -8000, 2000, -3000\}$	$\{6000, -8000, 2000, -3000\}$
$list2 := \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$npv(10, 5000, list1, list2)$	4769.91

CF0 on alkuperäinen kassavirta aikana 0; arvon on oltava kokonaisluku.

CFLista on lista kassavirtamääristä ensimmäisen kassavirran *CF0* 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.

nSolve()

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* $\Rightarrow$ *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.

$\text{nSolve}(x^2 + 5 \cdot x - 25 = 9, x)$	3.84429
$\text{nSolve}(x^2 = 4, x = -1)$	-2.
$\text{nSolve}(x^2 = 4, x = 1)$	2.

Huomaa: Jos ratkaisuja on useita, voit yrittää löytää tietyn ratkaisun käyttämällä apuna arvausta.

$\text{nSolve}(x^2 + 5 \cdot x - 25 = 9, x) x < 0$	-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"

nSolve() yrittää määrittää joko yhden pisteen, jossa jännös on nolla, tai kaksi toisiaan suhteellisen lähellä olevaa pistettä, jossa jännöksen etumerkki on vastakkainen ja jännöksen 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()**.

O

OneVar [1,]*X*],[*Frekv*],[*Luokka*,*Sisällytä*]]

OneVar [*n*,]*X*1,*X*2[*X*3[,...[,*X*20]]]

Laskee yhden muuttujan tilaston enintään 20 listasta. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 272.

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

or (tai)

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.

$x \geq 3$ 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

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 määritysten 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 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 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.

Huomaa: Katso xor.

Heksadesimaalisessa kantelukutilassa:

0h7AC36 or 0h3D5F	0h7BD7F
-------------------	---------

Tärkeää: Nolla, ei O-kirjain.

Binaarisessa kantelukutilassa:

0b100101 or 0b100	0b100101
-------------------	----------

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

ord()**Katalogi** > **ord**(*Merkkijono*) $\Rightarrow$ *kokonaisluku***ord**(*Lista1*) $\Rightarrow$ *lista*

Määrittää merkkijonon *Merkkijono* ensimmäisen merkin numerokoodin tai luo listan jokaisen listaelementin ensimmäisistä merkeistä.

<code>ord("hello")</code>	104
<code>char(104)</code>	"h"
<code>ord(char(24))</code>	24
<code>ord({"alpha","beta"})</code>	{ 97,98 }

P**P► Rx()****Katalogi** > **P► Rx**(*rLaus*, θ *Laus*) $\Rightarrow$ *lauseke***P► Rx**(*rLista*, θ *Lista*) $\Rightarrow$ *lista***P► Rx**(*rMatriisi*, θ *Matriisi*) $\Rightarrow$ *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 °, G tai R.

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

Radiaanikulmatilassa:

<code>P►Rx(r,θ)</code>	$\cos(\theta) \cdot r$
<code>P►Rx(4,60°)</code>	2
<code>P►Rx({-3,10,1.3},{π/3,π/4},0)</code>	$\left\{ \frac{-3}{2}, 5 \cdot \sqrt{2}, 1.3 \right\}$

P► Ry()**Katalogi** > **P► Ry**(*rLaus*, θ *Laus*) $\Rightarrow$ *lauseke***P► Ry**(*rLista*, θ *Lista*) $\Rightarrow$ *lista***P► Ry**(*rMatriisi*, θ *Matriisi*) $\Rightarrow$ *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.

Radiaanikulmatilassa:

<code>P►Ry(r,θ)</code>	$\sin(\theta) \cdot r$
<code>P►Ry(4,60°)</code>	$2 \cdot \sqrt{3}$
<code>P►Ry({-3,10,1.3},{π/3,π/4},0)</code>	$\left\{ \frac{-3 \cdot \sqrt{3}}{2}, -5 \cdot \sqrt{2}, 0 \right\}$

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

PassErr

PassErr

Ohittaa virheen siirtyen seuraavalle tasolle.

Jos järjestelmän muuttuja *errCode* on nolla, **PassErr** ei tee mitään.

Else-lauseessa lohkoissa **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 27, ja **Try**, sivu 207.

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 207.

piecewise()

piecewise(*Laus1* [, *Ehto1* [, *Laus2* [, *Ehto2* [, ...]]]])

Laatii määritelmät paloittain määritellylle funktiolle listan muodossa. Voit luoda paloittain määriteltyjen funktioiden määrittäjiä 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(λ , alaraja, yläaraja) $\Rightarrow$ luku, jos alaraja ja yläaraja ovat lukuja, lista, jos alaraja ja yläaraja ovat listoja

poissCdf(λ , yläaraja) (kun $P(0 \leq X \leq \text{yläaraja}) \Rightarrow$ luku, jos yläaraja on luku, lista, jos yläaraja on lista

Laskee kumulatiivisen todennäköisyyden diskreetille Poissonin jakaumalle, jolla on määritetty keskiarvo λ .

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

poissPdf(λ , XArvo) $\Rightarrow$ luku, jos XArvo on luku, lista, jos XArvo on lista

Laskee todennäköisyyden diskreetille Poissonin jakaumalle, jolla on määritetty keskiarvo λ .

Vektori ► **Polar**

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

Näyttää *vektorin* polaarisisä muoossa [$r \angle \theta$]. Vektorin on oltava kooltaan 2, ja se voi olla rivi tai sarake.

Huomaa: ► **Polar** on näyttömuodon ohje, ei muunnosfunktio. Voit käyttää komentoa ainoastaan syöterivin lopussa, eikä se päivitä *ans*:n arvoa.

Huomaa: Katso myös ► **Rect**, sivu 158.

kompleksiArvo ► **Polaarinen**

Näyttää *kompleksiVektorin* polaarisisä muoossa.

- Astekulmatilassa vastauksena on ($r \angle \theta$).

[1 3.]►Polar	[3.16228 1.24905]
[x y]►Polar	$\left[\sqrt{x^2+y^2} \angle \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right) \right]$

Radiaanikulmatilassa:

- Radiaanikulmatilassa vastauksena on $\text{rei}\theta$.

kompleksiArvolla voi olla mikä tahansa kompleksilukumuoto. Syöte $\text{rei}\theta$ aiheuttaa kuitenkin virheen astekulmatilassa.

Huomaa: Polaarisisä syötteessä ($r\angle\theta$) on käytettävä sulkeita.

$$(3+4\cdot i)\blacktriangleright\text{Polar} \quad e^{i\cdot\left(\frac{\pi}{2}-\tan^{-1}\left(\frac{3}{4}\right)\right)}\cdot 5$$

$$\left(\left(4\angle\frac{\pi}{3}\right)\right)\blacktriangleright\text{Polar} \quad \frac{i\cdot\pi}{e^3}\cdot 4$$

Graadikulmatilassa:

$$(4\cdot i)\blacktriangleright\text{Polar} \quad (4\angle 100.)$$

Astekulmatilassa:

$$(3+4\cdot i)\blacktriangleright\text{Polar} \quad \left(5\angle 90-\tan^{-1}\left(\frac{3}{4}\right)\right)$$

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*.

$$\text{polyCoeffs}((x+y+z)^2,x) \quad \{1,2\cdot(y+z),(y+z)^2\}$$

$$\text{polyCoeffs}((x+y+z)^2,y) \quad \{1,2\cdot(x+z),(x+z)^2\}$$

$$\text{polyCoeffs}((x+y+z)^2,z) \quad \{1,2\cdot(x+y),(x+y)^2\}$$

polyDegree()

Katalogi > 

polyDegree(*Poly* [,*Muutt*]) $\Rightarrow$ *arvo*

Määrittää polynomilausekkeen *Poly* asteen muuttujan *Muutt* suhteen. Jos jätät muuttujan *Muutt* pois, **polyDegree**()-funktio valitsee oletusarvon polynomin *Poly* sisältämistä muuttujista.

Poly on oltava polynomilauseke muuttujassa *Muutt*. Suositeltavaa on, ettet jätä muuttujaa *Muutt* pois, ellei *Poly* ole lauseke yhdessä muuttujassa.

polyDegree(5)	0
polyDegree($\ln(2)+\pi,x$)	0

Vakiopolynomit

polyDegree($4 \cdot x^2 - 3 \cdot x + 2, x$)	2
polyDegree($(x-1)^2 \cdot (x+2)^3$)	5

polyDegree($(x+y^2+z^3)^2, x$)	2
polyDegree($(x+y^2+z^3)^2, y$)	4

polyDegree($(x-1)^{10000}, x$)	10000
----------------------------------	-------

Asteesta voidaan ottaa juuri, vaikka kertoimista ei voida. Tämä johtuu siitä, että asteesta voidaan ottaa juuri laimentamatta polynomia.

polyEval()

Katalogi > 

polyEval(*Lista1*, *Laus1*) $\Rightarrow$ *lauseke*

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

Tulkitsee ensimmäisen argumentin laskeva-asteisen polynomin kertoimeksi ja antaa vastauksena polynomin, josta on laskettu toisen argumentin arvo.

polyEval($\{a,b,c\}, x$)	$a \cdot x^2 + b \cdot x + c$
polyEval($\{1,2,3,4\}, 2$)	26
polyEval($\{1,2,3,4\}, \{2,-7\}$)	$\{26, -26\}$

polyGcd()

Katalogi > 

polyGcd(*Laus1*, *Laus2*) $\Rightarrow$ *lauseke*

Laskee kahden argumentin suurimman yhteisen jakajan.

Laus1:n ja *Laus2*:n on oltava polynomilausekkeita.

Lista-, matriisi- ja Boolean argumentteja ei sallita.

polyGcd(100,30)	10
polyGcd($x^2-1, x-1$)	$x-1$
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*]) $\Rightarrow$ lauseke

Laskee polynomien *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
$\text{polyQuotient}(x^2-1, x-1)$	$x+1$
$\text{polyQuotient}(x^3-6x^2+11x-6, x^2-6x+8)$	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*]) $\Rightarrow$ lauseke

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, ettet jätä muuttujaa *Muutt* pois, elleivät *Poly1* ja *Poly2* ole lausekkeita yhdessä ja samassa muuttujassa.

$\text{polyRemainder}(x-1, x-3)$	2
$\text{polyRemainder}(x-1, x^2-1)$	$x-1$
$\text{polyRemainder}(x^2-1, x-1)$	0
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, x)$	$-(y-z) \cdot (2 \cdot y + z)$
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, y)$	$-2 \cdot x^2 - 5 \cdot x \cdot z - 2 \cdot z^2$
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, z)$	$(x-y) \cdot (x+2 \cdot y)$

polyRoots(Poly,Muutt) $\Rightarrow$ lista

polyRoots(Kertoinlista) $\Rightarrow$ lista

Ensimmäinen syntaksi, **polyRoots(Poly,Muutt)**, laskee polynomin *Poly* reaalitylukujuurten listan muuttujan *Muutt* suhteen. Jos reaalitylukujuurta ei ole, tuloksena on tyhjä lista: {}.

Poly on oltava polynomi yhdessä muuttujassa.

Toinen syntaksi, **polyRoots(Kertoinlista)**, askee reaalitylukujuurten listan kertoimille, jotka sisältyvät *Kertoinlistaan*.

Huomaa: Katso myös **cPolyRoots()**, sivu 38.

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

PowerReg

PowerReg X,Y [, Frekv] [, Luokka, Sisällytä]]

Laskee potenssiregression $y = (a \cdot (x)^b)$ listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttuun. (Katso sivu 191.)

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 tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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	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

Prgm

Katalogi >

Prgm

Lohko

EndPrgm

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.

Laske GCD ja näytä välitulokset.

```

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

Prgm

Katalogi > 

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

$\text{proggcd}(4560,450)$	
	450 60
	60 30
	30 0
	GCD=30
	Done

prodSeq()

Katso $\Pi()$, sivu 242.

Product (PI)

Katso $\Pi()$, sivu 242.

product()

Katalogi > 

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.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 272.

$\text{product}(\{1,2,3,4\})$	24
$\text{product}(\{2,x,y\})$	$2 \cdot x \cdot y$
$\text{product}(\{4,5,8,9\},2,3)$	40
$\text{product}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	$\begin{bmatrix} 28 & 80 & 162 \end{bmatrix}$
$\text{product}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},1,2\right)$	$\begin{bmatrix} 4 & 10 & 18 \end{bmatrix}$

propFrac()

Katalogi > 

propFrac(LausI[, Muutt])⇒*lauseke*

$\text{propFrac}\left(\frac{4}{3}\right)$	$1+\frac{1}{3}$
$\text{propFrac}\left(\frac{-4}{3}\right)$	$-1-\frac{1}{3}$

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* samanklaiset 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.

$$\begin{array}{rcl} \text{propFrac}\left(\frac{x^2+x+1}{x+1}+\frac{y^2+y+1}{y+1},x\right) & & \\ & & \frac{1}{x+1}+x+\frac{y^2+y+1}{y+1} \\ \text{propFrac(Ans)} & & \frac{1}{x+1}+x+\frac{1}{y+1}+y \end{array}$$

$$\begin{array}{rcl} \text{propFrac}\left(\frac{11}{7}\right) & & 1+\frac{4}{7} \\ \text{propFrac}\left(3+\frac{1}{11}+5+\frac{3}{4}\right) & & 8+\frac{37}{44} \\ \text{propFrac}\left(3+\frac{1}{11}-\left(5+\frac{3}{4}\right)\right) & & -2-\frac{29}{44} \end{array}$$

Q

QR *Matriisi, qMatNimi, rMatNimi[, Tol]*

Liukuluku (9.) muuttujassa m1 aiheuttaa sen, että vastaukset lasketaan liukulukumuodossa.

Suorittaa Householderin QR-dekomponoinnin reaali- tai kompleksilukumatriisista. Tuloksena olevat Q- ja R-matriisit tallentuvat määritettyihin *MatNimi*-muuttujiin. Q-matriisi on jakamaton. R-matriisi on yläkolmionmatriisi.

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 **ctrl** **enter** 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.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
QR m1,qm,rm Done	
qm	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$
rm	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
QR m1,qm,rm Done	
qm	$\begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$
rm	$\begin{bmatrix} \sqrt{m^2+o^2} & \frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{ m \cdot p - n \cdot o }{\sqrt{m^2+o^2}} \end{bmatrix}$

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

Laskee 2. asteen polynomiregression $y = a \cdot x^2 + b \cdot x + c$ listaista X ja Y frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 272.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Regressiokertoimet
stat.R ²	Määrityskerroin
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

QuartReg *X,Y* [, *Frekv*] [, *Luokka*,
Sisällytä]]

Laskee 4. asteen polynomiregression
 $y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$ listaista *X* ja *Y*
 frekvenssillä *Frekv*. Tulosten yhteenveto
 tallentuu *stat.results*-muuttujaan.
 (Katso sivu sivu 191.)

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 tyhjiä
 elementtien vaikutuksesta, katso Tyhjät
 elementtisivulla sivu 272.

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	Muokatun <i>Y</i> Lista:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen <i>Frekv</i> , <i>Luokkalista</i> ja <i>Sisällytä</i> luokat rajoitusten mukaisesti

Tulosmuuttuja	Kuvaus
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

R

R ► P0()

Luettelo > 

R ► P0 (*xLaus*, *yLaus*) ⇒ *lauseke*

R ► P0 (*xLista*, *yLista*) ⇒ *lista*

R ► P0 (*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:

$$\text{R ► P0}(x,y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Graadikulmatilassa:

$$\text{R ► P0}(x,y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Radiaanikulmatilassa:

$$\begin{aligned} \text{R ► P0}(3,2) & \quad \tan^{-1}\left(\frac{2}{3}\right) \\ \text{R ► P0}\left([3 \ -4 \ 2], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right) & \quad \left[0 \ \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} \ 0.643501\right] \end{aligned}$$

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:

$$\begin{aligned} \text{R ► Pr}(3,2) & \quad \sqrt{13} \\ \text{R ► Pr}(x,y) & \quad \sqrt{x^2+y^2} \\ \text{R ► Pr}\left([3 \ -4 \ 2], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right) & \quad \left[3 \ \frac{\sqrt{\pi^2+256}}{4} \ 2.5\right] \end{aligned}$$

Lausl ► Rad ⇒ lauseke

Astekulmatilassa:

Muuntaa argumentin
radiaanikulmayksikköön.

$(1.5)^\text{►Rad}$	$(0.02618)^\text{r}$
---------------------	----------------------

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

Graadikulmatilassa:

$(1.5)^\text{►Rad}$	$(0.023562)^\text{r}$
---------------------	-----------------------

rand()

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()

randBin(n, p) ⇒ *lauseke*

randBin(n, p, Kokeiden lkm) ⇒ *lista*

randBin(n, p) laskee satunnaisen
reaaliluvun määrätystä
binomijakaumasta.

randBin(80,0,5)	42
randBin(80,0,5,3)	{ 41, 32, 39 }

randBin(n, p, Kokeiden lkm) laskee
listan, joka sisältää *Kokeiden lkm*
satunnaista reaalilukua määrätystä
binomijakaumasta.

randInt()

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

(*Muutt.alaraja*,
Muutt.yläraja) laskee
satunnaisen
kokonaisluvun
rajojen*Muutt.alaraja*
ja*Muutt.yläraja*
määääällä
kokonaislukuvälillä.

randInt

(*Muutt.alaraja*,
Muutt.yläraja
,*Kokeiden lkm*)
laskee listan, joka
sisältää*Kokeiden lkm*
satunnaista
kokonaislukua
määrätyllä välillä.

randMat()

randMat(*numRivit*, *numSarakkeet*) ⇒
matriisi

Laskee välillä -9 ja 9 määrätyle mitalle
kokonaislukumatriisin.

Kummankin argumentin on
sievennyttyvä kokonaisluvuksi.

RandSeed 1147	Done									
randMat(3,3)	<table><tr><td>8</td><td>-3</td><td>6</td></tr><tr><td>-2</td><td>3</td><td>-6</td></tr><tr><td>0</td><td>4</td><td>-6</td></tr></table>	8	-3	6	-2	3	-6	0	4	-6
8	-3	6								
-2	3	-6								
0	4	-6								

Huomaa: Tämän matriisin arvot muuttuvat
aina, kun painat .

randNorm()

randNorm(μ , σ) ⇒ *lauseke*
randNorm(μ , σ , *Kokeiden lkm*) ⇒ *lista*

randNorm(μ , σ) laskee desimaaliluvun
määrätystä normaalijakaumasta. Luku
voi olla mikä tahansa reaalityluku, mutta
se keskittyy voimakkaasti välille $[\mu-3\cdot\sigma$,
 $\mu+3\cdot\sigma]$.

randNorm(μ , σ , *Kokeiden lkm*) listan
(list), joka sisältää *Kokeiden lkm*
desimaalitylukua määrätystä
normaalijakaumasta.

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* määrätyssä 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 }
--------------	-----------------

RandSeedLuettelo > **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 reaalilukuosan.

Huomaa: Kaikkia määrittelemättömiä muuttujia käsitellään reaalimuuttujina. Katso myös **imag()**, page 96.

real(*Listl*) $\Rightarrow$ *lista*

Laskee kaikkien alkutekijöiden reaaliosat

real(*Matriisi1*) $\Rightarrow$ *matriisi*

$\text{real}(2+3 \cdot i)$	2
----------------------------	---

$\text{real}(z)$	z
------------------	-----

$\text{real}(x+i \cdot y)$	x
----------------------------	-----

$\text{real}(\{a+i \cdot b, 3, i\})$	$\{a, 3, 0\}$
--------------------------------------	---------------

$\text{real}\left(\begin{bmatrix} a+i \cdot b & 3 \\ c & i \end{bmatrix}\right)$	$\begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$
--	--

Laskee kaikkien alkutekijöiden reaalisosat.

► Rect

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 143.

kompleksiArvo ► Rect

Näyttää kompleksisen arvon *complexValues* suorakulmaisessa muodossa a+bi. Arvolla *complexValue* voi olla mikä tahansa kompleksinen muoto. Syöte $e^{i\theta}$ aiheuttaa kuitenkin virheen astekulmatilassa.

Huomaa: Polaarisisä syötteessä ($r \angle \theta$) on käytettävä sulkeita.

$$\left(\begin{bmatrix} 3 & \angle \frac{\pi}{4} & \angle \frac{\pi}{6} \end{bmatrix} \right) \text{► Rect} \quad \left[\begin{array}{ccc} \frac{3 \cdot \sqrt{2}}{4} & \frac{3 \cdot \sqrt{2}}{4} & \frac{3 \cdot \sqrt{3}}{2} \\ \frac{a \cdot \cos(b) \cdot \sin(c)}{a \cdot \sin(b) \cdot \sin(c)} & \frac{a \cdot \sin(b) \cdot \sin(c)}{a \cdot \cos(c)} & \frac{a \cdot \cos(c)}{a \cdot \sin(c)} \end{array} \right]$$

Radiaanikulmatilassa:

$$\left(\begin{bmatrix} \frac{\pi}{4 \cdot e^3} \end{bmatrix} \right) \text{► Rect} \quad \frac{\pi}{4 \cdot e^3}$$

$$\left(\begin{bmatrix} 4 \angle \frac{\pi}{3} \end{bmatrix} \right) \text{► Rect} \quad 2 + 2 \cdot \sqrt{3} \cdot i$$

Graadikulmatilassa:

$$\left(\begin{bmatrix} 1 \angle 100 \end{bmatrix} \right) \text{► Rect} \quad i$$

Astekulmatilassa:

$$\left(\begin{bmatrix} 4 \angle 60 \end{bmatrix} \right) \text{► Rect} \quad 2 + 2 \cdot \sqrt{3} \cdot i$$

Huomaa: Näppäilläksesi merkin $\angle$ valitse se Luettelon symboliluettelosta.

ref(MatriisiI[, Tol]) ⇒ *matriisi*

Laskee matriisiin *MatriisiI* rivi-echelon-muodossa.

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.

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

Vältä määrittelemättömiä alkutekijöitä matriisissa *MatriisiI*. Tuloksena saattaa olla odottamattomia vastauksia.

Jos esimerkiksi *a* on määrittelemätön seuraavassa lausekkeessa, näkyviin tulee varoitusviesti, ja vastaus näyttää seuraavalta:

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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.

$$\text{ref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow mI \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\text{ref}(mI) \quad \begin{bmatrix} 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix}$$



$$\text{ref} \left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) | a=0 \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Huomaa: Katso myös **rref()**, page 169.

RefreshProbeVars

RefreshProbeVars

Päaset mittausrvoon 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™i-laitteen keskiössä.

remain()Luettelo >

remain(Laus1, Laus2) ⇒ lauseke

remain(Lista1, Lista2) ⇒ lista

remain(Matriisi1, Matriisi2) ⇒ matriisi

Laskee ensimmäisen argumentin jäännöksen toisen argumentin suhteen seuraavien identtisten yhtälöiden määritelmien mukaisesti:

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

$\text{remain}(x,y) = x - y \cdot \text{iPart}(x/y)$

Tämän seurauksena huomaa, että **remain(-x,y) = remain(x,y)**. Vastaus on joko nolla tai samanmerkkinen kuin ensimmäinen argumentti.

Huomaa: Katso myös **mod()**, sivu 125.

$\text{remain}(7,0)$	7
$\text{remain}(7,3)$	1
$\text{remain}(-7,3)$	-1
$\text{remain}(7,-3)$	1
$\text{remain}(-7,-3)$	-1
$\text{remain}(\{12,-14,16\},\{9,7,-5\})$	$\{3,0,1\}$

$\text{remain}\left(\begin{bmatrix} 9 & -7 \\ 6 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 3 \\ 4 & -3 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$
--	---

PyydäLuettelo >

Pyydä *promptString*, *muutt*
[, *NäytäLippu* [, *statusVar*]]

Pyydä *promptString*, *func(arg1, ...argn)*
[, *NäytäLippu* [, *statusVar*]]

Ohjelmointikomento: Keskeyttää ohjelman ja näyttää valintaruudun, jossa on viesti *kehotemerkkijono* 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.

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:

Valinnainen *NäytäLippu* -argumentti voi olla mikä tahansa lauseke.

Säde: 6/2
Ala = 28.2743

- Jos *NäytäLippu* jätetään pois, tai jos se sieventyy arvoon **1**, kehoteviesti ja käyttäjän vastaus näkyvät laskimen historiassa.
- Jos *NäytäLippu* sieventyy arvoon **0**, kehote ja vastaus eivät näy historiassa.

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 muuttujasta *statusVar* 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:

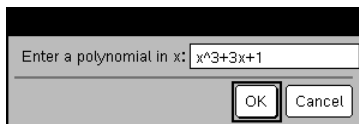
- **Kämmenlaite:** Pidä  -painiketta painettuna ja paina toistuvasti  -painiketta.
- **Windows®:** Pidä **F12**-näppäintä pohjassa ja paina toistuvasti **Enter**.

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: $\{-0.322185\}$

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

Huomaa: Katso myös **RequestStr**, page 163.

RequestStr

RequestStr *kehotemerkkijono, muutt[, NäytäLippu]*

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 161.

Määrittele ohjelma:

```
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(*List1*, *Num*) ⇒ *lista*

Laskee oikeanpuoleisimmat *Num*-alkutekijät, jotka sisältyvät listaan *List1*.

Jos jätät pois komennon *Num*, laskee kaiken listasta *List1*.

right(*lähdemerkkijono*, *Num*) ⇒ *merkkijono*

Laskee oikeanpuoleisimmat *Num*-merkit, jotka sisältyvät merkkijonoon *lähdemerkkijono*

Jos jätät pois komennon *Num*, laskee kaiken *lähdemerkkijonosta*.

right(*Vertailu*) ⇒ *lauseke*

Laskee yhtälön tai epäyhtälön oikean puolen.

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

right("Hello",2) "lo"

right(x<3) 3

rk23 ()

rk23(*Laus*, *Var*, *depVar*, {*Var0*, *VarMax*}, *depVar0*, *VarStep* [, *diftoI*]) ⇒ *matriisi*

rk23(*SystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep* [, *diftoI*]) ⇒ *matriisic*

Differentiaaliyhtälö:

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

rk23{ $0.001 \cdot y \cdot (100 - y)$, 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*, *diftoI*) $\Rightarrow$ *matriisi*

Käyttää Runge-Kutta-menetelmää järjestelmän ratkaisuun

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

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.

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 haluat nähdä koko vastauksen, paina $\blacktriangle$ ja siirrä sen jälkeen osoitinta painikkeilla $\blacktriangleleft$ ja $\blacktriangleright$.

Sama yhtälö, jossa *diftoI* on asetettu 1.E-6

$$\text{rk23}\left(0.001 \cdot y \cdot \{100-y\}, t, y, \{0, 100\}, 10, 1, 1.E-6\right) \\ \left[\begin{array}{ccccc} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9367 & 11.9495 & 13.0423 & 14.2189 \end{array} \right]$$

Vertaile edellä olevaa tulosta CAS:n täsmälliseen tulokseen, joka on saatu käyttämällä *deSolve*()- ja *seqGEN*()-funktioita:

$$\text{deSolve}\left(y'=0.001 \cdot y \cdot \{100-y\} \text{ and } y(0)=10, t, y\right) \\ y = \frac{100 \cdot \{1.10517\}^t}{\{1.10517\}^t + 9.}$$

$$\text{seqGen}\left(\frac{100 \cdot \{1.10517\}^t}{\{1.10517\}^t + 9.}, t, y, \{0, 100\}\right) \\ \{10., 10.9367, 11.9494, 13.0423, 14.2189, 15.49\}$$

Yhtälöryhmä:

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

kun *y1*(θ)=2 and *y2*(θ)=5

$$\text{rk23}\left(\left\{ \begin{array}{l} y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{array} \right\}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right) \\ \left[\begin{array}{ccccc} 0. & 1. & 2. & 3. & 4. \\ 2. & 1.94103 & 4.78694 & 3.25253 & 1.82848 \\ 5. & 16.8311 & 12.3133 & 3.51112 & 6.27245 \end{array} \right]$$

rk23 ()

Luettelo >

Jos $VarStep$ on arvoltaan nolasta eroava numero: merkki($VarStep$) = merkki($VarMax - Var0$) ja ratkaisu lasketaan muuttujalla $Var0 + i * VarStep$ kaikille $i=0,1,2,\dots$ niin, että $Var0 + i * VarStep$ on alueella [$muutt0, VarMax$] (muuttujalla $VarMax$ ei ehkä ole ratkaisuarvoa).

jos *VarStep* sievenee nolaksi, ratkaisut lasketaan "Runge-Kutta" -muuttujan *Muutt* arvoilla.

diftol on virhetoleranssi (oletuksena 0.001).

root()

Luettelo > 

root(*Laus*) $\Rightarrow$ *juuri*

$$\text{root}(Laus1, Laus2) \Rightarrow juuri$$

root(*Laus*) laskee neliöjuuren lausekkeelle *Laus*.

root(*Laus1*, *Laus2*) laskee lausekkeena *Laus2* lausekkeen *Laus1* juuren. *Laus1* voi olla reaallinen tai kompleksinen liukulukuvakio, kokonaisluku tai kompleksinen rationaalilukuvakio tai yleinen symbolinen lauseke.

Huomaa: Katso myös N:s juuri -malli, sivu 2.

$\sqrt[3]{8}$	2
$\sqrt[3]{3}$	$\frac{1}{3^3}$
$\sqrt[3]{3.}$	1.44225

rotate()

Luettelo > 

```
rotate(KokonaislukuI[,KiertojenLkm]) => kokonaisluku
```

Binaarisessa kantalukutilassa:

Kiertää bittejä binaarisessa kokonaisluvussa. Voit syöttää luvun *Kokonaisluku1* minä tahansa kantalukuna; se muunnetaan automaattisesti 64 bitin binaarimuotoon. Jos *Kokonaisluku1* on liian suuri tälle muodolle, symmetrinen modulo-operaatio sovittaa sen alueelle sopivaksi Katso lisätieto kohdasta ► **Kantaluku2**, sivu 18.

```
rotate(0b11111111111111111111111111111111)
0b1000000000000000000000000000000001
rotate(256,1)                                0b1000000000
```

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

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:

0b10000000000000111101011000011010

Vastaus näkyy kantalukutilan mukaisesti.

rotate(Lista1[,KiertojenLkm]) $\Rightarrow$ lista

Laskee *Listal:n* kopion, jota on kierretty oikealle tai vasemmalle *KiertojenLkm:n* elementtien määrittämän määrän. Ei muuta *Listaal*.

Jos *KiertojenLkm* on positiivinen, kierto on vasemmalle. Jos *KiertojenLkm* on negatiivinen, kierto on oikealle. Oletusarvo on -1 (kierrä oikealle yksi elementti).

rotate(Merkkijono1[,KiertojenLkm]) $\Rightarrow$ merkkijono

Laskee *Merkkijono1:n* kopion, jonka on kiertänyt oikealle tai vasemmalle *KiertojenLkm:n* merkkien määrittämän määrän. Ei muuta *Merkkijono1:tä*

Jos *KiertojenLkm* on positiivinen, kierto on vasemmalle. Jos *KiertojenLkm* on negatiivinen, kierto on oikealle. Oletusarvo on -1 (kierrä oikealle yksi merkki).

Heksadesimaalisessa kantalukutilassa:

rotate(0h78E)	0h3C7
rotate(0h78E,-2)	0h80000000000001E3
rotate(0h78E,2)	0h1E38

Tärkeää: Binaariluvun edelle tulee aina merkitä etumerkki 0b ja heksadesimaaliluvun edelle 0h (nolla, ei 0-kirjain).

Desimaalisessa kantalukutilassa:

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

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"bcda"

round(Laus1[, numeroa]) $\Rightarrow$ lauseke

round(1.234567,3)	1.235
-------------------	-------

Laskee argumentin pyöristettynä jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

round()

Luettelo > 

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(Lista1[, numerot]) ⇒ lista

Pyöristää alkutekijöiden listan jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

round(Matriisi1[, numerot]) ⇒ matriisi

Pyöristää alkutekijöiden matriisin jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

$$\text{round}\left(\left\{\pi, \sqrt{2}, \ln(2)\right\}, 4\right) \\ \{3.1416, 1.4142, 0.6931\}$$

$$\text{round}\left(\begin{pmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{pmatrix}, 1\right) \quad \begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$$

rowAdd()

Luettelo > 

rowAdd(Matriisi1, rIndeksi1, rIndeksi2)
⇒ matriisi

Laskee matriisiin *Matriisi1* kopion rivin *rIndeksi2* korvattuna rivien summalla *rIndeksi1* ja *rIndeksi2*.

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

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

rowDim()

Luettelo > 

rowDim(Matriisi) ⇒ lauseke

Laskee *Matriisin* sisältämien rivien lukumäärän.

Huomaa: Katso myös **colDim()**, sivu 28.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \\ \text{rowDim}(m1) \quad 3$$

rowNorm()

Luettelo > 

rowNorm(Matriisi) ⇒ lauseke

Laskee *Matriisin* riveillä olevien elementtien itseisarvojen summien maksimin.

Huomaa: Kaikkien matriisien alkutekijöiden on sievennyttävä luvuiksi. Katso myös **colNorm()**, sivu 28.

$$\text{rowNorm}\left(\begin{pmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{pmatrix}\right) \quad 25$$

rowSwap()

Luettelo > 

rowSwap(*Matriisi1*, *rIndeksi1*,
rIndeksi2) $\Rightarrow$ *matriisi*

Laskee matriisiin *Matriisi1*, jossa rivien *rIndeksi1* ja *rIndeksi2* paikkoja on vaihdettu.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
rowSwap(mat,1,3)	$\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$

rref()

Luettelo > 

rref(*Matriisi1*[, *Tol*]) $\Rightarrow$ *matriisi*

Laskee matriisiin *Matriisi1* pelkistetyn rivi-echelon-muodon.

$\text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$
--	---

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 määritetty arvoa. Muussa tapauksessa kommentoa *Tol* ei huomioida.

 $\text{rref}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
--	--

- 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:
 $5E-14 \cdot \max(\dim(\text{Matriisi1}))$
 $\cdot \text{rowNorm}(\text{Matriisi1})$

Huomaa: Katso myös **ref()**, page 159.

S

sec()

 -painike

sec(*Laus1*) $\Rightarrow$ *lauseke*

Astekulmatilassa:

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

sec()

 -painike

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 °, G tai r .

sec⁻¹()

 -painike

sec⁻¹(*LausI*) ⇒
lauseke

Astekulmatilassa:

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

sec⁻¹(*ListaI*) ⇒ *lista*

Määrittää kulman, jonka sekantti on *LausI*, tai määrittää listan, joka sisältää *ListaI*: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 (<i>LausI</i>) ⇒ <i>lauseke</i>	$\text{sech}(3) \quad \frac{1}{\cosh(3)}$
sech (<i>Listal</i>) ⇒ <i>lista</i>	$\text{sech}(\{1,2,3,4\}) \quad \left\{ \frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)} \right\}$
Laskee <i>LausI</i> :n hyperbolisen sekantin tai määrittää listan, joka sisältää <i>Listal</i> :n kaikkien elementtien hyperboliset sekantit.	

sech ⁻¹ ()	Katalogi > 	
sech⁻¹ (<i>LausI</i>) ⇒ <i>lauseke</i>	Radiaanikulma- ja suorakulmakompleksitilassa:	
sech⁻¹ (<i>Listal</i>) ⇒ <i>lista</i>		
Laskee <i>LausI</i> :n käänteisen hyperbolisen sekantin tai määrittää listan, joka sisältää <i>Listal</i> :n kaikkien elementtien käänteiset hyperboliset sekantit.		
Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla arcsech (...).		
	$\text{sech}^{-1}(1) \quad 0$	
	$\text{sech}^{-1}(\{1, -2, 2, 1\}) \quad \left\{ 0, \frac{2 \cdot \pi}{3} \cdot i, 8 \cdot \text{E-}15 + 1.07448 \cdot i \right\}$	

Send	Laite-valikko
Send <i>lausTaiMerkkijono1</i> [, <i>lausTaiMerkkijono2</i>] ...	Esimerkki: Sytytä sisäänrakennetun RGB-ledin sininen väri 0,5 sekunniksi.
Ohjelmointikomento: Lähettää yhden tai useamman TI-Innovator™ Hub komennon liitettyyn laitteeseen.	Send "SET COLOR.BLUE ON TIME .5" Done
<i>lausTaiMerkkijono</i> n täytyy olla validi TI-Innovator™ Hub Komento. Tavallisesti <i>lausTaiMerkkijono</i> sisältää "SET ..."komento laitteen tai "READ ..."hallitsemiseksi komento tiedon kysymiseksi.	Esimerkki: Kysy laitteen sisäänrakennetun valotasoaunturin tämänhetkinen arvo. Get -komento noutaa arvon ja sijoittaa sen muuttujaan <i>lightval</i> .
Argumentit lähetetään laitteeseen peräkkäisinä.	Send "READ BRIGHTNESS" Done Get <i>lightval</i> Done <i>lightval</i> 0.347922
Huomio: Komentoa Send voi käyttää käyttäjän määrittelemän ohjelman sisällä mutta ei funktion sisällä.	Esimerkki: Lähetä laskettu taajuus laitteen sisäänrakennettuun kaiuttimeen. Käytä erikoismuuttujaa <i>iostr.SendAns</i> näyttääksesi laitekomennon ilmaus arvioituna.
Huomio: Katso myös Get (sivu 84), GetStr (sivu 91), ja eval() (sivu 66).	

$n:=50$	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 .

$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	1.54977
--	---------

seqGen()

Katalogi >

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 sekvenssin 5 ensimmäistä termiä $u(n) = u(n-1)^2/2$, kun $u(1)=2$ ja $\text{VarStep}=1$.

$\text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right)$	$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$
---	---

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.

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:

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u2(n-1)}{2} + u1(n-1)\right\}, n, \{u1, u2\}, \{1, 5\}, \left[\begin{array}{c} 1 \\ 2 \end{array}\right]\right)$$

$$\left[\begin{array}{ccccc} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{array}\right]$$

Huomaa: Muuttujaa Tyhjä () yllä olevassa aloitustermimatriisissa käytetään ilmoittamaan, että *u1*(*n*):n aloitustermi on laskettu käyttämällä täsmällistä sekvenssikaavaa *u1*(*n*)=1/*n*.

seqn()

seqn(*Expr*{*u*, *n* [, *ListOfInitTerms* [, *nMax* [, *CeilingValue*]]])⇒*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*]]})⇒*list*

Luo sekvenssin 6 ensimmäistä termiä *u*(*n*) = *u* (*n*-1)/2, kun *u*(1)=2.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

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 $nstep=1$

series()

series(LausI, Muutt, Aste [, Piste])⇒muuttuja

series(LausI, Muutt, Aste [, Piste]) | Muutt>Piste⇒lauseke

series(LausI, Muutt, Aste [, Piste]) | Muutt<Piste⇒lauseke

Laskee yleisen tyypistetyt potenssisarjaesityksen *LausI*:stä, joka on lavennettu *Pisteen* 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.

Pisteen oletusarvo on 0. *Piste* voi olla ∞ tai $-\infty$, jolloin lavennus tapahtuu asteen *Aste* kautta lausekkeessa $1/(Muutt - Piste)$.

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1\right) \quad \frac{1}{2} \cdot \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$$

$$\text{series}\left(\frac{-1}{e^{z-}}, z, -1\right) \quad z - 1$$

$$\text{series}\left(\left(1 + \frac{1}{n}\right)^n, n, 2, \infty\right) \quad e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$$

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5\right), x > 0 \quad \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$$

$$\text{series}\left(\int \frac{\sin(x)}{x} dx, x, 6\right) \quad x - \frac{x^3}{18} + \frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7\right) \quad \frac{x^3}{2} - \frac{x^5}{24} - \frac{29 \cdot x^7}{720}$$

$$\text{series}\left(\left(1 + e^x\right)^2, x, 2, 1\right) \quad (e+1)^2 + 2 \cdot e \cdot (e+1) \cdot (x-1) + e \cdot (2 \cdot e+1) \cdot (x-1)^2$$

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 hyppyepäjatkuvuus kohdassa *Piste*, tulos sisältää todennäköisesti alalausekkeita, jotka ovat muotoa **sign(...)** tai **abs(...)** reaaliselle kehitelmän muuttujalle tai **(-1)^{floor(...angle(...))}** kompleksille kehitelmän muuttujalle, joka on merkkiin "**_**" päättyvä muuttuja. Jos tarkoituksesi on käyttää sarjaa vain *Pisteen* 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.

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 60.

setMode(tilaNimiKokonaisluku, asetusKokonaisluku) $\Rightarrow$ kokonaisluku
setMode(lista) $\Rightarrow$ 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.

Jos olet tallentanut kaikki tila-asetukset komennolla **getMode(0)** $\rightarrow$ *muutt*, voit palauttaa nämä asetukset komennolla **setMode(muutt)** aina funktion tai ohjelman sulkemiseen saakka. Katso **getMode()**, sivu 89.

Huomaa: Nykyiset tila-asetukset siirtyvät haettuihin alarutiineihin. Jos jokin alarutiini muuttaa tila-asetusta, tilamuutos häviää, kun ohjaus palautuu hakurutiiniin.

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>progI()</i> = Prgm	Done
Disp approx(π)	
setMode(1,16)	
Disp approx(π)	
EndPrgm	
<i>progI()</i>	
	3.14159
	3.14
	Done

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()

shift(*Kokonaisluku*
[,*SiirtojenLkm*])⇒*kokonaisluku*

Binaarisessa kantalukutilassa:

```
shift(0b1111010110000110101)
                                0b111101011000011010
shift(256,1)                    0b1000000000
```

Heksadesimaalisessa kantalukutilassa:

Siirtää binaarisen kokonaisluvun bittejä. Voit syöttää *Kokonaisluku1*:n minä tahansa kantalukuna; se muunnetaan automaattisesti etumerkilliseen 64 bitin binaarimuotoon. Jos *Kokonaisluku1* 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 bittiä. Vasemmalle tapahtuvassa siirrossa vasemmanpuoleisin bitti pudotetaan, ja 0 lisätään vastaamaan oikeanpuoleisinta bittiä.

Esimerkki siirrosta oikealle:

Jokainen bitti siirtyy oikealle.

0b00000000000000111101011000011010

Lisää 0:n, jos vasemmanpuoleisin bitti on 0,
tai 1:n, jos vasemmanpuoleisin bitti on 1.

tuottaa:

0b000000000000000111101011000011010

Vastaus näkyy kantalukutilan mukaisesti. Alkunollia ei näytetä.

shift(Lista1 [,SiirtojenLkm])⇒*lista*

Luo *Listal*:n kopion, jota on siirretty oikealle tai vasemmalle *SiirtojenLkm*:n elementtien määrittämän määrän. Ei muuta *Listal*:ä.

Jos *SiirtojenLkm* on positiivinen, siirto tapahtuu vasemmalle. Jos *SiirtojenLkm* on negatiivinen, siirto tapahtuu oikealle. Oletusarvo on -1 (siirrä yksi elementti oikealle).

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 0-kirjain).

Desimaalisessa kantalukutilassa:

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}

shift()Katalogi > 

Elementit, jotka siirto vie *listan* alkuun tai loppuun, asettuvat symboliksi "undef".

shift(*Merkkijono* *l*,*SiirtojenLkm*) $\Rightarrow$ *merkkijono*

$\text{shift}(\text{"abcd"})$	" abc "
$\text{shift}(\text{"abcd"}, -2)$	" ab "
$\text{shift}(\text{"abcd"}, 1)$	"bcd "

Luo *Merkkijono* *l*:n kopion, jota on siirretty oikealle tai vasemmalle *SiirtojenLkm*:n merkkien määrittämän määrän. Ei muuta *Merkkijono* *l*: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.

sign() (etumerkki)Katalogi > 

sign(*Laus* *l*) $\Rightarrow$ *lauseke*

$\text{sign}(-3.2)$	-1.
$\text{sign}(\{2, 3, 4, -5\})$	$\{1, 1, 1, -1\}$
$\text{sign}(1 + x)$	1

sign(*Lista* *l*) $\Rightarrow$ *lista*

sign(*Matriisi* *l*) $\Rightarrow$ *matriisi*

Kun kyseessä on reaali- tai kompleksilukulauseke *Laus* *l*, antaa vastauksena *Laus* *l*/**abs**(*Laus* *l*), kun *Laus* *l* $\neq$ 0.

Jos kompleksilukutila on Reaali:

$\text{sign}(\begin{bmatrix} -3 & 0 & 3 \end{bmatrix})$	$\begin{bmatrix} -1 & \pm 1 & 1 \end{bmatrix}$
---	--

Vastaus on 1, jos *Laus* *l* on positiivinen. Vastaus on -1, jos *Laus* *l* on negatiivinen.

sign(0) edustaa kompleksialueen yksikköpiiriä.

Kun kyseessä on lista tai matriisi, määrittää kaikkien elementtien etumerkit.

simult()Katalogi > 

simult(*kerroinMatriisi*, *vakioVektori* *l*,*tol*) $\Rightarrow$ *matriisi*

Ratkaise yhtälöstä x ja y:
 $x + 2y = 1$
 $3x + 4y = -1$

Määrittää sarakevektorin, joka sisältää lineaarisen yhtälöryhmän ratkaisut.

Huomaa: Katso myös **linSolve()**, sivu 110.

kerroinMatriisi 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.

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

Ratkaisu on $x=-3$ and $y=2$.

Ratkaise:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \text{matx1} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\text{simult}\left(\text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \quad \begin{bmatrix} -(2 \cdot b - d) \\ a \cdot d - b \cdot c \\ 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix}$$

Ratkaise:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right) \quad \begin{bmatrix} -3 & -7 \\ 2 & \frac{9}{2} \end{bmatrix}$$

Ensimmäisessä ryhmässä $x=-3$ ja $y=2$. Toisessa ryhmässä $x=-7$ ja $y=9/2$.

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

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

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.

sin()

 -painike

sin(Laus I) ⇒ lauseke

Astekulmatilassa:

sin(Lista I) ⇒ lista

$$\sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2}$$

sin(Laus I) määrittää argumentin sinin lausekkeena.

$$\sin(45^\circ) = \frac{\sqrt{2}}{2}$$

sin(Lista I) määrittää listan, joka sisältää *Listal*:n kaikkien elementin sinin.

$$\sin(\{0, 60, 90\}) = \left\{0, \frac{\sqrt{3}}{2}, 1\right\}$$

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

Graadikulmatilassa:

$$\sin(50^\circ) = \frac{\sqrt{2}}{2}$$

Radiaanikulmatilassa:

$$\sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2}$$

$$\sin(45^\circ) = \frac{\sqrt{2}}{2}$$

sin(neliömatriisiI)⇒neliömatriisi

Määrittää *neliömatriisiI*:n matriisin sinin. Tämä ei ole sama kuin kunkin elementin sinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Radiaanikulmatilassa:

$$\sin\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

sin⁻¹(LausI)⇒lauseke

sin⁻¹(ListaI)⇒lista

sin⁻¹(LausI) määrittää lausekkeena kulman, jonka sini on *LausI*.

sin⁻¹(ListaI) määrittää listan, joka sisältää *ListaI*: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(neliömatriisiI)⇒neliömatriisi

Laskee *neliömatriisiI*:n matriisin käänteissinin. Tämä ei ole sama kuin kunkin elementin käänteissinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Astekulmatilassa:

$$\sin^{-1}(1) = 90$$

Graadikulmatilassa:

$$\sin^{-1}(1) = 100$$

Radiaanikulmatilassa:

$$\sin^{-1}(\{0,0.2,0.5\}) = \{0,0.201358,0.523599\}$$

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

$$\sin^{-1}\left(\begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}\right) = \begin{bmatrix} -0.174533-0.12198 \cdot i & 1.74533-2.35591 \cdot i \\ 1.39626-1.88473 \cdot i & 0.174533-0.593162 \cdot i \end{bmatrix}$$

sinh(*Laus I*) $\Rightarrow$ *lauseke* $\sinh(1.2)$ 1.50946**sinh**(*Lista I*) $\Rightarrow$ *lista* $\sinh(\{0,1.2,3\})$ $\{0,1.50946,10.0179\}$

sinh (*Laus I*) laskee argumentin hyperbolisen kosinin lausekkeena.

sinh (*Lista I*) määrittää listan, joka sisältää *Lista I*:n jokaisen elementin hyperbolisen sinin.

sinh(*neliömatrissi I*) $\Rightarrow$ *neliömatrissi*

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

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

Radiaanikulmatilassa:

$$\sinh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$

sinh⁻¹(*Laus I*) $\Rightarrow$ *lauseke* $\sinh^{-1}(0)$ 0**sinh⁻¹**(*Lista I*) $\Rightarrow$ *lista* $\sinh^{-1}(\{0,2.1,3\})$ $\{0,1.48748,\sinh^{-1}(3)\}$

sinh⁻¹(*Laus I*) laskee argumentin käänteisen hyperbolisen sinin lausekkeena.

sinh⁻¹(*Lista I*) määrittää listan, joka sisältää *Lista I*:n jokaisen elementin käänteiset hyperboliset sinit.

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

sinh⁻¹(*neliömatrissi I*) $\Rightarrow$ *neliömatrissi*

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

Radiaanikulmatilassa:

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

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

SinReg

SinReg *X*, *Y* [, [*Iteraatiot*] , [*Jakso*] [,
Luokka, *Sisällytä*]]

Laskee siniregression listoista *X* ja *Y*. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Kaikkien listojen on oltava samankokoisia *Sisällytä*-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 arvioitun 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..

Sisällytä on yhden tai usemman 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 tyhjien elementtien vaikutuksesta, katso Tyhjät elementtisivulla sivu 272.

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	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

solve()

Katalogi > 

solve(Yhtälö, Muutt)⇒*Boolean lauseke*

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

solve(Epäyhtälö, Muutt)⇒*Boolean lauseke*

Laskee yhtälön tai epäyhtälön mahdollisia reaaliuratkaisuja 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.

$$\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x)$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a} \text{ or } x = \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a}$$

Ans| $a=1$ and $b=1$ 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 reaaliukuratkaisuja 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.

$$(x+1) \cdot \frac{x-1}{x-1} + x - 3 \quad 2 \cdot x - 2$$

$$\text{solve}(5 \cdot x - 2 \geq 2 \cdot x, x) \quad x \geq \frac{2}{3}$$

$$\text{exact}(\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x)) \quad e^x + x = 0 \text{ or } x = a$$

Radiaanikulmatilassa:

$$\text{solve}\left(\tan(x) = \frac{1}{x}, x\right) | x > 0 \text{ and } x < 1 \quad x = 0.860334$$

$$\begin{array}{ll} \text{solve}(x = x + 1, x) & \text{false} \\ \text{solve}(x = x, x) & \text{true} \end{array}$$

$$2 \cdot x - 1 \leq 1 \text{ and } \text{solve}(x^2 \neq 9, x) \quad x \neq -3 \text{ and } x \leq 1$$

Radiaanikulmatilassa:

$$\text{solve}(\sin(x) = 0, x) \quad x = n1 \cdot \pi$$

Reaalilukutilassa murtolukupotenssit, joilla on pariton nimittäjä, viittaavat ainoastaan reaalilukualueeseen. Muussa tapauksessa haaralausekkeet, kuten murtolukupotenssit, logaritmit ja käänteiset trigonometriset funktiot, viittaavat vain päähaaraan. Näin ollen **solve()** tuottaa vain ratkaisuja, jotka vastaavat tätä yhtä reaaliluku- tai päähaaraa.

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

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

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

**solve({Yht1, Yht2 [...]}
{MuuttTaiArvaus1, MuuttTaiArvaus2 [,
...]})** ⇒ *Boolean lauseke*

Antaa ehdotuksia reaalilukuratkaisuista 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 $x=3$.

$\text{solve}\left(x^{\frac{1}{3}}=1,x\right)$	$x=1$
$\text{solve}\left(\sqrt{x}=2,x\right)$	false
$\text{solve}\left(\sqrt{x}=-2,x\right)$	$x=-4$

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

Jos kaikki yhtälöt ovat polynomeja, ja jos ET määritä ensimmäisiä arvauksia, **solve()** käyttää leksikaalista Gröbner/Buchbergerin eliminaatiomenetelmää yrittäessään määrittää kaikki reaaliuratkaisut.

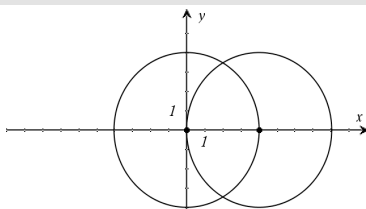
Oletetaan esimerkiksi, että origossa on ympyrän säde r , ja toinen ympyrän säde r on keskitetty kohtaan, jossa ensimmäinen ympyrä leikkaa positiivisen x -akselin. Määritä leikkauskohdat **solve()**-funktion avulla.

Kuten on kuvattu merkinnällä r oikealla olevassa esimerkissä, samanaikaisissa polynomi yhtälöissä 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 kokonaislukuliite väliltä 1-255.

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



$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2}$$

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1$$

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

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 reaalilukuratkaisuja.

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.

Arvausten avulla voit etsiä lisäratkaisuja yksi kerrallaan. Suppenemista varten arvauksen on mahdollisesti oltava melko lähellä ratkaisua.

$$\text{solve}\left(x+e^z \cdot y=1 \text{ and } x-y=\sin(z), \{x, y\}\right)$$

$$x=\frac{e^z \cdot \sin(z)+1}{e^z+1} \text{ and } y=\frac{-(\sin(z)-1)}{e^z+1}$$

$$\text{solve}\left(e^z \cdot y=1 \text{ and } -y=\sin(z), \{y, z\}\right)$$

$$y=2.812\text{E-}10 \text{ and } z=21.9911 \text{ or } y=0.001871\text{I}$$

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

$$\text{solve}\left(e^z \cdot y=1 \text{ and } -y=\sin(z), \{y, z=2 \cdot \pi\}\right)$$

$$y=0.001871 \text{ and } z=6.28131$$

SortA

SortA *List1* [, *List2*] [, *List3*] ...
SortA *Vektori1* [, *Vektori2*] [, *Vektori3*]
 ...

Lajittelee ensimmäisen argumentin elementit nousevaan järjestykseen.

Jos otat mukaan lisäargumentteja, 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 272.

$\{2,1,4,3\} \rightarrow \text{list1}$	$\{2,1,4,3\}$
SortA list1	Done
list1	$\{1,2,3,4\}$
$\{4,3,2,1\} \rightarrow \text{list2}$	$\{4,3,2,1\}$
SortA list2,list1	Done
list2	$\{1,2,3,4\}$
list1	$\{4,3,2,1\}$

SortD

Katalogi >

SortD *List1* [, *List2*] [, *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 272.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
SortD <i>list1</i> , <i>list2</i>	Done
<i>list1</i>	$\{4,3,2,1\}$
<i>list2</i>	$\{3,4,1,2\}$

►Sphere

Katalogi >

Vektori ►Sphere

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

Näyttää rivi- tai sarakevektorin pallonmuotoisena [ρ $\angle$ θ $\angle$ ϕ].

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.

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina .

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

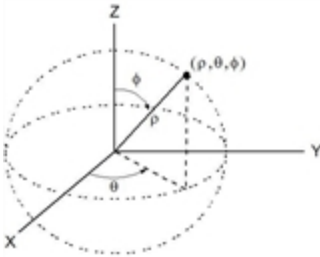
iPad®: Pidä **enter** ja valitse .

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \text{►Sphere} \\ \left[3.74166 \quad \angle 1.10715 \quad \angle 0.640522 \right]$$

$$\left(\begin{bmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{bmatrix} \right) \text{►Sphere} \\ \left[3.60555 \quad \angle 0.785398 \quad \angle 0.588003 \right]$$

Paina

$$\left(\begin{bmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{bmatrix} \right) \text{►Sphere} \\ \left[\sqrt{13} \quad \angle \frac{\pi}{4} \quad \angle \sin^{-1} \left(\frac{2 \cdot \sqrt{13}}{13} \right) \right]$$



sqrt()

sqrt(*Lausl*)⇒*lauseke*

$\sqrt{4}$	2
------------	---

sqrt(*Listal*)⇒*lista*

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

Laskee argumentin neliöjuuren.

Kun kyseessä on lista, laskee kaikkien *Listal*:n elementtien neliöjuuret.

Huomaa: Katso myös **Neliöjuurimalli**, sivu 1.

stat.results

stat.results

$xlist:=\{1,2,3,4,5\}$	$\{1,2,3,4,5\}$
------------------------	-----------------

Näyttää tilastollisen laskutoimituksen tulokset.

$ylist:=\{4,8,11,14,17\}$	$\{4,8,11,14,17\}$
---------------------------	--------------------

Vastaukset näytetään nimiarvoparien sarjana. Näytetyt nimenomaiset nimet riippuvat viimeksi sievennetystä tilastofunktiosta tai komennosta.

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
"t"	0.998053
"Resid"	" {... } "

Voit kopioida nimen tai arvon ja liittää sen muihin paikkoihin.

<i>stat.values</i>	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	" {-0.4,0.4,0.2,0,-0.2} "

Huomaa: Älä määritä muuttujia, joilla on sama nimi kuin tilastoanalyseissä käytettävillä muuttujilla. Joissakin tapauksissa tästä voi olla seurauksena virhetilanne. Tilastoanalyseissä käytettävät muuttujanimet on esitetty alla olevassa taulukossa.

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.SSEError
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

Katso esimerkki kohdassa
stat.results.

Näyttää matriisiin 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 272.

stDevPop(MatriisiI[, frekvMatriisi]) ⇒ matriisi

Laskee rivivektorin *MatriisiI*:n sarakkeiden perusjoukon keskihajonnoista.

Jokainen *frekvMatriisin* elementti näyttää *MatriisiI*:n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *MatriisiI*:ssä on oltava vähintään kaksi riviä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjästä elementeistä, katso sivu 272.

Radiaanikulma- ja automaattisissa tiloissa:

$$\begin{aligned} \text{stDevPop}(\{a, b, c\}) &= \frac{\sqrt{2 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3} \\ \text{stDevPop}(\{1, 2, 5, -6, 3, -2\}) &= \frac{\sqrt{465}}{6} \\ \text{stDevPop}(\{1, 3, 2, 5, -6, 4\}, \{3, 2, 5\}) &= 4.11107 \end{aligned}$$

$$\begin{aligned} \text{stDevPop} \left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix} \right) &= \begin{bmatrix} \frac{4 \cdot \sqrt{6}}{3} & \frac{\sqrt{78}}{3} & \frac{2 \cdot \sqrt{6}}{3} \end{bmatrix} \\ \text{stDevPop} \left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix} \right) &= \begin{bmatrix} 2.52608 & 5.21506 \end{bmatrix} \end{aligned}$$

stDevSamp()

Katalogi > 

stDevSamp(*List*[,
frekvList])⇒*lauseke*

Laskee *List*an sisältämien elementtien otoksen keskihajonnan.

Jokainen *frekvList*an elementti näyttää *List*an vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *List*assa tulee olla vähintään kaksi elementtiä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjästä elementeistä, katso sivu 272.

stDevSamp(*Matriisi1*[,
frekvMatriisi1])⇒*matriisi*

Laskee rivivektorin *Matriisi1*:n sarakkeiden otosten keskihajonnoista.

Jokainen *frekvMatriisi1*in elementti näyttää *Matriisi1*:n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Matriisi1* :ssä on oltava vähintään kaksi riviä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjästä elementeistä, katso sivu 272.

$$\text{stDevSamp}(\{a, b, c\}) = \frac{\sqrt{3 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3}$$

$$\text{stDevSamp}(\{1, 2, 5, -6, 3, -2\}) = \frac{\sqrt{62}}{2}$$

$$\text{stDevSamp}(\{1.3, 2.5, -6.4\}, \{3, 2, 5\}) = 4.33345$$

$$\text{stDevSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right) = \begin{bmatrix} 4 & \sqrt{13} & 2 \end{bmatrix}$$

$$\text{stDevSamp}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right) = \begin{bmatrix} 2.7005 & 5.44695 \end{bmatrix}$$

Stop

Katalogi > 

Stop

Ohjelmointikomento: Pysäyttää ohjelman.

Stop ei ole sallittu funktioissa.

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

<i>i</i> :=0	0
Define <i>prog1</i> ()=Prgm	Done
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1</i> ()	Done
<i>i</i>	5

Store

Katso → (tallenna), sivu 253.

string() (merkkijono)	Katalogi > 	
string (<i>Laus</i>) $\Rightarrow$ <i>merkkijono</i>	<code>string(1.2345)</code>	"1.2345"
Sieventää lausekkeen <i>Laus</i> ja antaa vastauksen merkkijonona.	<code>string(1+2)</code>	"3"
	<code>string(cos(x)+$\sqrt{3}$)</code>	"cos(x)+ $\sqrt{3}$ "

subMat()	Katalogi > 	
subMat (<i>MatriisiI</i> [, <i>alkurivi</i>] [, <i>alkusarake</i>] [, <i>loppurivi</i>] [, <i>loppusarake</i>]) $\Rightarrow$ <i>matriisi</i>	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
Laskee <i>MatriisiI</i> :n määritetyn alimatriisin.	<code>subMat(m1,2,1,3,2)</code>	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
Oletusarvot: <i>alkurivi</i> =1, <i>alkusarake</i> =1, <i>loppurivi</i> =viimeinen rivi, <i>endCol</i> =viimeinen sarake.	<code>subMat(m1,2,2)</code>	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)	Katso $\Sigma()$, sivu 243.
--------------------	------------------------------

sum()	Katalogi > 	
sum (<i>Lista</i> [, <i>Alku</i>] [, <i>Loppu</i>]) $\Rightarrow$ <i>lauseke</i>	<code>sum({1,2,3,4,5})</code>	15
Laskee <i>Listan</i> elementtien summan.	<code>sum({a,2*a,3*a})</code>	$6 \cdot a$
<i>Alku</i> ja <i>Loppu</i> ovat valinnaisia. Ne määrittävät elementtien alueen.	<code>sum(seq(n,n,1,10))</code>	55
	<code>sum({1,3,5,7,9},3)</code>	21
Mikä tahansa tyhjä argumentti tuottaa tyhjän vastauksen. <i>Listassa</i> olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 272.	$\text{sum} \left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \right)$	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
sum (<i>MatriisiI</i> [, <i>Alku</i>] [, <i>Loppu</i>]) $\Rightarrow$ <i>matriisi</i>	$\text{sum} \left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \right)$	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
Laskee rivivektorin, joka sisältää <i>MatriisiI</i> :n sarakkeiden elementtien summat.	$\text{sum} \left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, 2, 3 \right)$	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$
<i>Alku</i> ja <i>Loppu</i> ovat valinnaisia. Ne määrittävät rivialueen.		

Mikä tahansa tyhjä argumentti tuottaa tyhjän vastauksen. *Matriisi1*:ssä olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 272.

sumIf()

**sumIf(Lista,Kriteerit[,
SummaLista])**⇒arvo

sumIf($\{ \{ 1,2,e,3,\pi,4,5,6 \}, 2.5 < ? < 4.5 \}$)

$e + \pi + 7$

Laskee kaikkien niiden *Listan* sisältämien elementtien kumuloituneen summan, jotka vastaavat määritettyjä kriteereitä *Kriteerit*. Voit halutessasi antaa kumuloivat elementit määrittämällä vaihtoehtoisen listan, *summaLista*.

sumIf($\{ \{ 1,2,3,4 \}, 2 < ? < 5, \{ 10,20,30,40 \} \}$)

70

Lista 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 niitä *Listan* elementtejä, jotka ovat alle 10.

Kun jokin *Listan* elementti vastaa kriteereitä *Kriteerit*, elementti lisätään kumuloituvaan summaan. Jos sisällytät funktion *summaListan*, summaan lisätäänkin sen sijaan vastaava *summaListan* elementti.

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

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 272.

sumlf()

Katalogi >

Huomaa: Katso myös **countlf()**, sivu 37.

sumSeq()Katso $\Sigma()$, sivu 243.**system()**

Katalogi >

system(*Yht1* [, *Yht2* [, *Yht3* [, ...]]])

$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right) \quad x=4 \text{ and } y=-4$$

system(*Laus1* [, *Laus2* [, *Laus3* [, ...]]])

Laskee yhtälöryhmän listaksi muotoiltuna. Voit luoda yhtälöryhmän myös mallin avulla.

Huomaa: Katso myös **Yhtälöryhmä**, sivu 4.

T**T (transponoi)**

Katalogi >

*Matriisi*1 $\mathbf{T} \Rightarrow$ *matriisi*

Laskee *Matriisi*1:n transponoidun liittokompleksimatriisin.

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla @t.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T \quad \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T \quad \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

$$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^T \quad \begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$$

tan()

-painike

tan(*Laus1*) $\Rightarrow$ *lauseke*

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

tan(*Laus1*) laskee argumentin tangentin lausekkeena.

tan(*Lista1*) määrittää *Lista1*:n kaikkien elementtien tangenttien listan.

Astekulmatilassa:

$$\tan\left(\frac{\pi}{4}\right) \quad 1$$

$$\tan(45) \quad 1$$

$$\tan(\{0,60,90\}) \quad \{0,\sqrt{3},\text{undef}\}$$

Graadikulmatilassa:

tan() -painike

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

$\tan\left(\frac{\pi}{4}\right)$	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(neliömatriisiI)⇒neliömatriisi

Laskee *neliömatriisiI*:n matriisin tangentin. Tämä ei ole sama kuin kunkin elementin tangentin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisiI:n on oltava diagonaloitavissa. 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(LausI)⇒lauseke

tan⁻¹(ListI)⇒lista

tan⁻¹(LausI) laskee kulman, jonka tangentti on *LausI*, määrittäen vastauksen lausekkeena.

tan⁻¹(ListI) luo listan *ListI*:n jokaisen elementin käänteistangenteista.

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

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

tan⁻¹(neliömatriisiI)⇒neliömatriisi

Astekulmatilassa:

$\tan^{-1}(1)$	45
----------------	----

Graadikulmatilassa:

$\tan^{-1}(1)$	50
----------------	----

Radiaanikulmatilassa:

$\tan^{-1}(\{0,0.2,0.5\})$	$\{0,0.197396,0.463648\}$
----------------------------	---------------------------

Radiaanikulmatilassa:

$\tan^{-1}()$

 -painike

Laskee *neliömatriisi1*:n matriisin käänteistangentin. Tämä ei ole sama kuin kunkin elementin käänteisen tangentin laskeminen.

Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisi1:n on oltava diagonalisoitavissa. Vastaus sisältää aina liukulukuja.

$$\tan^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

tangentLine()

Katalogi > 

tangentLine

(Laus1,Muutt,Piste)⇒lauseke

tangentLine

(Laus1,Muutt=Piste)⇒lauseke

Määrittää tangenttisuoran *Laus1*: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**($f_1(x),x,2$) 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(Laus1)⇒lauseke

tanh(Lista1)⇒lista

tanh(Laus1) laskee argumentin hyperbolisen tangentin lausekkeena.

tanh(Lista1) luo listan *Lista1*:n jokaisen elementin hyperbolisista tangenteista.

tanh(neliömatriisi1)⇒neliömatriisi

Laskee *neliömatriisi1*:n matriisin hyperbolisen tangentin. Tämä ei ole sama kuin kunkin elementin hyperbolisen tangentin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

$\tanh(1.2)$	0.833655
$\tanh(\{0,1\})$	$\{0,\tanh(1)\}$

Radiaanikulmatilassa:

$$\tanh\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

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

tanh⁻¹()

tanh⁻¹(*Laus l*) ⇒ *lauseke*

Suorakulmakompleksimuodossa:

tanh⁻¹(*Listal*) ⇒ *lista*

tanh ⁻¹ (0)	0
------------------------	---

tanh⁻¹(*Laus l*) laskee argumentin käänteisen hyperbolisen tangentin lausekkeena.

tanh ⁻¹ ({ 1,2,1,3 })
{ undef, 0.518046−1.5708⋅ <i>i</i> , $\frac{\ln(2)}{2}$ − $\frac{\pi}{2}$ ⋅ <i>i</i> }

tanh⁻¹(*Listal*) luo listan *Listal*:n jokaisen elementin käänteisistä hyperbolisista tangenteista.

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

tanh⁻¹(*neliömatrissi l*) ⇒ *neliömatrissi*

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

tanh ⁻¹ ($\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$)
$\begin{bmatrix} -0.099353+0.164058\cdot i & 0.267834-1.4908 \\ -0.087596-0.725533\cdot i & 0.479679-0.94730 \\ 0.511463-2.08316\cdot i & -0.878563+1.7901 \end{bmatrix}$

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

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

taylor()Katalogi > **taylor**(*Laus*1, *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 typistettyjä 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 lavennuspiste.

$$\begin{array}{l} \text{taylor}\left(e^{\sqrt{x}}, x, 2\right) \qquad \text{taylor}\left(e^{\sqrt{x}}, x, 2, 0\right) \\ \text{taylor}\left(e^t, t, 4\right) | t = \sqrt{x} \qquad \frac{3}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1 \\ \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right) \qquad \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right) \\ \text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}, x\right) \\ -x^3 - x^2 - x - \frac{1}{x} - 1 \end{array}$$

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 määritetylle vapausasteelle *df*.

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

tCollect()Katalogi > **tCollect**(*Laus*1) $\Rightarrow$ lauseke

Laskee lausekkeen, jossa tulot ja sinien ja kosinien kokonaislukupotenssit on muunnettu useiden kulmien sinien ja kosinien lineaariseksi kombinaatioksi, kulmien summiksi ja kulmien erotuksiksi. Transformaatio muuntaa trigonometriset polynomit niiden harmonisten lineaariseksi kombinaatioksi.

$$\begin{array}{l} \text{tCollect}\left((\cos(\alpha))^2\right) \qquad \frac{\cos(2 \cdot \alpha) + 1}{2} \\ \text{tCollect}\left(\sin(\alpha) \cdot \cos(\beta)\right) \qquad \frac{\sin(\alpha - \beta) + \sin(\alpha + \beta)}{2} \end{array}$$

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(Lausl)⇒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()**-funktioita tulee käyttää radiaanikulmatilassa.

$tExpand(\sin(3 \cdot \phi))$	$4 \cdot \sin(\phi) \cdot \{\cos(\phi)\}^2 - \sin(\phi)$
$tExpand(\cos(\alpha - \beta))$	$\cos(\alpha) \cdot \cos(\beta) + \sin(\alpha) \cdot \sin(\beta)$

Text*kehotemerkkijono[, Näytälippu]*

Ohjelmointikomento: Keskeyttää ohjelman ja näyttää merkkijonon *kehotemerkkijono* valintaruudussa.

Kun käyttäjä valitsee **OK**-näppäimen, ohjelman suoritus jatkuu.

Valinnainen *lippu*-argumentti voi olla mikä tahansa lauseke.

- Jos *NäytäLippu* jätetään pois, tai jos se sieventyy arvoksi **1**, tekstimuotoinen viesti lisätään laskimen historiaan.
- Jos *NäytäLippu* sieventyy arvoon **0**, tekstimuotoista viestiä ei lisätä historiaan.

Jos ohjelma vaatii käyttäjän kirjoittaman vastauksen, katso **Request, sivu 161, tai RequestStr, sivu 163**.

Huomaa: Tätä komentoa voi käyttää käyttäjän määrittämän ohjelman sisällä mutta ei funktion sisällä.

Määritä ohjelma, joka keskeytyy ja näyttää kunkin viidestä satunnaisluvusta valintaruudussa.

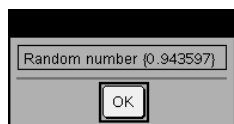
Paina mallin Prgm...EndPrgm jokaisen rivin lopussa näppäintä  näppäimen  sijaan. Tietokoneen näppäimistöllä **Alt**-näppäintä pidetään alhaalla ja painetaan **Enter**.

```
Define text_demo()=Prgm
  For i,1,5
    strinfo:="Random number " &
    string(rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Suorita ohjelma:

```
text_demo()
```

Esimerkki yhdestä valintaruudusta:



TInterval *Lista[,Frekv[,CTaso]]*

(Datalistan syöte)

TInterval $\bar{x}, Sx, n[,CTaso]$

(Yhteenvetotilaston syöte)

Laskee t -luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Lisätietoja listassa olevien tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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 *Lista1,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 191.)

Poolaus=1 poolaa varianssit; *Poolaus=0* ei poolaa variansseja.

Lisätietoja listassa olevien tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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.df	Vapausasteet
stat. $\bar{x}_1$, stat. $\bar{x}_2$	Satunnaisesta normaalijakaumasta olevien datasekvenssien otosten keskiarvot
stat. σ_1 , stat. σ_2	<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* °lämpYsikkö, _
°lämpYsikkö2)
⇒*lauseke* _°lämpYsikkö2

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* _°lämpYsikkö, _

Merkin Δ saat Katalogin symboleista.

Δ tmpCnv()

Katalogi >  $^{\circ}$ lämpYksikkö2) $\Rightarrow$ lauseke $_{^{\circ}$ lämpYksikkö2

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

Muuttaa *Laus*:n määrittämän lämpötila-alueen (kahden lämpötila-arvon välisen eron) yksiköstä toiseen. Kelpaavat lämpötilayksiköt ovat:

$_{^{\circ}$ CCelsius
 $_{^{\circ}$ FFahrenheit
 $_{^{\circ}$ KKelvin
 $_{^{\circ}$ RRankine

Voit syöttää merkin $^{\circ}$ valitsemalla sen symbolipaletista tai kirjoittamalla @d.

alaviivan $_$ voi syöttää painamalla

 .

$1_{^{\circ}$ C ja $1_{^{\circ}$ K ovat samansuuruisia, samoin kuin $1_{^{\circ}$ F ja $1_{^{\circ}$ R. $1_{^{\circ}$ C on kuitenkin 9/5 Fahrenheit-asteen arvosta $1_{^{\circ}$ F.

Esimerkiksi $100_{^{\circ}$ C:n alue ($0_{^{\circ}$ C - $100_{^{\circ}$ C) vastaa Fahrenheit-asteiden aluetta $180_{^{\circ}$ F.

Jos haluat muuttaa jonkin lämpötila-arvon alueen sijaan, käytä funktiota **tmpCnv()**.

Δ tmpCnv($100_{^{\circ}$ C, $_{^{\circ}$ F)	$180_{^{\circ}$ F
Δ tmpCnv($180_{^{\circ}$ F, $_{^{\circ}$ C)	$100_{^{\circ}$ C
Δ tmpCnv($100_{^{\circ}$ C, $_{^{\circ}$ K)	$100_{^{\circ}$ K
Δ tmpCnv($100_{^{\circ}$ F, $_{^{\circ}$ R)	$100_{^{\circ}$ R
Δ tmpCnv($1_{^{\circ}$ C, $_{^{\circ}$ F)	$1.8_{^{\circ}$ F

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(neliömatriisi)⇒lauseke

Laskee *neliömatriisin* jäljityksen (päälävistäjän kaikkien elementtien summan).

trace	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	15
trace	$\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}$	$2 \cdot a$

Try (Kokeile)

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 282.

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 komentojen **Try**, **ClrErr** ja **PassErr** toiminnan, syötä oikealla näkyvä *eigenvals()*-ohjelma. Suorita ohjelma suorittamalla kukin seuraavista lausekkeista.

<i>eigenvals</i>	$\begin{pmatrix} -3 \\ -41 \\ 5 \end{pmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}$
<i>eigenvals</i>	$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Define *progI()*=Prgm

Try

z:=z+1

Disp "z incremented."

Else

Disp "Sorry, z undefined."

EndTry

EndPrgm

Done

z:=1:*progI()*

z incremented.

Done

DelVar z:*progI()*

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

Huomaa: Katso myös **ClrErr**, sivu 27, ja **PassErr**, sivu 142.

Disp "Error: Product of A-B must be a square matrix"

ClrErr

Else

PassErr

EndIf

EndTry

EndPrm

tTest

tTest μ_0 ,*Lista*[,*Frekv*[,*Hypot*]]

(Datalistan syöte)

tTest μ_0 , $\bar{x}$,*sx*,*n*[,*Hypot*]

(Yhteenvetotilaston syöte)

Testaa hypoteesia yhden tuntemattoman perusjoukon keskiarvoon μ , kun perusjoukon keskihajontaa σ ei tunneta. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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 tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.t	$(\bar{x} - \mu_0) / (\text{stdev} / \text{sqrt}(n))$
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat.df	Vapausasteet

Tulosmuuttuja	Kuvaus
stat. $\bar{x}$	Listan sisältämän datasekvenssin otoksen keskiarvo
stat.sx	Datasekvenssin otoksen keskihajonta
stat.n	Otoksen koko

tTest_2Samp

Katalogi > 

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 191.)

Testaa $H_0: \mu_1 = \mu_2$, jonkin seuraavan vaihtoehdon suhteen:

Kun $H_a: \mu_1 < \mu_2$, 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 tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.t	Keskiarvojen erotukselle laskettu vakio-ohjearvo
stat.PVal	Alin merkitsevyystaso, 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()Katalogi > 

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.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 211. 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 211. 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 211. 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 211. Katso myös **amortTbl()**, sivu 8.

tvmPV()

tvmPV($N, I, Pmt, FV, [PpY], [CpY], [PmtAt]$) $\Rightarrow$ arvo

tvmPV(48, 4, -500, 30000, 12, 12) -3426.7

Talouselaskentafunktio, joka laskee nykyarvon.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 211. 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 talouselaskentatoiminnossa. Talouselaskentafunktioiden argumenttien arvot tai vastaukset eivät kuitenkaan tallennu TVM-muuttujiin.

TwoVar

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

Laskee kahden muuttujan tilastot. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

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.

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 272.

Tulosmuuttuja	Kuvaus
stat. $\bar{x}$	$x:n$ arvojen keskiarvo
stat. x	$x:n$ arvojen summa
stat. x^2	$x^2:n$ 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	$y^2:n$ arvojen summa

Tulosmuuttuja	Kuvaus
stat.sy	y:n otoksen keskihajonta
stat.σy	y:n perusjoukon keskihajonta
stat.Σxy	x · y -arvojen summa
stat.r	Korrelaatiokerroin
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 3. neljännes
stat.MaxX	x:n arvojen maksimi
stat.MinY	y:n arvojen minimi
stat.Q ₁ Y	y:n ensimmäinen neljännes
stat.MedY	y:n mediaani
stat.Q ₃ Y	x:n kolmas neljännes
stat.MaxY	y:n arvojen maksimi
stat.Σ(x- $\bar{x}$) ²	x:n keskiarvon poikkeamien neliöiden summa
stat.Σ(y- $\bar{y}$) ²	y:n keskiarvon poikkeamien neliöiden summa

U

unitV()

Katalogi >

yksikköV(VektoriI)⇒vektori

Laskee joko rivi- tai sarakkeyksikkövektorin riippuen VektoriI:n muodosta.

VektoriI :n on oltava joko yksirivinen matriisi tai yksisarakkeinen matriisi.

$$\text{unitV}\left(\begin{bmatrix} a & b & c \end{bmatrix}\right) = \begin{bmatrix} \frac{a}{\sqrt{a^2+b^2+c^2}} & \frac{b}{\sqrt{a^2+b^2+c^2}} & \frac{c}{\sqrt{a^2+b^2+c^2}} \end{bmatrix}$$

$$\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{3} & \frac{\sqrt{6}}{6} \end{bmatrix}$$

$$\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{14}}{14} \\ \frac{\sqrt{14}}{7} \\ \frac{3\sqrt{14}}{14} \end{bmatrix}$$

Luettelo aakkosjärjestyksessä 213

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

unLock

unLock*Muutt1* [, *Muutt2*] [, *Muutt3*] ...
unLock*Muutt*.

Vapauttaa määritetyt muuttujat tai muuttujaryhmän. Lukittuja muuttujia ei voi muokata eikä poistaa.

Katso **Lock**, sivu 114, ja **getLockInfo()**, sivu 89.

<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()

varPop(*List*a [, *frekvList*a])⇒*lauseke*

Laskee *List*a:n perusjoukon varianssin.

Jokainen *frekvList*a:n elementti näyttää *List*a:n 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 272.

varPop({ { 5,10,15,20,25,30 } })	875
	12
Ans•1.	72.9167

varSamp(Lista[, frekvLista]) ⇒ lausekeLaskee *Listan* otosten varianssin.

Jokainen *frekvListan* elementti näyttää *Listan* vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Listassa* 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 272.

varSamp(Matriisi1[, frekvMatriisi]) ⇒ matriisi

Laskee rivivektorin, joka sisältää *Matriisi1*:n kaikkien sarakkeiden otoksen varianssin.

Jokainen *frekvMatriisin* elementti näyttää *Matriisi1*: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 tyhjästä elementeistä, katso sivu 272.

Huomaa: *Matriisi1*:ssä on oltava vähintään kaksi elementtiä.

varSamp({a,b,c})

$$\frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$$

varSamp({1,2,5,-6,3,-2}) 31
2varSamp({1,3,5},{4,6,2}) 68
33varSamp($\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}$) [4.75 1.03 4]varSamp($\begin{bmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{bmatrix}, \begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}$) [3.91731 2.08411]

W

Wait aikaSekunteina

Keskeyttää toiminnon *aikaSekunteina* sekunnin ajaksi.

Odottaaksesi 4 sekuntia:

Wait 4

Odottaaksesi 1/2 sekuntia:

Wait 0.5

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 -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.

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.

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 290.



$$\text{warnCodes}\left(\text{solve}\left(\sin(10 \cdot x) = \frac{x^2}{x}, x\right), \text{warn}\right)$$

$$x = -0.84232 \text{ or } x = -0.706817 \text{ or } x = -0.2852$$

warn	{ 10007, 10009 }
------	------------------

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

when()

Katalogi > 

when(*Ehto*, *tosiTulos* [, *epätosiTulos*][, *tuntematonTulos*])
⇒ *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.

Jätä pois sekä *epätosiTulos* että *tuntematonTulos*, kun haluat, että lauseke määritetään vain alueella, jolla *Ehto* on tosi.

Käytä komentoa **undef** *epätosiTulos*, kun haluat määrittää lausekkeen, joka piirtyy vain jollekin välille.

when() on hyödyllinen komento rekursiivisten funktioiden määrittämisessä.

$\text{when}(x < 0, x + 3)$	$x = 5$	undef
-----------------------------	---------	-------

$\text{when}(n > 0, n \cdot \text{factorial}(n - 1), 1)$	$\rightarrow \text{factorial}(n)$	Done
--	-----------------------------------	------

$\text{factorial}(3)$	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*) = Func
Local *i*, *tempsum*
 $1 \rightarrow i$
 $0 \rightarrow \text{tempsum}$
While $i \leq n$
 $\text{tempsum} + \frac{1}{i} \rightarrow \text{tempsum}$
 $i + 1 \rightarrow i$
EndWhile
Return *tempsum*
EndFunc

$\text{sum_of_recip}(3)$	11	6
----------------------------	----	---

X

BooleanLaus1 **xor** *BooleanLaus2* antaa vastauksena *Boolean* lausekkeen *BooleanList1*

xor *BooleanList2* antaa vastauksena *Boolean* listan *BooleanMatriisi1*

xor *BooleanMatriisi2* antaa vastauksena *Boolean* matriisin

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 kummarkaan argumentin totuusarvoa ei voi määrittää todeksi tai epätodeksi.

Huomaa: Katso **or**, **täältä**.

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).

true xor true	false
5>3 xor 3>5	true

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.

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**, **täältä**.

Z

zeros()

zeros(Laus, Muutt)⇒lista

zeros(Laus, Muutt=Arvaus)⇒lista

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:

$$\text{zeros}(a \cdot x^2 + b \cdot x + c, x) \\ \left\{ \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}, \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a} \right\} \\ a \cdot x^2 + b \cdot x + c | x = \text{Ans}[2] \quad 0$$

$$\text{exact}\left(\text{zeros}\left(a \cdot \left(e^x + x\right) \cdot \left(\text{sign}(x) - 1\right), x\right)\right) \quad \left\{ \begin{matrix} \square \end{matrix} \right\} \\ \text{exact}\left(\text{solve}\left(a \cdot \left(e^x + x\right) \cdot \left(\text{sign}(x) - 1\right) = 0, x\right)\right) \\ e^x + x = 0 \text{ or } x > 0 \text{ or } a = 0$$

muuttuja

– tai –

*muuttuja = reaali tai ei-reaali luku*Esimerkiksi x kelpaa ja samoin $x=3$.

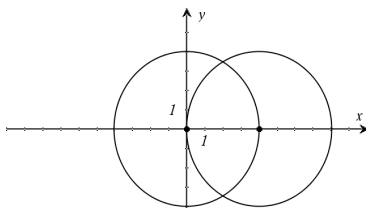
Jos kaikki lausekkeet ovat polynomeja, ja ET määrittää ensimmäisiä arvauksia, **zeros()** käyttää leksikaalista Gröbner/Buchbergerin eliminaatiomenetelmää yrittäessään määrittää kaikki reaaliset nollakohdat.

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.

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 kokonaisluku väliltä 1-255.



$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x, y\}\right)$$

$$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} \\ \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \end{bmatrix}$$

Määritä rivin 2 juuri:

$$\text{Ans}[2] \quad \begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \end{bmatrix}$$

$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x, y, z\}\right)$$

$$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} & c1 \\ \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} & c1 \end{bmatrix}$$

Polynomisarjoissa laskutoimituksen suoritus aika tai muistin käyttö voivat riippua merkittävästi tuntemattomien muuttujien järjestyksestä. Jos ensimmäinen valintasi kuluttaa muistia, tai et jaksa 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.

Jos yhtälöryhmä ei ole polynominen kaikilta muuttujiltaan eikä lineaarinen tuntemattomilta muuttujiltaan, **zeros()** määrittää korkeintaan yhden nollakohdan käyttäen likimääristä 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.

$$\text{zeros}\left(\left\{x + e^z \cdot y - 1, x - y - \sin(z)\right\}, \{x, y\}\right)$$

$$\left[\begin{array}{cc} \frac{e^z \cdot \sin(z) + 1}{e^z + 1} & \frac{-\sin(z) - 1}{e^z + 1} \end{array} \right]$$

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \{y, z\}\right)$$

$$\left[\begin{array}{cc} 0.041458 & 3.18306 \\ 0.001871 & 6.28131 \\ 4.76\text{E-}11 & 1796.99 \\ 2.\text{E-}13 & 254.469 \end{array} \right]$$

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \{y, z = 2 \cdot \pi\}\right)$$

$$\left[0.001871 \quad 6.28131 \right]$$

zInterval

zInterval σ , *Lista*, *Frekv*, *CTaso*]]

(Datalistan syöte)

zInterval σ , $\bar{x}$, *n* [, *CTaso*]

(Yhteenvetotilaston syöte)

Laskee z -luottamusvälin. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Lisätietoja listassa olevien tyhjiä elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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
stat.n	Otoksen keskiarvon sisältävän datasekvenssin pituus
stat. σ	Datasekvenssin <i>Lista</i> tunnettu perusjoukon keskihajonta

zInterval_1Prop

zInterval_1Prop x, n [,CTaso]

Laskee yhden osuuden z -luottamusvälin. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

x on ei-negatiivinen kokonaisluku.

Lisätietoja listassa olevien tyhjiä elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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 $x1, n1, x2, n2[, CTaso]$

Laskee kahden osuuden z -luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

$x1$ ja $x2$ ovat ei-negatiivisia kokonaislukuja.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Luottamusväli, joka sisältää jakauman luottamusvälin todennäköisyyden
stat. $\hat{p}$ Diff	Osuuksien välinen laskettu erotus
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 $\sigma_1, \sigma_2, Lista1, Lista2$
 $[, Frekv1[, Frekv2[, CTaso]]]$

(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 yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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. $\sigma x2$	Listat 1:n ja Listat 2:n otosten keskihajonnat
stat.n1, stat.n2	Otosten lukumäärä datasekvensseissä
stat.r1, stat.r2	Datasekvenssien Listat 1 ja Listat 2 tunnetut perusjoukon keskihajonnat

zTest

Katalogi > 

zTest $\mu0, \sigma, Lista, [Frekv[, Hypot]]$

(Datalistan syöte)

zTest $\mu0, \sigma, \bar{x}, n[, Hypot]$

(Yhteenvetotilaston syöte)

Suorittaa z-testin frekvenssillä *frekvlista*. Tulosten yhteenvedo tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

Testaa $H_0: \mu = \mu0$, jonkin seuraavan vaihtoehdon suhteen:

Kun $H_a: \mu < \mu0$, aseta *Hypot*<0

Kun $H_a: \mu \neq \mu0$ (oletus), aseta *Hypot*=0

Kun $H_a: \mu > \mu0$, aseta *Hypot*>0

Lisätietoja listassa olevien tyhjen elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.z	$(\bar{x} - \mu0) / (\sigma / \sqrt{n})$
stat.P Value	Pienin todennäköisyys, jolla nollassuhteeksi voidaan hylätä
stat. $\bar{x}$	Listan sisältämän datasekvenssin otoksen keskiarvo
stat.sx	Datasekvenssin otoksen keskihajonta. Lasketaan vain Data-syötteelle.

Tulosmuuttuja	Kuvaus
stat.n	Otoksen koko

zTest_1Prop

Katalogi > 

zTest_1Prop $p0, x, n[, Hypot]$

Laskee yhden osuuden z -testin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

x on ei-negatiivinen kokonaisluku.

Testaa $H_0: p = p0$ jonkin seuraavan vaihtoehdon suhteen:

Kun $H_a: p > p0$, aseta *Hypot*>0

Kun $H_a: p \neq p0$ (oletus), aseta *Hypot*=0

Kun $H_a: p < p0$, aseta *Hypot*<0

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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

Katalogi > 

zTest_2Prop $x1, n1, x2, n2[, Hypot]$

Laskee kahden osuuden z -testin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 191.)

$x1$ ja $x2$ ovat ei-negatiivisia kokonaislukuja.

Testaa $H_0: p1 = p2$ jonkin seuraavan vaihtoehdon suhteen:

Kun $H_a: p1 > p2$, aseta *Hypot*>0

Kun $H_a: p1 \neq p2$ (oletus), aseta *Hypot*=0

Kun $H_a: p < p0$, aseta *Hypot*<0

Lisätietoja listassa olevien tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

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 191.)

Testaa $H_0: \mu1 = \mu2$, jonkin seuraavan vaihtoehdon suhteen:

Kun $H_a: \mu1 < \mu2$, aseta *Hypot*<0

Kun $H_a: \mu1 \neq \mu2$ (oletus), aseta *Hypot*=0

Kun $H_a: \mu1 > \mu2$, *Hypot*>0

Lisätietoja listassa olevien tyhjiin elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 272.

Tulosmuuttuja	Kuvaus
stat.z	Keskiarvojen erotukselle laskettu vakio-ohjearvo
stat.PVal	Alin merkitsevyystaso, jolla nollasshypoteesi 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
<i>Laus1 + Laus2⇒lauseke</i>	56	56
Laskee kahden argumentin summan.	56+4	60
	60+4	64
	64+4	68
	68+4	72
<i>List1 + List2⇒listaMatriisi1 + Matriisi2⇒matriisi</i>	$\left\{22,\pi,\frac{\pi}{2}\right\}\rightarrow I1$	$\left\{22,\pi,\frac{\pi}{2}\right\}$
Määrittää listan (tai matriisin), joka sisältää <i>List1</i> :n ja <i>List2</i> :n (tai <i>Matriisi1</i> :n ja <i>Matriisi2</i> :n) vastaavien elementtien summat.	$\left\{10,5,\frac{\pi}{2}\right\}\rightarrow I2$	$\left\{10,5,\frac{\pi}{2}\right\}$
	$I1+I2$	$\{32,\pi+5,\pi\}$
	$Ans+\{\pi,-5,\pi\}$	$\{\pi+32,\pi,0\}$
Argumenttien tulee olla mitoiltaan samanlaisia.	$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$
<i>Laus + List1⇒lista</i>	$15+\{10,15,20\}$	$\{25,30,35\}$
<i>List1 + Laus⇒lista</i>	$\{10,15,20\}+15$	$\{25,30,35\}$
Määrittää listan, joka sisältää <i>Laus</i> :n ja <i>List1</i> :n kunkin elementin summat.		
<i>Laus + Matriisi1⇒matriisi</i>	$20+\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
<i>Matriisi1 + Laus⇒matriisi</i>		
Laskee matriisin, jossa <i>Laus</i> on lisätty jokaiseen elementtiin <i>Matriisi1</i> :n diagonaalimatriisissa. <i>Matriisi1</i> :n on oltava neliö.		
Huomaa: Käytä merkintää .+ (piste plus), kun haluat lisätä lausekkeen jokaiseen elementtiin.		

-(vähennyslasku)		- painike
<i>Laus1 - Laus2⇒lauseke</i>	6-2	4
Laskee <i>Laus1</i> miinus <i>Laus2</i> .	$\pi-\frac{\pi}{6}$	$\frac{5\cdot\pi}{6}$

–(vähennyslasku)

 painike

$Listal - Lista2 \Rightarrow lista$

$$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\} = \{ 12, \pi - 5, 0 \}$$

$Matriisi1 - Matriisi2 \Rightarrow matriisi$

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

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

$Laus1 \cdot Laus2 \Rightarrow lauseke$

$$2 \cdot 3.45 = 6.9$$

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 $Listal$: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 · *Lista1* ⇒ *lista*

Lista1 · *Laus* ⇒ *lista*

Määrittää listan, joka sisältää lausekkeen *Laus* ja kunkin *Lista1*:n elementin tulon.

Laus · *Matriisi1* ⇒ *matriisi*
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.

Lista1 / *Lista2* ⇒ *lista*

Määrittää listan, joka sisältää osamäärät laskutoimituksista *Lista1* jaettuna *Lista2*:lla.

Listojen tulee olla mitoiltaan samanlaisia.

Laus / *Lista1* ⇒ *lista*

Lista1 / *Laus* ⇒ *lista*

$$\frac{2}{3.45} = 0.57971$$

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

$$\frac{\{1, 2, 3\}}{\{4, 5, 6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

$$\frac{a}{\{3, a, \sqrt{a}\}} = \left\{\frac{a}{3}, 1, \sqrt{a}\right\}$$

$$\frac{\{a, b, c\}}{a \cdot b \cdot c} = \left\{\frac{1}{b \cdot c}, \frac{1}{a \cdot c}, \frac{1}{a \cdot b}\right\}$$

/ (jakolasku)

 painike

Määrittää listan, joka sisältää osamäärät laskutoimituksista *Laus* jaettuna *Lista1*:llä tai *Lista1* jaettuna lausekkeella *Laus*.

Matriisi1 / *Laus* $\Rightarrow$ *matriisi*

$$\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{a \cdot b \cdot c} \quad \begin{bmatrix} \frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b} \end{bmatrix}$$

Laskee matriisiin, joka sisältää osamäärät laskutoimituksesta *Matriisi1*/*Laus*.

Huomaa: Käytä merkintää . / (piste jaa), kun haluat jakaa lausekkeen jokaisella elementillä.

^ (potenssi)

 painike

Laus1 ^ *Laus2* $\Rightarrow$ lauseke

$$4^2 \quad 16$$

Lista1 ^ *Lista2* $\Rightarrow$ lista

$$\{a, 2, c\}^{\{1, b, 3\}} \quad \{a, 2^b, c^3\}$$

Laskee ensimmäisen argumentin korotettuna toisen argumentin potenssiin.

Huomaa: Katso myös **Eksponenttimalli**, sivu 1.

Jos kyseessä on lista, laskee *Lista1*:n elementit korotettuna *Lista2*:n vastaavien elementtien potenssiin.

Reaalilukujen alueella murtolukupotenssit, joilla on supistetut eksponentit ja parittomat nimittäjät, käyttävät reaalista aluetta versus pääalue kompleksitilassa.

Laus ^ *Lista1* $\Rightarrow$ lista

$$p^{\{a, 2, -3\}} \quad \left\{ p^a, p^2, \frac{1}{p^3} \right\}$$

Laskee lausekkeen *Laus* korotettuna *Lista1*:n elementtien potenssiin.

Lista1 ^ *Laus* $\Rightarrow$ lista

$$\{1, 2, 3, 4\}^{-2} \quad \left\{ 1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16} \right\}$$

Laskee *Lista1*:n elementit korotettuna *Laus*:n potenssiin.

^ (potenssi) **painike**

*neliömatrissi*1 ^ *kokonaisluku* ⇒
matrissi

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \quad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

Laskee *neliömatrissi*1:n korotettuna
kokonaisluvun *kokonaisluku* potenssiin.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

*neliömatrissi*1:n on oltava neliömatrissi.

Jos *kokonaisluku* = -1, laskee
käänteismatriisin.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

Jos *kokonaisluku* < -1, laskee
käänteismatriisin korotettuna sopivaan
positiiviseen potenssiin.

x2 (neliö) **painike**

*Laus*12 ⇒ *lauseke*

$$4^2 \quad 16$$

Laskee argumentin neliön.

$$\{2,4,6\}^2 \quad \{4,16,36\}$$

*List*12 ⇒ *lista*

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2 \quad \begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$$

Laskee listan, joka sisältää *List*1:n
elementtien neliöt.

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^{\wedge 2} \quad \begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$$

*neliömatrissi*12 ⇒ *matrissi*

Laskee *neliömatrissi*1:n matriisinelion.
Tämä ei ole sama kuin kunkin elementin
neliön laskeminen. Käytä merkintää .^2,
kun haluat laskea jokaisen elementin
neliön.

.+ (piste lisää) **painikkeet**

*Matriisi*1 .+ *Matriisi*2 ⇒ *matrissi*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} .+ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \quad \begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$$

Laus .+ *Matriisi*1 ⇒ *matrissi*

$$x .+ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \quad \begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$$

*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

Matriisi1 .- *Matriisi2* $\Rightarrow$ *matriisi*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix} = \begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$$

Laus .-*Matriisi1* $\Rightarrow$ *matriisi*

$$x \cdot - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix} = \begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$$

Matriisi1 .-*Matriisi2* laskee matriisin, joka on *Matriisi1*:n ja *Matriisi2*:n vastaavien elementtiparien välinen erotus.

Laus .-*Matriisi1* laskee matriisin, joka on *Laus*:n ja kunkin *Matriisi1*:n elementin erotus.

.· (pistetulo)

  painikkeet

Matriisi1 .· *Matriisi2* $\Rightarrow$ *matriisi*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$$

Laus .·*Matriisi1* $\Rightarrow$ *matriisi*

$$x \cdot \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$$

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*

$$x \cdot / \left(\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \right) = \begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$$

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.

painikkeet

$$Matriisi1.^{\wedge} Matriisi2 \Rightarrow matriisi$$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a^c & 16 \\ b^5 & 3^d \end{bmatrix}$$

Laus .^ Matriisi1 ⇒ matriisi

Matr_iisi1 $\wedge$ *Matr_iisi2* laskee matriisin, jossa *Matr_iisi2*:n jokainen elementti on eksponentti *Matr_iisi1*:n vastaavalle elementille.

$$x \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} x^c & x^4 \\ x^5 & x^d \end{bmatrix}$$

Laus .^Λ *Matriisi* *l* laskee matriisin, jossa *Matriisi* *l*:n jokainen elementti on eksponentti lausekkeelle *Laus*.

(-) painike

-*Lausl* $\Rightarrow$ *lauseke*

-2.43	-2.43
$-\{1, 0.4, 1.2\mathbf{E}19\}$	$\{1, -0.4, -1.2\mathbf{E}19\}$
$-a \cdot b$	$a \cdot b$

$$\neg List\ 1 \Rightarrow lista$$

-*Matriisi* $\Rightarrow$ *matriisi*

Laskee argumentin negaation.

Kun kyseessä on lista tai matriisi, määrittää kaikkien elementtien negaatiot.

Binaarisessa kantalukutilassa:

```
-0b100101  
0b11111111111111111111111111111111▶
```

Jos argumentti on binaarinen tai
heksadesimaalinen kokonaisluku,
negaatio antaa kakkosen komplementin.

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

ctrl  **painikkeet**

Lausl % $\Rightarrow$ *lauseke*

$$Lista1 \% \Rightarrow lista$$

*Matriisi*1 % $\Rightarrow$ *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 ctrl enter.

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**

$Laus1 = Laus2 \Rightarrow \text{Boolean lauseke}$

$Lista1 = Lista2 \Rightarrow \text{Boolean lista}$

$Matriisi1 = Matriisi2 \Rightarrow \text{Boolean matriisi}$

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.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

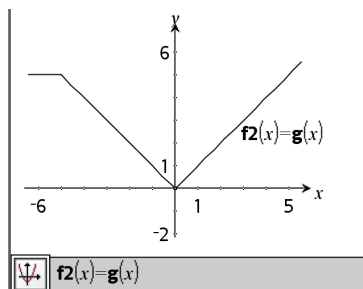
Huomaa esimerkkiä syöttäessäsi: Ohjeet monirivisten ohjelmien ja funktion määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

Esimerkkifunktio, jossa on käytetty matematiikkatestisymboleita: =, ≠, <, ≤, >, ≥

```
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)

ctrl  **painikkeet**

$Laus1 \neq Laus2 \Rightarrow \text{Boolean lauseke}$

$Lista1 \neq Lista2 \Rightarrow \text{Boolean lista}$

$Matriisi1 \neq Matriisi2 \Rightarrow \text{Boolean matriisi}$

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.

Katso esimerkki kohdasta “=” (on yhtä kuin).

$\neq$ (ei yhtä kuin)

  **painikkeet**

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla $\neq$

$<$ (pienempi kuin)

  **painikkeet**

$Laus1 < Laus2 \Rightarrow \text{Boolean lauseke}$

Katso esimerkki kohdasta “=” (on yhtä kuin).

$Lista1 < Lista2 \Rightarrow \text{Boolean lista}$

$Matriisi1 < Matriisi2 \Rightarrow \text{Boolean matriisi}$

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).

$Lista1 \leq Lista2 \Rightarrow \text{Boolean lista}$

$Matriisi1 \leq Matriisi2 \Rightarrow \text{Boolean matriisi}$

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.

$\leq$ (pienempi tai yhtä suuri kuin)

  painikkeet

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla $\leq$

$>$ (suurempi kuin)

  painikkeet

$Laus1 > Laus2 \Rightarrow \text{Boolean lauseke}$

Katso esimerkki kohdasta “=” (on yhtä kuin).

$Lista1 > Lista2 \Rightarrow \text{Boolean lista}$

$Matriisi1 > Matriisi2 \Rightarrow \text{Boolean matriisi}$

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).

$Lista1 \geq Lista2 \Rightarrow \text{Boolean lista}$

$Matriisi1 \geq Matriisi2 \Rightarrow \text{Boolean matriisi}$

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 $\geq$

⇒ (looginen seuraus)**ctrl [=] näppäimet**

BooleanLaus1 ⇒ BooleanLaus2 antaa vastauksena *Boolean lausekkeen*

$5 > 3 \text{ or } 3 > 5$	true
---------------------------	------

BooleanList1 ⇒ BooleanList2 antaa vastauksena *Boolean listan*

$5 > 3 \Rightarrow 3 > 5$	false
---------------------------	-------

$3 \text{ or } 4$	7
-------------------	---

$3 \Rightarrow 4$	-4
-------------------	----

BooleanMatriisi1 ⇒ BooleanMatriisi2 antaa vastauksena *Boolean matriisin*

$\{1, 2, 3\} \text{ or } \{3, 2, 1\}$	$\{3, 2, 3\}$
---------------------------------------	---------------

$\{1, 2, 3\} \Rightarrow \{3, 2, 1\}$	$\{-1, -1, -3\}$
---------------------------------------	------------------

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*

$5 > 3 \text{ xor } 3 > 5$	true
----------------------------	------

BooleanList1 ⇔ BooleanList2 antaa vastauksena *Boolean listan*

$5 > 3 \Leftrightarrow 3 > 5$	false
-------------------------------	-------

$3 \text{ xor } 4$	7
--------------------	---

$3 \Leftrightarrow 4$	-8
-----------------------	----

BooleanMatriisi1 ⇔ BooleanMatriisi2 antaa vastauksena *Boolean matriisin*

$\{1, 2, 3\} \text{ xor } \{3, 2, 1\}$	$\{2, 0, 2\}$
--	---------------

$\{1, 2, 3\} \Leftrightarrow \{3, 2, 1\}$	$\{-3, -1, -3\}$
---	------------------

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

LausI! ⇒ *lauseke*

5! 120

ListaI! ⇒ *lista*

{ { 5,4,3 } }! { 120,24,6 }

MatriisiI! ⇒ *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)

ctrl [⌘] painikkeet

Merkkijono1 & *Merkkijono2* ⇒ *merkkijono*

"Hello "&"Nick" "Hello Nick"

Antaa vastauksena tekstimerkkijonon, joka on *Merkkijono2* liitettynä *Merkkijono1*:een.

d() (derivaatta)

Katalogi >

d(*LausI*, *Muutt*[, *Aste*])⇒*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(*ListaI*, *Muutt*[, *Aste*])⇒*lista*

$\frac{d}{dy} \left(\frac{d}{dx} (x^2 \cdot y^3) \right)$ $6 \cdot y^2 \cdot x$

d(*MatriisiI*, *Muutt*[, *Aste*])⇒*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)

$f(Laus\ I, Muutt[, Ala, Ylä]) \Rightarrow lauseke$

$f(Laus\ I, Muutt[, Vakio]) \Rightarrow lauseke$

Laskee *LausI*:n integraalin muuttujan *Muutt* suhteen välillä *Ala* - *Ylä* olevista arvoista.

Huomaa: Katso myös **Määrätyn** tai **Määrittämättömän integraalin malli**, sivu 7.

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

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*.

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

$$\int (a \cdot x^2, x, c) = \frac{a \cdot x^3}{3} + c$$

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 *LausI*:n paloille, joita se ei pysty määrittämään sisäänrakennettujen funktioidensa ja operaattoreidensa eksplisiittisenä äärellisenä kombinaationa.

$$\int b \cdot e^{-x^2 + \frac{a}{x^2 + a^2}} dx = b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$$

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.

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina .

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

iPad®: Pidä **enter** ja valitse .

$$\int_{-1}^1 e^{-x^2} dx \quad 1.49365$$

∫()-komentoa voidaan sijoittaa sisäkkäin usean integraalin suorittamiseksi. Integroinnin raja-arvot voivat riippua niiden ulkopuolella olevista integrointimuuttujista.

Huomaa: Katso myös **nInt()**, sivu 131.

$$\int_0^a \int_0^x \ln(x+y) dy dx$$

$$\frac{a^2 \cdot \ln(a)}{2} + \frac{a^2 \cdot (4 \cdot \ln(2) - 3)}{4}$$

√() (neliöjuuri)

  **painikkeet**

√(*Lausl*) ⇒ *lauseke*

$$\sqrt{4} \quad 2$$

√(*Lista1*) ⇒ *lista*

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

Laskee argumentin neliöjuuren.

Kun kyseessä on lista, laskee kaikkien *Lista1*:n elementtien neliöjuuret.

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

Huomaa: Katso myös **Neliöjuurimalli**, sivu 1.

Π() (tulo)

Katalogi > 

Π(*Lausl*, *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** (...).

$$\prod_{k=1}^n (k^2) \quad (n!)^2$$

Sieventää *Lausl*:n jokaisen *Muutt*:n arvon suhteen väliltä *Matala* - *Korkea* ja laskee vastausten tulon.

$$\prod_{n=1}^5 \left(\left\{ \frac{1}{n}, n, 2 \right\} \right) \quad \left\{ \frac{1}{120}, 120, 32 \right\}$$

Huomaa: Katso myös **Kertolaskumalli** (Π), sivu 6.

$\Pi()$ (tulo)

Katalogi >

 $\Pi(Laus\ I, Muutt, Matala, Matala-1) \Rightarrow 1$

$$\prod_{k=4}^3 (k) = 1$$

 $\Pi(Laus\ I, Muutt, Matala, Korkea) \Rightarrow 1/\Pi(Laus\ I, Muutt, Korkea+1, Matala-1)$, jos $Korkea < Matala-1$

Esimerkkien kertolaskukaavat 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.

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) = 6$$

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \cdot \prod_{k=2}^4 \left(\frac{1}{k}\right) = \frac{1}{4}$$

$\Sigma()$ (summa)

Katalogi >

 $\Sigma(Laus\ I, Muutt, Matala, Korkea) \Rightarrow$ 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.

 $\Sigma(Laus\ I, Muutt, Matala, Matala-1) \Rightarrow 0$

$$\sum_{n=1}^5 \left(\frac{1}{n}\right) = \frac{137}{60}$$

$$\sum_{k=1}^n (k^2) = \frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$$

$$\sum_{n=1}^{\infty} \left(\frac{1}{n^2}\right) = \frac{\pi^2}{6}$$

$$\sum_{k=4}^3 (k) = 0$$

 $\Sigma(Laus\ I, Muutt, Matala, Korkea) \Rightarrow -\Sigma(Laus\ I, Muutt, Korkea+1, Matala-1)$, jos $Korkea < Matala-1$

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.

$$\sum_{k=4}^1 (k) = -5$$

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) = 4$$

ΣInt(*NPmt1*, *NPmt2*, *N*, *I*, *PV*, [*Pmt*],
[*FV*], [*PpY*], [*CpY*], [*PmtAt*],
[*pyörArvo*])⇒*arvo*

ΣInt(1,3,12,4.75,20000,,,12,12) -218.11

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 211.

- 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.

tbl:=amortTbl(12,12,4.75,20000,,,12,12)

0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

ΣInt(1,3,*tbl*) -218.11

ΣPrn(*NPmt1*, *NPmt2*, *N*, *I*, *PV*, [*Pmt*],
[*FV*], [*PpY*], [*CpY*], [*PmtAt*],
[*pyörArvo*])⇒*arvo*

ΣPrn(1,3,12,4.75,20000,,,12,12) -4911.47

ΣPrn

(*NPmt1*,*NPmt2*,*amortTable*)⇒*arvo*

Lyhennystoiminto, joka laskee pääoman summan määritetyn maksujakson ajalta.

NPmt1 ja *NPmt2* määrittävät maksujakson alku- ja loppurajat.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 211.

- 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 pyörityksessä käytettävien desimaalien määrän.
Oletusarvo=2.

ΣPrn(*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 ΣInt(), edellä, sekä Bal(), sivu 17.

<i>tbl:=amortTbl(12,12,4.75,20000,,,12,12)</i>			
0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01
ΣPrn(1,3,tbl)			-4911.47

(epäsuora operaattori)

  **painikkeet**

muuttNimiMerkkijono

Viittaa muuttujaan, jonka nimi on *muuttNimiMerkkijono*. Tällä operaattorilla voit luoda muuttujanimiä funktion sisältä merkkijonojen avulla.

#("x"&"y"&"z") xyz

Luo muuttujan xyz tai viittaa siihen.

10→r	10
"r"→s1	"r"
#s1	10

Laskee arvon muuttujalle (r), jonka nimi on tallennettu muuttujaan s1.

E (kymmenpotenssimuoto)

EE painike

*mantissa*E*eksponentti*

23000.	23000.
2300000000.+4.1E15	4.1E15
3·10 ⁴	30000

Syöttää luvun kymmenpotenssimuodossa. Luku tulkitaan seuraavasti: *mantissa* × 10*eksponentti*.

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)

1 painike

*Laus l*g⇒*lauseke*

Aste-, graadi- tai radiaanikulmatilassa.

*Lista l*g⇒*lista*

$$\cos(50^{\circ}) \quad \frac{\sqrt{2}}{2}$$

*Matriisi l*g⇒*matriisi*

$$\cos\left\{\left\{0, 100^{\circ}, 200^{\circ}\right\}\right\} \quad \left\{1, 0, -1\right\}$$

Tämän funktion avulla voit määrittää graadikulman ollessasi aste- tai radiaanikulmatilassa.

Kun laskin on radiaanikulmatilassa, kertoo *Laus l*:n arvolla $\pi/200$.

Kun laskin on astekulmatilassa, kertoo *Laus l*:n arvolla g/100.

Graadikulmatilassa antaa vastauksena lausekkeen *Laus l* muuttumattomana.

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla @g.

r (radiaani)

1 painike

*Laus l*r⇒*lauseke*

Aste-, graadi- tai radiaanikulmatilassa:

r (radiaani)

1 painike

$\text{Lista } I^r \Rightarrow \text{lista}$

$\text{Matriisi } I^r \Rightarrow \text{matriisi}$

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 I^r , jos haluat pakottaa funktion määritelmän yksiköksi radiaanit riippumatta tilasta, joka on käytössä funktion käytön aikana.

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla @r.

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

$$\cos\left(\left\{0^r, \frac{\pi}{12}, \frac{\pi}{r}, \left(\frac{\pi}{r}\right)^r\right\}\right) \quad \left\{1, \frac{(\sqrt{3+1})\sqrt{2}}{4}, -1\right\}$$

° (aste)

1 painike

$\text{Laus } I^\circ \Rightarrow \text{lauseke}$

$\text{Lista } I^\circ \Rightarrow \text{lista}$

$\text{Matriisi } I^\circ \Rightarrow \text{matriisi}$

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 **ctrl** **enter**.

Windows®: Paina **Ctrl+Enter**.

Macintosh®: Paina **⌘+Enter**.

iPad®: Pidä **enter** ja valitse $\approx$.

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) \quad \{1, 0.707107, 0., 0.864976\}$$

$dd^{\circ}mm'ss.ss'' \Rightarrow lauseke$

Astekulmatilassa:

 dd Positiivinen tai negatiivinen luku

25°13'17.5" 25.2215

 mm Ei-negatiivinen luku

25°30' 51

 $ss.ss$ Ei-negatiivinen luku

2

Laskee $dd+(mm/60)+(ss.ss/3600)$.

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)

 $[Säde, \angle \theta_Kulma] \Rightarrow vektori$
(polaarinen syöte)

Radiaanikulmatilassa ja kun vektorimuoto on asetettu valintaan:
suorakulma

 $[Säde, \angle \theta_Kulma, Z_Koordinaatti] \Rightarrow vektori$
(lieriömainen syöte)

$$\left[5 \quad \angle 60^{\circ} \quad \angle 45^{\circ} \right] \left[\frac{5 \cdot \sqrt{2}}{4} \quad \frac{5 \cdot \sqrt{6}}{4} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

 $[Säde, \angle \theta_Kulma, \angle \theta_Kulma] \Rightarrow vektori$
(pallonmuotoinen syöte)

sylinteri

Laskee koordinaatit vektorina riippuen vektorimuotoalan 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

$$\left[5 \quad \angle 60^{\circ} \quad \angle 45^{\circ} \right] \left[5 \quad \angle \frac{\pi}{3} \quad \angle \frac{\pi}{4} \right]$$

 $(Magnitudi \angle Kulma) \Rightarrow kompleksiarvo$
(polaarinen syöte)

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

Syöttää kompleksilukuarvon ($r \angle \theta$) polaarissa muodossa. *Kulma* tulkitaan nykyisen kulmatila-asetuksen mukaisesti.

$$5+3 \cdot i \left(10 \angle \frac{\pi}{4} \right) \quad 5-5 \cdot \sqrt{2} + (3-5 \cdot \sqrt{2}) \cdot i$$

∠ (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.

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

_ (alaviiva edustaa tyhjää elementtiä)

KatsoTyhjät elementit, sivu 272.

_ (alaviiva edustaa yksikön nimeä)

  painikkeet

Laus_Yksikkö

Määrittää lausekkeen *Laus* yksiköt. Kaikkien yksiköiden nimien alussa on oltava alaviiva.

Voit käyttää valmiiksi määritettyjä yksiköitä tai luoda omia yksiköitä. Valmiiksimääritetyt yksiköt on esitetty vakioden ja mittausyksiköiden moduulissa. Voit valita yksiköiden nimet Katalogista tai kirjoittaa ne suoraan.

Muuttuja_

$$3\cdot_m\blacktriangleright_ft \quad 9.84252\cdot_ft$$

Huomaa: Muunnoksen symboli $\blacktriangleright$ löytyy

Katalogista. Napsauta ensin  ja sen jälkeen **Matemaattiset operaattorit**.

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*

$3 \cdot \text{m} \rightarrow \text{ft}$ $9.84252 \cdot \text{ft}$

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^ (*Lista l*) $\Rightarrow$ *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 *Lista l*:n elementtien potenssiin.

10^(*neliömatrissi l*) $\Rightarrow$ *neliömatrissi*

Laskee luvun 10 korotettuna *neliömatrissi l*:n potenssiin. Tämä ei ole sama kuin kunkin elementin 10-potenssiin korottamisen laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} \quad \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

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

^-1(käänteisluku)

Katalogi > 

Laus l ^-1 $\Rightarrow$ *lauseke*

$$(3.1)^{-1} \quad 0.322581$$

Lista l ^-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 *Lista l*:n elementtien käänteisluvut.

neliömatrissi l ^-1 $\Rightarrow$ *neliömatrissi*

Laskee *neliömatrissi l*:n käänteisluvun.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

neliömatrissi l:n oltava ei-singulaarinen neliömatrissi.

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} \frac{-2}{a-2} & \frac{1}{a-2} \\ \frac{a}{2 \cdot (a-2)} & \frac{-1}{2 \cdot (a-2)} \end{bmatrix}$$

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.

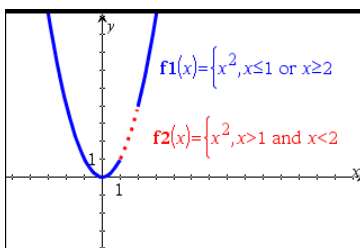
Laus | *Muuttuja* = *arvo* korvaa *arvon* jokaisessa *Muuttujan* esiintymiskohdassa lausekkeessa *Laus*.

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.

$$\begin{array}{rcl} x+1|x=3 & & 4 \\ x+y|x=\sin(y) & & \sin(y)+y \\ x+y|\sin(y)=x & & x+y \end{array}$$

$$\begin{array}{rcl} x^3-2\cdot x+7 \rightarrow f(x) & & Done \\ f(x)|x=\sqrt{3} & & \sqrt{3}+7 \\ (\sin(x))^2+2\cdot \sin(x)-6|\sin(x)=d & & d^2+2\cdot d-6 \end{array}$$

$$\begin{array}{rcl} \text{solve}(x^2-1=0,x)|x>0 \text{ and } x<2 & & x=1 \\ \sqrt{x}\cdot \sqrt{\frac{1}{x}}|x>0 & & 1 \\ \sqrt{x}\cdot \sqrt{\frac{1}{x}} & & \sqrt{\frac{1}{x}}\cdot \sqrt{x} \end{array}$$



I (rajoittava operaattori)

  -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.

$\text{solve}(x^2-1=0, x) x \neq 1$	$x = -1$
---------------------------------------	----------

→ (tallenna)

  **painike**

Laus → *Muutt*

Lista → *Muutt*

Matriisi → *Muutt*

Laus → *Funktio*(*Param1*,...)

Lista → *Funktio*(*Param1*,...) *Matriisi* → *Funktio*(*Param1*,...)

$\frac{\pi}{4} \rightarrow \text{myvar}$	$\frac{\pi}{4}$
$2 \cdot \cos(x) \rightarrow y1(x)$	<i>Done</i>
$\{1, 2, 3, 4\} \rightarrow \text{lst5}$	$\{1, 2, 3, 4\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \text{matg}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
"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**.

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.

$myvar := \frac{\pi}{4}$	$\frac{\pi}{4}$
$y1(x) := 2 \cdot \cos(x)$	Done
$lst5 := \{1, 2, 3, 4\}$	$\{1, 2, 3, 4\}$
$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
$str1 := \text{"Hello"}$	"Hello"

© [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 $g(n) = \text{Func}$	
© Declare variables	
Local $i, result$	
$result := 0$	
For $i, 1, n, 1$ © Loop n times	
$result := result + i^2$	
EndFor	
Return $result$	
EndFunc	
	Done
$g(3)$	14

Ob, Oh		[0] [B] painikkeet, [0] [H] painikkeet	
Ob <i>binaariluku</i>		Desimaalisessa kantalukutilassa:	
Oh <i>heksadesimaaliluku</i>			
Määrittää binaari- (Ob) tai heksadesimaaliluvun (Oh). Syöttääksesi binaari- tai heksadesimaaliluvun sinun on syötettävä etuliite Ob tai Oh riippumatta kantalukutilasta. Ilman etuliitettä lukua käsitellään desimaalilukuna (kantalu 10).			
Tulokset näytetään kantalukutilan mukaisesti.		Binaarisessa kantalukutilassa:	
		Ob10+OhF+10	27
		Heksadesimaalisessa kantalukutilassa:	
		Ob10+OhF+10	0b11011
		Ob10+OhF+10	0h1B

TI-Nspire™ CX II – Piirtokomennot

Tämä on täydentävä asiakirja TI-Nspire™-viiteoppaalle ja TI-Nspire™ CAS -viiteoppaalle. Kaikki TI-Nspire™ CX II -komennot sisällytetään ja julkaistaan TI-Nspire™-viiteoppaassa ja TI-Nspire™ CAS -viiteoppaassa.

Kuvaajien ohjelmointi

Uusia komentoja on lisätty TI-Nspire™ CX II -kämmenlaitteisiin ja TI-Nspire™-työpöytäsovellukseen kuvaajaohjelmointia varten.

TI-Nspire™ CX II -kämmenlaitteet siirtyvät tähän kuvaajatilaan, kun ne suorittavat kuvaajakomentoja, ja siirtyvät ohjelman suorittamisen jälkeen takaisin kontekstiin, jossa ohjelma suoritettiin.

Näytön yläreunassa näkyy ”Running...”, kun ohjelmaa suoritetaan. Näkyviin tulee ”Finished”, kun ohjelma on valmis. Minkä tahansa näppäimen painaminen siirtää järjestelmän pois kuvaajatilasta.

- Siirtyminen kuvaajatilaan käynnistyy automaattisesti, kun jokin Draw (kuvaajat) -komennoista kohdataan TI-Basic -ohjelman suorituksen aikana.
- Tämä siirtyminen tapahtuu vain, kun ohjelma suoritetaan laskimella; asiakirjassa tai muistiolehtiön laskimessa.
- Siirtyminen pois kuvaajatilasta tapahtuu ohjelman sulkemisen jälkeen.
- Kuvaajatila on käytettävissä vain TI-Nspire™ CX II- kämmenlaitteissa ja TI-Nspire™ CX II -kämmenlaitteiden työpöytäversion näkymässä. Tämä tarkoittaa, että se ei ole käytettävissä tietokoneen asiakirjanäkymässä työpöytäversiossa tai iOS-käyttöjärjestelmässä.
 - Jos kuvaajakomento kohdataan, kun TI-Basic -ohjelmaa suoritetaan virheellisestä kontekstista, näkyviin tulee virheviesti ja TI-Basic -ohjelma suljetaan.

Kuvaajanäyttö

Kuvaajanäytön yläosassa on otsikko, jota ei voi kirjoittaa kuvaajakomennoilla.

Kuvaajanäytön piirustusalue tyhjennetään (color = 255,255,255), kun kuvaajanäyttö alustetaan.

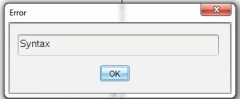
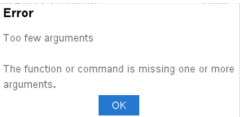
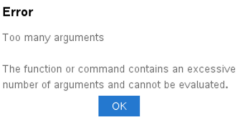
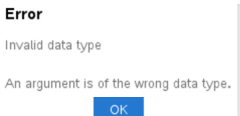
Kuvaajanäyttö	Oletus
Korkeus	212
Leveys	318
Väri	valkoinen: 255,255,255

Oletusnäkymä ja asetukset

- Yläpalkin tilakuvakkeet (akun tila, press-to-start-tila, verkon ilmainen jne.) eivät näy, kun kuvaajaohjelma on käynnissä.
- Piirtämisen oletusväri: Musta (0,0,0)
- Kynän oletustyyli – normaali, tasainen
 - Paksuus: 1 (tasainen), 2 (normaali), 3 (paksuin)
 - Tyyli 1 (tasainen), 2 (pisteet), 3 (viivat)
- Kaikki piirtokomennot käyttävät nykyisiä väri- ja kynäasetuksia eli joko oletusarvoja tai TI-Basic-komentojen avulla asetettuja arvoja.
- Tekstin fontti on pysyvä eikä sitä voi muuttaa.
- Kaikki kuvaajanäytön piirrokset piirretään leikkausikkunassa, joka on kuvaajanäytön piirustusalueen kokoinen. Mitään piirroksia, jotka ulottuvat leikatun kuvaajanäytön piirustusalueen ulkopuolelle, ei piirretä. Virheviestiä ei näytetä.
- Kaikki piirtokomennoille määritetyt x- ja y-koordinaatit on määritelty siten, että 0,0 on kuvaajanäytön piirustusalueen vasemmassa yläkulmassa.
 - **Poikkeukset:**
 - **DrawText** käyttää koordinaatteja tekstin rajaavan ruudun vasemmassa alakulmassa.
 - **SetWindow** käyttää näytön vasenta alakulmaa.
- Kaikki komentojen parametrit voidaan antaa lausekkeina, jotka arvioidaan numeroksi, joka pyöristetään sitten lähimpään kokonaislukuun.

Kuvaajanäytön virheviestit

Jos validointi epäonnistuu, näyttöön ilmestyy virheviesti.

Virheviestit	Kuvaus	Tarkastele
Virhe Syntaksi	Jos syntaksin tarkastuksessa löytyy syntaksivirheitä, näkyviin tulee virheilmoitus, ja editori yrittää asettaa kohdistimen ensimmäisen virheen kohdalle, jotta voit korjata sen.	
Virhe Liian vähän argumentteja	Funktiosta tai komennosta puuttuu yksi tai useampia argumentteja	
Virhe Liian monta argumenttia	Funktio tai komento sisältää liian monta argumenttia eikä sitä voida arvioida.	
Virhe Virheellinen tietotyyppi	Argumentin tietotyyppi on väärä.	

Virheelliset komennot kuvaajatilassa

Jotkin komennot eivät ole sallittuja, kun ohjelma siirtyy kuvaajatilaan. Jos nämä komennot kohdataan kuvaajatilassa, näyttöön tulee virheviesti ja ohjelma suljetaan.

Komentoa ei sallita	Virheviestit
Pyyntö	Pyyntöä ei voida toteuttaa kuvaajatilassa
PyydäMerkkij	RequestStr-toimintoa ei voida toteuttaa kuvaajatilassa
Teksti	Tekstiä ei voida toteuttaa kuvaajatilassa

Komennot, jotka kirjoittavat tekstin laskimeen – **disp** ja **dispAt** – ovat tuettuja komentoja kuvaajakontekstissa. Näiden komentojen teksti lähetetään Laskin-näyttöön (ei Kuvaaja-näyttöön) ja se näkyy sen jälkeen, kun ohjelma on suljettu ja järjestelmä siirtyy takaisin Laskin-sovellukseen.

Tyhjennä

Katalogi > 
CXII**Clear** *x, y, width, height*

Tyhjentää koko näytön, jos parametreja ei ole.

Jos *x, y, width* ja *height* on määritetty, parametrien määrittämä suorakulmio tyhjennetään.

Tyhjennä

Tyhjentää koko näytön

Clear 10,10,100,50

Tyhjentää suorakulmioalueen, jonka vasen yläkulma on (10, 10), leveys on 100 ja korkeus on 50.

DrawArcKatalogi > 
CXII**DrawArc** *x, y, width, height, startAngle, arcAngle*

Piirrä kaari rajaavan suorakulmion sisällä annetulla aloitus- ja kaarikulmalla.

x, y: rajaavan suorakulmion vasen yläkoordinaatti*width, height*: rajaavan suorakulmion mitat

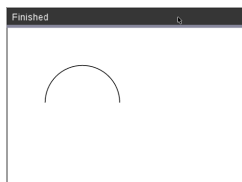
"Arc Angle" määrittää kaaren kaarevuuden.

Nämä parametrit voidaan antaa lausekkeina, jotka arvioidaan numeroksi, joka pyöristetään sitten lähimpään kokonaislukuun.

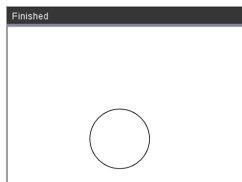
DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180

Katso myös: [FillArc](#)**DrawCircle**Katalogi > 
CXII**DrawCircle** *x, y, radius**x, y*: keskikohdan koordinaatti*radius*: ympyrän säde

DrawCircle 150,150,40

Katso myös: [FillCircle](#)

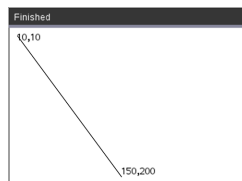
DrawLine $x1, y1, x2, y2$

Piirrä viiva pisteillä $x1, y1, x2, y2$.

Lausekkeet, jotka arvioidaan numeroksi, joka pyöristetään sitten lähimpään kokonaislukuun.

Näytön rajat: Jos määritetyt koordinaatit aiheuttavat viivan osan piirtämisen kuvaajanäytön ulkopuolelle, kyseinen viivan osa leikataan eikä virheilmoitusta näytetä.

DrawLine 10,10,150,200



DrawPoly

Komennoilla on kaksi varianttia:

DrawPoly $xlist, ylist$

tai

DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$

Huomaa: DrawPoly $xlist, ylist$

Muoto yhdistää kohteet $x1, y1$ to $x2, y2, x2, y2$ to $x3, y3$ ja niin edelleen.

Huomaa: DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$

Kohteita xn, yn EI yhdistetä automaattisesti kohteisiin $x1, y1$.

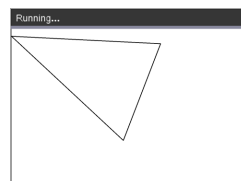
Lausekkeet, jotka arvioidaan todellisten liukulukujen luetteloon $xlist, ylist$

Lausekkeet, jotka arvioidaan yksittäiseksi liukuluvuksi $x1, y1...xn, yn$ = monikulmion kärkien koordinaatit

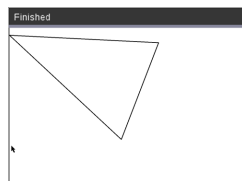
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

DrawPoly $xlist, ylist$



DrawPoly 0,10,200,20,150,150,0,10



Huomaa: DrawPoly: Lisää kokomitat (leveys/korkeus), jotka vastaavat piirrettyjä viivoja.

Viivat piirretään rajaavan ruudun määrittelyn koordinaatin ja mittojen ympärille siten, että piirretyn monikulmion todellinen koko on suurempi kuin leveys ja korkeus.

Katso myös: [FillPoly](#)

DrawRect

DrawRect *x, y, width, height*

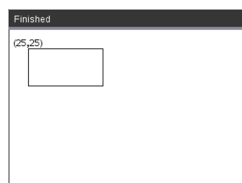
x, y: suorakulmion vasen yläkoordinaatti

width, height: suorakulmion leveys ja korkeus (suorakulmio, joka on piirretty alaspäin ja oikealle lähtökoordinaatista)

Huomaa: Viivat piirretään rajaavan ruudun määrittelyn koordinaatin ja mittojen ympärille siten, että piirretyn suorakulmion todellinen koko on suurempi kuin leveys ja korkeus osoittavat.

Katso myös: [FillRect](#)

DrawRect 25,25,100,50



DrawText

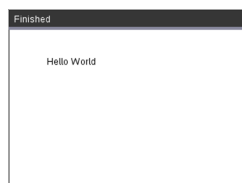
DrawText *x, y, exprOrString1*
[,exprOrString2]...

x, y: tekstisyötön koordinaatti

Piirtää tekstin kohteeseen *exprOrString* määritetyssä *x, y* -koordinaattien sijainnissa.

Säännöt kohteelle *exprOrString* ovat samat kuin kohteelle **Disp** – **DrawText** voi käsitellä useita argumentteja.

DrawText 50,50,"Hello World"



FillArc

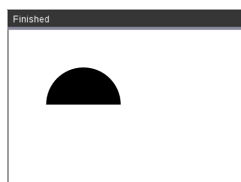
Katalogi > 
CXII**FillArc** *x, y, width, height, startAngle, arcAngle**x, y*: rajaavan suorakulmion vasen yläkoordinaatti

Piirrä ja täytä kaari rajaavan suorakulmion sisällä annetulla aloitus- ja kaarikulmalla.

Täytön oletusväri on musta. Täyttöväri voidaan asettaa komennolla [SetColor](#).

"Arc Angle" määrittää kaaren kaarevuuden.

FillArc 50,50,100,100,0,180



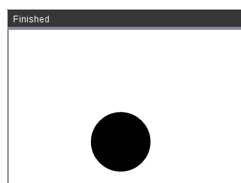
FillCircle

Katalogi > 
CXII**FillCircle** *x, y, radius**x, y*: keskikohdan koordinaatti

Piirrä ja täytä ympyrä määritellyllä keskikohdalla ja määritetyllä säteellä.

Täytön oletusväri on musta. Täyttöväri voidaan asettaa komennolla [SetColor](#).

FillCircle 150,150,40

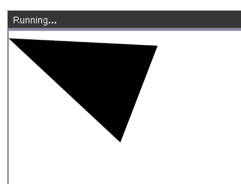


Täällä!

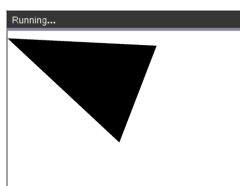
FillPoly

Katalogi > 
CXII**FillPoly** *xlist, ylist*

tai

FillPoly *x1, y1, x2, y2, x3, y3...xn, yn***Huomaa:** Viivan ja värin määrittää komennot [SetColor](#) ja [SetPen](#)*xlist*:= {0,200,150,0}*ylist*:= {10,20,150,10}FillPoly *xlist, ylist*

FillPoly 0,10,200,20,150,150,0,10



FillRect

FillRect *x, y, width, height*

FillRect 25,25,100,50

x, y: suorakulmion vasen yläkoordinaatti*width, height*: suorakulmion leveys ja korkeus

Piirrä ja täytä suorakulmio, jonka vasen yläkulma koordinaatissa, jonka määrittää (*x,y*).

Täytön oletusväri on musta. Täyttöväri voidaan asettaa komennolla [SetColor](#).

Huomaa: Viivan ja värin määrittää komennot [SetColor](#) ja [SetPen](#)



getPlatform()Katalogi > 
CXII**getPlatform()**

getPlatform()

"dt"

Palauttaa:

"dt" työpöytäohjelmistosovelluksissa

"hh" TI-Nspire™ CX -kämmenlaitteissa

"ios" TI-Nspire™ CX iPad® -sovelluksessa

PaintBuffer

Katalogi > 
CXII

PaintBuffer

Maalaa kuvaajan puskurin näytölle

Tätä komentoa käytetään yhdessä UseBuffer-komennon kanssa näyttönopeuden lisäämiseksi, kun ohjelma luo useita kuvaajaobjekteja.

UseBuffer

Luvulle n,1,10

x:=randInt(0,300)

y:=randInt(0,200)

radius:=randInt(10,50)

Wait 0,5

DrawCircle x,y,radius

EndFor

PaintBuffer

Tämä ohjelma näyttää 10 ympyrää samanaikaisesti.

Jos UseBuffer-komento poistetaan, jokainen ympyrä näytetään sellaisena kuin se on piirretty.

Katso myös: [UseBuffer](#)

PlotXY $x, y, shape$

x, y : koordinaatit muodon muodostamiseen

$shape$: numero väliltä 1–13, joka määrittää muodon

- 1 – Täytetty ympyrä
- 2 – Tyhjä ympyrä
- 3 – Täytetty neliö
- 4 – Tyhjä neliö
- 5 – Risti
- 6 – Plusmerkkikuvio
- 7 – Ohut
- 8 – keskikokoinen piste, täytetty
- 9 – keskikokoinen piste, tyhjä
- 10 – suurempi piste, täytetty
- 11 – suurempi piste, tyhjä
- 12 – suurin piste, täytetty
- 13 – suurin piste, tyhjä

PlotXY 100,100,1

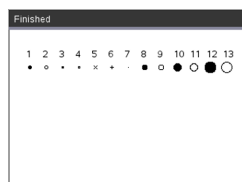


Luvulle $n, 1, 13$

DrawText $1+22*n, 40, n$

PlotXY $5+22*n, 50, n$

EndFor



SetColor
Katalogi > 
CXII
SetColor

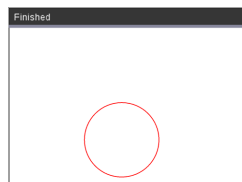
Red-arvo, Green-arvo, Blue-arvo

Kelvolliset arvot punaiselle, vihreälle ja siniselle ovat välillä 0–255

Asettaa tulevien Draw-komentojen värin

SetColor 255,0,0

DrawCircle 150,150,100

**SetPen**
Katalogi > 
CXII
SetPen

paksuus, tyyli

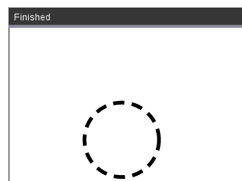
Paksuus: 1 <= paksuus <= 3 | 1 on ohuin, 3 on paksuin

style: 1 = tasainen, 2 = pisteet, 3 = viivat

Asettaa tulevien Draw-komentojen kynän tyylin

SetPen 3,3

DrawCircle 150,150,50

**SetWindow**
Katalogi > 
CXII
SetWindow

xMin, xMax, yMin, yMax

Luo loogisen ikkunan, joka kartoittaa kuvaajan piirustusalueen. Kaikki parametrit vaaditaan.

Jos piirretyn objektin osa on ikkunan ulkopuolella, syöttö leikataan (ei näytetä) eikä virheilmoitusta näytetä.

SetWindow 0,160,0,120

Asettaa syöttöikkunan vasemman alakulman kohtaan 0,0. Leveys on 160 ja korkeus on 120.

DrawLine 0,0,100,100

SetWindow 0,160,0,120

SetPen 3,3

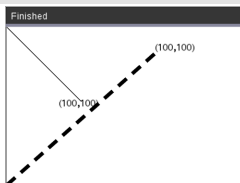
DrawLine 0,0,100,100

Jo $x_{\min}$ on suurempi tai yhtä suuri kuin $x_{\max}$ tai $y_{\min}$ on suurempi tai yhtä suuri kuin $y_{\max}$, näyttöön tulee virheviesti.

Mitään objekteja, jotka on piirretty ennen SetWindow-komentoa, ei piirretä uudelleen uudessa kokoonpanossa.

Käytä seuraavia ikkunan parametrien nollaamiseksi oletusarvoihin:

SetWindow 0,0,0,0



UseBuffer

Katalogi > 
CXII

UseBuffer

Piirrä kuvaaja puskuriin näytön sijaan (suorituskyvyn parantamiseksi)

Tätä komentoa käytetään yhdessä PaintBuffer-komennon kanssa näyttönopeuden lisäämiseksi, kun ohjelma luo useita kuvaajaobjekteja.

UseBuffer-komennolla kaikki kuvaajat näytetään vasta seuraavan PaintBuffer-komennon suorittamisen jälkeen.

UseBufferia täytyy käyttää vain kerran ohjelmassa, eli jokainen käytetty PaintBuffer-komento ei tarvitse vastaavaa UseBuffer-komentoa.

UseBuffer

```
Luvulle n,1,10
x:=randInt(0,300)
y:=randInt(0,200)
radius:=randInt(10,50)
Wait 0,5
DrawCircle x,y,radius
EndFor
PaintBuffer
```

Tämä ohjelma näyttää 10 ympyrää samanaikaisesti.

Jos UseBuffer-komento poistetaan, jokainen ympyrä näytetään sellaisena kuin se on piirretty.

Katso myös: [PaintBuffer](#)

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 52, ja **isVoid()**, sivu 101.

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.

$_$	—
$\gcd(100,_)$	—
$3+_$	—
$\{5,_,10\}-\{3,6,9\}$	$\{2,_,1\}$

Tyhjiä elementtejä sisältävät lista-argumentit

Seuraavat funktiot ja komennot eivät huomioi (ohittavat ne) lista-argumenteista löytyviä tyhjiä elementtejä.

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

$\text{sum}(\{2,_,3,5,6,6\})$	16.6
$\text{median}(\{1,2,_,_,3\})$	2
$\text{cumulativeSum}(\{1,2,_,4,5\})$	$\{1,3,_,7,12\}$
$\text{cumulativeSum}\left(\begin{bmatrix} 1 & 2 \\ 3 & - \\ 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 2 \\ 4 & - \\ 9 & 8 \end{bmatrix}$

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>
<code> </code> (itseisarvo)	<code>abs(...)</code>
<code>sqrt()</code>	<code>sqrt(...)</code>
<code>d()</code>	<code>derivative(...)</code>
<code>int()</code>	<code>integral(...)</code>
$\Sigma()$ (yhteenlaskumalli)	<code>sumSeq(...)</code>
$\Pi()$ (tulon malli)	<code>prodSeq(...)</code>
<code>sin⁻¹()</code> , <code>cos⁻¹()</code> , ...	<code>arcsin(...)</code> , <code>arccos(...)</code> , ...
<code>ΔList()</code>	<code>deltaList(...)</code>
<code>ΔtmpCnv()</code>	<code>deltaTmpCnv(...)</code>

Tietokoneen näppäimistöltä

Syötettävä kohde:	Pikavalinta:
<code>c1</code> , <code>c2</code> , ... (vakiot)	<code>@c1</code> , <code>@c2</code> , ...
<code>n1</code> , <code>n2</code> , ... (kokonaislukuvakiot)	<code>@n1</code> , <code>@n2</code> , ...

Syötettävä kohde:	Pikavalinta:
i (imaginaarinen vakio)	@i
e (luonnollisen logaritmin kantaluku e)	@e
E (kymmenpotenssimuoto)	@E
T (transponointi)	@t
r (radiaanit)	@r
° (asteet)	@d
g (graadit)	@g
∠ (kulma)	@<
► (muunnos)	@>
►Decimal, ►approxFraction (), jne.	@>Decimal, @>approxFraction(), jne.

EOS-järjestelmän (yhtälökäyttöjärjestelmä) hierarkia

Tässä kappaleessa kuvataan yhtälökäyttöjärjestelmä (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 sulkuryhmyityksiä 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 (^r), alaindeksi ([]), transponointi (^T)
 - 5 Potenssiin korotus, potenssioperaattori (^)
 - 6 Negaatio (-)
 - 7 Merkkijonojen ketjutus (&)
 - 8 Kertolasku (•), jakolasku (/)
 - 9 Yhteenlasku (+), vähennyslasku (-)
 - 10 Yhtäläisyysuhteet: on yhtä kuin (=), ei ole yhtä kuin (≠ tai /=), pienempi kuin (<), pienempi tai yhtä suuri kuin (≤ tai <=), suurempi tai yhtä suuri kuin (>), suurempi tai yhtä suuri kuin (≥ tai >=)
 - 11 Looginen **not**
 - 12 Looginen **and**
 - 13 Looginen **or**
 - 14 **xor, nor, ei-ja**
 - 15 Looginen seuraus (⇒)
 - 16 Looginen kaksoisseuraus, XNOR (⇔)
 - 17 Rajoittava operaattori ("|")
 - 18 Tallenna (→)
-

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 komennoiksi 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 2^4 (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 negatiotiota. 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.

TI-Nspire CX II – TI-Basic-ohjelmointiominaisuudet

Automaattinen sisennys Ohjelmointieditorissa

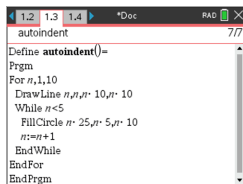
TI-Nspire™-ohjelmaeditori sisentää nyt automaattisesti lauseita lohkokomennon sisällä.

Lohkokomennot ovat If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

Editori lisää automaattisesti välilyönnit ohjelmakäskyihin lohkokomennon sisällä.

Lohkon sulkemiskomento kohdistetaan avauskomennon kanssa.

Alla oleva esimerkki näyttää automaattisen sisennyksen sisäkkäisissä lohkokomennossa.



Kopioidut ja liitetyt koodifragmentit säilyttävät alkuperäisen sisennyksen.

Ohjelmiston aikaisemmassa versiossa luodun ohjelman avaaminen säilyttää alkuperäisen sisennyksen.

TI-Basicin virheviestien parannukset

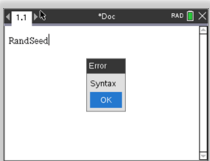
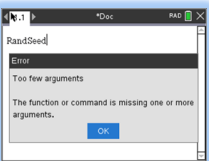
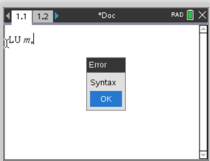
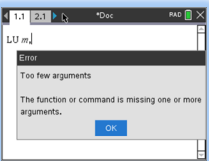
Virheet

Virhetila	Uusi viesti
Ehtokohteen virhe (If/While)	Ehdollista lausuntoa ei voitu ratkaista arvolla TRUE tai FALSE HUOMAA: Koska muutos on sijoitettu riville virheen kanssa, ei ole enää tarpeen tarkentaa, onko virhe "If"- vai "While"-kohde.
Puuttuva EndIf	Odotettiin kohdetta EndIf , mutta löydettiin toinen End-kohde
Puuttuva EndFor	Odotettiin kohdetta EndFor , mutta löydettiin toinen End-kohde
Puuttuva EndWhile	Odotettiin kohdetta EndWhile , mutta löydettiin toinen End-kohde
Puuttuva EndLoop	Odotettiin kohdetta EndLoop , mutta löydettiin toinen End-kohde
Puuttuva EndTry	Odotettiin kohdetta EndTry , mutta löydettiin

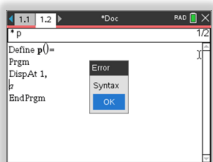
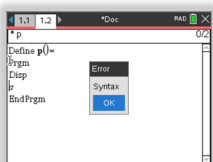
Virhetila	Uusi viesti
	toinen End-kohde
"Then" jätetty pois kohteen If <condition> jälkeen	Puuttuva If..Then
"Then" jätetty pois kohteen Elseif <condition> jälkeen	Then puuttuu kohdasta: Elseif.
Kun "Then", "Else", ja "Elseif" kohdattiin ohjauskohtien ulkopuolella	Then virheellinen kohtien ulkopuolella: If..Then..EndIf tai Try..EndTry
"Elseif" esiintyy kohdan "If..Then..EndIf" ulkopuolella	Elseif virheellinen kohdan ulkopuolella: If..Then..EndIf
"Then" esiintyy kohdan "If....EndIf" ulkopuolella	Then virheellinen kohdan ulkopuolella: If..EndIf

Syntaksivirheet

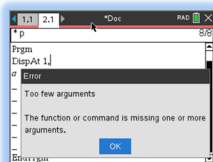
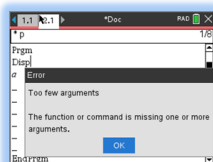
Jos tapauskomentoja, jotka odottavat yhtä tai useampaa argumenttia, kutsutaan epätäydellisellä argumenttiluettelolla, "Too few argument error" -virhe näytetään "syntax" -virheen sijaan.

Nykyinen käyttäytyminen	Uusi CX II -käyttäytyminen
	
	

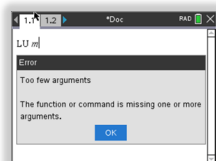
Nykyinen käyttäytyminen



Uusi CX II -käyttäytyminen



Huomaa: Kun epätäydellisen argumenttiluettelon jäljessä ei ole pilkkua, virheviesti on: "too few arguments". Tämä on sama kuin aiemmissa julkaisuissa.



Vakiot ja arvot

Seuraavassa taulukossa on lueteltu vakiot ja niiden arvot, jotka ovat käytettävissä suoritettaessa yksikköjen muunnoksia (NIST 2018). 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	8987551792.261 _m/_F
_Fc	Faradayn vakio	96485.33212 _coul/_mol
_g	Painovoiman kiihtyvyys	9.80665 _m/_s ²
_Gc	Gravitaativakio	6.6743E-11 _m ³ /_kg/_s ²
_h	Planckin vakio	6.62607015E-34 _J _s
_k	Boltzmannin vakio	1.380649E-23 _J/_°K
_μ0	Tyhjiön permeabiliteetti	1.25663706127E-6 _N/_A ²
_μb	Bohr magnetoni	9.274009994E-24 _J _m ² /_Wb
_Me	Elektronin lepomassa	9.10938371390E-31 _kg
_Mμ	Myonin massa	1.883531627E-28 _kg
_Mn	Neutronin lepomassa	1.67492750058E-27 _kg
_Mp	Protonin lepomassa	1.67262192595E-27 _kg
_Na	Avogadron vakio	6.02214076E23 _/_mol
_q	Elektronin varaus	1.602176634E-19 _coul
_R	Paine vakiona vesipotentiaalille	0.0831 L bars/mole K
_Rb	Bohrin säde	5.29177210903E-11 _m
_Rc	Moolinen kaasuvakio	8.314462618 _J/_mol/_°K
_Rdb	Rydbergin vakio	10973731.568160/_m
_Re	Elektronin säde	2.8179403262E-15 _m
_u	Atomimassa	1.66053906893E-27 _kg
_Vm	moolinen tilavuus	2.241396954E-2 _m ³ /_mol
_ε0	Tyhjiön permittiivisyys (vakio)	8.8541878188E-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, täältä.

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 $\text{If } a < b$ 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

Virhekoodi	Kuvaus
	Alarajan on oltava ylärajaa pienempi hakuväliä määritettäessä.
180	Keskeytys Näppäintä <code>esc</code> tai <code>on</code> on painettu pitkän laskutoimituksen tai ohjelman suorituksen aikana.
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	Datatyypin 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	Dimensiiovirhe 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 <code>rand(0)</code> ei kelpaa.
270	Muuttujan nimi jo käytössä
280	Else ja Elseif eivät kelpaa lohkon <code>If..EndIf</code> ulkopuolella
290	EndTry-komennosta puuttuu sopiva Else-lauseke
295	Liian monta iteraatiota
300	Odotettavissa 2- tai 3-elementtinen lista tai matriisi

Virhekoodi	Kuvaus
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 solve($3x^2-4,x$) ei kelpaa, koska ensimmäinen argumentti ei ole yhtälö.
345	Ristiriitaiset yksiköt
350	Indeksi alueen ulkopuolella
360	Epäsuora merkkijono ei kelpaa muuttujan nimeksi.
380	Määrittämätön Ans Edellinen laskutoimitus ei luonut Ans:n arvoa, tai aikaisempaa laskutoimitusta ei ole syötetty.
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

Virhekoodi	Kuvaus
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 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 lohkoista If..EndIf

Virhekoodi	Kuvaus
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	Argumenttivirhe Mediaani-mediaani-mallia ei voitu käyttää datasarjaan.
910	Syntaksivirhe
920	Tekstiä ei löydy
930	Liian vähän argumentteja Funktioista tai komennosta puuttuu yksi tai useampia argumentteja.
940	Liian monta argumenttia 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: • <code>sto →</code> • <code>:=</code> • Define määrittääksesi muuttujille arvot.

Virhekoodi	Kuvaus
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 • $:=$ • $\text{sto} \rightarrow$ määrittääksesi funktion.
1100	Ei-reaalinen laskutoimitus 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).
1110	Virheelliset rajat
1120	Ei etumerkin muutosta
1130	Argumentti ei voi olla lista tai matriisi
1140	Argumenttinvirhe

Virhekoodi	Kuvaus
	Ensimmäisen argumentin on oltava toisen argumentin sisällä oleva polynomilauseke. Jos toinen argumentti jätetään pois, ohjelmisto yrittää valita oletusarvon.
1150	Argumenttivirhe Ensimmäisten kahden argumentin on oltava kolmannen argumentin sisällä olevia polynomilausekkeita. Jos kolmas argumentti jätetään pois, ohjelmisto yrittää valita oletusarvon.
1160	Virheellinen kirjaston polkunimi Polkunimen on oltava muodossa <code>xxx\yyy</code> , jossa: • <code>xxx</code>-osassa voi olla 1-16 merkkiä. • <code>yyy</code>-osassa voi olla 1-15 merkkiä. Lisätietoja löydät ohjekirjan osasta Kirjasto.
1170	Kirjaston polkunimeä on käytetty väärin • Polkunimelle ei voi määrittää arvoa komennon Define, <code>:=</code> tai <code>sto</code> → 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.
1210	Virheellinen kirjaston pikavalintanimi. Tarkista, että nimi: • ei sisällä pistettä

Virhekoodi	Kuvaus
	• ei ala alaviivalla • ei ylitä 16 merkkiä • ei ole varattu nimi Lisätietoja löydät ohjekirjan osasta Kirjasto.
1220	Määrittelyjoukkovirhe: Funktiot tangentLine ja normalLine tukevat vain reaaliarvoisia funktioita.
1230	Määrittelyjoukkovirhe. Trigonometristen 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 domain() 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()**, täältä.

Varoituskoodi	Viesti
10000	Laskutoimitus voi antaa väärä ratkaisuja. Kokeile tarvittaessa graafisia menetelmiä tulosten varmistamiseksi.
10001	Yhtälön derivointi voi antaa väärän yhtälön.
10002	Kyseenalainen ratkaisu Kokeile tarvittaessa graafisia menetelmiä tulosten varmistamiseksi.
10003	Kyseenalainen tarkkuus Kokeile tarvittaessa graafisia menetelmiä tulosten varmistamiseksi.
10004	Laskutoimituksessa voi hävitä ratkaisuja. Kokeile tarvittaessa graafisia menetelmiä tulosten varmistamiseksi.
10005	cSolve voi määrittää lisää nollakohtia.
10006	Solve voi määrittää lisää nollia. Kokeile tarvittaessa graafisia menetelmiä tulosten varmistamiseksi.
10007	Lisää ratkaisuja voi olla olemassa. Yritä määrittämällä oikeat ala- ja ylärajat ja/tai arvaamalla. Esimerkkejä solve() käytöstä: • solve(Yhtälö, Muut=Arvaus) alaraja<Muut<yläraja • solve(Yhtälö, Muut) alaraja<Muut<yläraja • solve(Yhtälö, Muut=Arvaus) Kokeile tarvittaessa graafisia menetelmiä tulosten varmistamiseksi.
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 1:llä
10014	undef^0 korvattu 1:llä
10015	1^∞ tai 1^undef korvattu 1:llä
10016	1^undef korvattu 1:llä

Varoituskoodi	Viesti
10017	Laskualueen ylitys korvattu ∞ :lla tai $-\infty$:lla
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. Tulos ei ehkä ole voimassa kaikille mahdollisille parametriarvoille.
10022	Ratkaisu on mahdollinen, kun määritetään oikeat ala- ja ylärajat.
10023	Skalaari on kerrottu tunnistematriisilla.
10024	Tulos saatu käyttäen likiarvo-aritmetiikkaa.
10025	Ekvivalenssia ei voida varmistaa EXACT-tilassa.
10026	Ehto saatetaan sivuuttaa. Määritä ehto muodossa "\ " 'Muuttujan MathTestSymbol-vakio' tai näiden yhteismuoto, esim. 'x<3 and 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, niin saat tietoa takuun kestosta ja ehdoista tai tuotepalvelusta.

Rajoitettu takuu. Tämä takuu ei vaikuta lainmukaisiin oikeuksiisi.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Indeksi

!		–	
!, kertoma	239	_, yksikön merkki	249
"		_, yksikön nimi	249
", sekuntimuoto	248	 	
#		, (rajoittava operaattori)	252
#, epäsuora operaattori	245, 277	+	
%		+, yhteenlasku	228
%, prosentti	234	/	
&		/, jakolasku[*]	230
&, liitä	239	=	
*		≠, ei yhtä kuin[*]	235
*, kertolasku	229	=, on yhtä kuin	235
,		>	
, jaoton	249	>, suurempi kuin	237
, minuuttimuoto	248	Π	
–		Π, tulo[*]	242
–, vähennyslasku[*]	228	Σ	
.		Σ(), summa[*]	243
., pistetulo	233	ΣInt()	244
., piste-erotus	233	ΣPrn()	244
./, pisteosamäärä	233	√	
^., pistepotenssi	234	√, neliöjuuri[*]	242
+. pistesumma	232	∫	
:		∫, integraali[*]	240
:=, määritä	254	≤	
^		≤, pienempi tai yhtä suuri kuin	236
^, potenssi	231	≥	
^-1, käänteisluku	251	≥, suurempi tai yhtä suuri kuin	237

►		°	
►, muunna graadikulmaksi[Grad] ...	93	°, astekulmamuuoto[*]	247
►, muunna yksiköt[*]	250	°, astetta/minuuttia/sekuntia[*]	248
►, asteen regressio, QuartReg	153	#	
►approxFraction()	14	Ob, binaarinen indikaattori	255
►cos, näytä kulman kosini[cos]	32	#	
►Cylind, näytä vektorina sylinderikoordinaatistossa [Cylind]	45	Oh, heksadesimaalinen indikaattori .	255
►DD, näytä desimaalikulmana[DD] ..	49	#	
►Desimaali, näytä tulos desimaalilukuna[Desimaali]	49	2 otoksen F-testi	81
►DMS, näytä asteina/minuutteina/sekun teina[DMS]	58	#	
►exp, näytä e:n arvolla[exp]	68	2. asteen regressio, QuadReg	151
►Kantaluku2, näytä binaarisena [Kantaluku2]	18	#	
►Kantaluku10, näytä desimaalikonaislukuna [Kantaluku10]	19	3. asteen regressio, CubicReg	43
►Kantaluku16, näytä heksadesimaalilukuna [Kantaluku16]	20	#	
►Polaarinen, näytä polaarisenä vektorina[Polaarinen]	143	10^(), kymmenen potenssi	251
►Rad, muunna radiaanikulmaan ...	155	A	
►Rect, näytä suorakulmavektorina ..	158	abs(), itseisarvo	8
►sin, näytä kulman sini[sin]	180	alaraja, floor()	76
►Sphere, näytä pallonmuotoisena vektorina[Sphere]	190	alaviiva, _	249
⇒		alimatriisi, subMat()	195, 197
⇒, looginen seuraus[*]	238, 274	amortTbl(), lyhennystaulukko	8, 17
→		and, Boolean operator	9
→, tallenna	253	angle(), kulma	10
↔		ANOVA, yksisuuntainen varianssianalyysi	11
↔, looginen kaksoisseuraus[*]	238	ANOVA2way, kaksisuuntainen varianssianalyysi	11
©		ans, viimeinen vastaus	13
©, kommentti	254	approx(), likimääräinen	13
		approxRational()	14
		arccos()	14
		arccosh()	14
		arccot()	15
		arcoth()	15
		arccsc()	15
		arccsch()	15

arcLen(), kaaren pituus	15	comDenom(), yhteinen nimittäjä	28
arcsec()	15	completeSquare(), complete square	29
arcsech()	15	conj(), liittokompleksiluku	30
arcsin()	16	constructMat(), luo matriisi	30
arcsinh()	16	corrMat(), korrelaatiomatriisi	32
arctan()	16	cos(), kosini	32
arctanh()	16	cos ⁻¹ (), arkuskosini	34
arkuskosini, cos ⁻¹ ()	34	cosh(), hyperbolinen kosini	34
arkussini, sin ⁻¹ ()	182	cosh ⁻¹ (), hyperbolinen arkuskosini	35
arkustangentti, tan ⁻¹ ()	198	cot(), kotangentti	36
aseta		cot ⁻¹ (), arkuskotangentti	36
tila, setMode()	176	coth(), hyperbolinen kotangentti	36
asetukset, hae nykyiset	89	coth ⁻¹ (), hyperbolinen	
aste/minuutti/sekunti-muoto	248	arkuskotangentti	37
astekulmamuoto, -	247	count(), laske listan kohtien	
augment(), lisää/ketjuta	16	lukumäärä	37
avgRC(), keskimääräinen		countif(), laske listan yksiköiden	
muutosnopeus	16	lukumäärä ehdollisesti	37
B		cPolyRoots()	38
binaarinen		crossP(), ristitulo	39
indikaattori, 0b	255	csc(), kosekantti	39
näytä, 4Kantaluku2	18	csc ⁻¹ (), käänteiskosekantti	39
binomCdf()	21, 99	csch(), hyperbolinen kosekantti	40
binomPdf()	21	csch ⁻¹ (), käänteinen hyperbolinen	
Boolean operators		kosekantti	40
and	9	cSolve(), ratkaise kompleksiyhtälö	41
Boolean operaattorit		CubicReg, 3. asteen regressio	43
⇒	238, 274	cumulativeSum(), kumulatiivinen	
⇔	238	summa	44
ei	135	Cycle, cycle	45
ei-ja	128	cycle, Cycle	45
eikä	133	cZeros(), kompleksiset nollakohdat	46
tai	139	D	
xor	218	d(), ensimmäinen derivaatta	239
C		dbd(), päivämäärien väliset päivät	48
Cdf()	74	Define (Määritä)	49
ceiling(), yläraja	21	Define LibPriv (Määritä LibPriv)	51
centralDiff()	22	Define LibPub (Määritä LibPub)	51
cFactor(), kompleksilukutekijä	23	Define, määritä	49
char(), merkkijono	24	deltaList()	52
charPoly()	24	deltaTmpCnv()	52
clearAZ	27	DelVar, poista muuttuja	52
ClrErr, poista virhe	27	delVoid(), poista tyhjät elementit	52
colAugment	27	derivaatat	
colDim(), matriisin sarakemitat	28	ensimmäinen derivaatta, d()	239
colNorm(), matriisin sarakenormi	28	numeerinen derivaatta, nDeriv()	131

numeerinen derivaatta,		end while, EndWhile	217
nDerivative()	130	EndTry, kokeilun loppu	207
derivaatta tai n:s derivaatta		EndWhile, end while	217
malli	6	ensimmäinen derivaatta	
desimaali		malli	6
kokonaislukunäyttö,		EOS (yhtälökäyttöjärjestelmä)	276
4Kantaluku10	19	epäsuora operaattori (#)	277
kulmanäyttö, ►DD	49	epäsuora operaattori, #	245
deSolve(), ratkaisu	53	etumerkki, sign()	179
det(), matriisin determinantti	55	euler(), Euler function	65
diag(), diagonaalimatriisi	55	exact(), täsmällinen	67
dim(), mitta	56	Exit, lopeta	67
Disp, näytä tiedot	56, 171	exp(), e korotettuna potenssiin	68
DispAt	56	exp►list(), lauseke listaksi	69
dominantTerm(), dominanttitermi ..	60	expand(), lavenna	69
dominanttitermi, dominantTerm() ..	60	expr(), merkkijono lausekkeeksi	71, 115
dotP(), pistetulo	61	ExpReg, eksponentiaalinen regressio ..	71

E

e eksponentti	
malli	2
e korotettuna potenssiin, e^()	62, 68
E, eksponentti	246
e, näytä lauseke e:n arvolla	68
e^(), e korotettuna potenssiin	62
efektiivinen korko, eff()	62
eff(), muunna nimelliskorko	
efektiiviseksi koroksi	62
ei yhtä kuin, ≠	235
ei, Boolean operaattori	135
eigVc(), ominaisvektori	63
eigVl(), ominaisarvo	63
ei-ja, Boolean operaattori	128
eikä, Boolean operaattori	133
eksklusio "I"-operaattorilla	252
eksponentiaalinen regressio, ExpReg	
eksponentti	
malli	1
eksponentti, E	246
else if, Elseif	64
Elseif, else if	64
end	
for, EndFor	78
funktio, EndFunc	82
ohjelma, EndPrgm	148
silmutta, EndLoop	118
try, EndTry	207
while, EndWhile	217

F

factor(), tekijä	72
Fill, matriisin täyttö	75
FiveNumSummary	75
floor(), alaraja	76
fMax(), funktion maksimi	76
fMin(), funktion minimi	77
For	78
For, for	78
for, For	78
format(), merkkijonon muoto	78
fpart(), funktion osa	79
freqTable()	80
frequency()	80
Frobeniusin normi, norm()	134
Func, funktio	82
Func, ohjelmafunktio	82
funktion loppu, EndFunc	82
funktiot	
käyttäjän määrittämät	49
maksimi, fMax()	76
minimi, fMin()	77
ohjelmafunktio, Func	82
osa, fpart()	79
funktiot ja muuttujat	
kopiointi	31

G

g, graadit	246
gcd(), suurin yhteinen jakaja	83

geomCdf()	83	integraali, %o	240
geomPdf()	83	interpolate(), interpoloi	98
Get	84, 266	invF()	98
getDenom(), hae/laske nimittäjä	85	invNorm(), käänteinen	
getKey()	85	kumulatiivinen	
getLangInfo(), hae/anna kielitiedot	88	normaalijakauma)	100
getLockInfo(), testaa muuttujan tai		invf()	100
muuttujaryhmän		Invx ² ()	98
lukitustilan	89	iPart(), kokonaisosa	100
getMode(), hae tila-asetukset	89	irr(), sisäinen palautuksen määrä	
getNum(), hae/laske luku	91	sisäinen palautuksen määrä , irr(	
GetStr	91	)	100
getType(), get type of variable	91	isPrime(), jaottoman testi	101
getVarInfo(), hae/laske muuttujien		isVoid(), tyhjän testi	101
tiedot	92	itseisarvo	
Goto, siirry	93	malli	4
graadimuoto, g	246		
		J	
H		jäännös, remain()	161
hae/laske		jakakaumafunktiot	
luku, getNum()	91	binomCdf()	99
muuttujien tiedot, getVarInfo()	88, 92	invNorm()	100
nimittäjä, getDenom()	85	invf()	100
heksadesimaali		jakaumafunktiot	
näytä, 4Kantaluku16	20	binomCdf()	21
heksadesimaalinen		binomPdf()	21
indikaattori, Oh	255	Invx ² ()	98
hyperbolinen		normCdf()	134
areatangenti, tanh ⁻¹ ()	200	normPdf()	134
arkuskosini, cosh ⁻¹ ()	35	poissCdf()	143
arkussini, sinh ⁻¹ ()	183	poissPdf()	143
kosini, cosh()	34	tCdf()	201
sini, sinh()	183	tPdf()	206
tangenti, tanh()	199	χ ² 2way()	25
		χ ² Cdf()	25
I		χ ² GOF()	26
identity(), identtinen matriisi	93	χ ² Pdf()	26
identtinen matriisi, identity()	93	jakolasku, P	230
If, jos	94	jaoton,	249
iffn()	95	jaottoman luvun testi, isPrime()	101
imag(), imaginaarinen osa	96	jos käytettävissäsi on	252
imaginaarinen osa, imag()	96	jos, EndIf	94
ImpDif(), implisiittinen derivaatta	96	jos, If	94
implisiittinen derivaatta, Impdif()	96		
Input, syöte	96	K	
inString(), merkkijonon sisällä	97	käänteinen kumulatiivinen	
int(), kokonaisluku	97	normaalijakauma (invNorm(	100
intDiv(), kokonaisluvun jakaminen	97		

)		laske listan kohtien lukumäärä,	
käänteinen, $\wedge^{-1}$	251	count()	37
käänteisluku, $\wedge^{-1}$	251	laske listan yksiköiden lukumäärä	
kaaren pituus, arclen()	15	ehdollisesti, countif()	37
kahden muuttujan tulokset, TwoVar	211	laske päivämäärien väliset päivät,	
käyttäjän määrittämät funktiot	49	dbd()	48
käyttäjän määrittämät funktiot ja		laske polynomi, polyEval()	145
ohjelmat	51	lausekkeet	
kertolasku, *	229	lauseke listaksi, exp $\rightarrow$ list()	69
kertoma, !	239	merkkijono lausekkeeksi, expr()	71, 115
keskiarvo, mean()	120	lavenna, expand()	69
keskihajonta, stdDev()	193-194, 214	Lbl, tunnus	102
keskimääräinen muutosnopeus,		lcm, pienin yhteinen jaettava	102
avgRC()	16	left(), vasen	102
keskimäinen merkkijono, mid()	123	LibPriv	51
kieli		LibPub	51
hae kielitiedot	88	libShortcut(), luo pikavalinnat	
kierrä, rotate()	166	kirjasto-objekteihin	103
kirjasto		liitä, &	239
luo pikavalinnat objekteihin	103	likimääräinen, approx()	13
kokonaisluku, int()	97	limit() tai lim(), raja-arvo	104
kokonaisluvun jakaminen, intDiv()	97	lineaarinen regressio, LinRegAx	106
kokonaisosa, iPart()	100	lineaarinen regressio, LinRegBx	105, 107
kombinaatiot, nCr()	129	LinRegBx, lineaarinen regressio	105
kommentti, ©	254	LinRegMx, lineaarinen regressio	106
kompleksi		LinRegIntervals, lineaarinen	
liitto, conj()	30	regressio	107
nollakohdat, cZeros()	46	LinRegtTest	109
ratkaise, cSolve()	41	linSolve()	110
tekijä, cFactor()	23	lisää/ketjuta, augment()	16
kopioi muuttuja tai funktio, CopyVar	31	list $\rightarrow$ mat(), listasta matriisiksi	111
korkoerien summa	244	lista, laske kohtien lukumäärä	37
korrelaatiomatriisi, corrMat()	32	lista, laske yksiköiden lukumäärä	
korvaaminen "I"-operaattorilla	252	ehdollisesti	37
kosini		listasta matriisiksi, list $\rightarrow$ mat()	111
näytä kulman	32	listat	
kosini, cos()	32	erotukset listassa, @list()	111
kotangenti, cot()	36	erotus, @list()	111
kulma, angle()	10	keskimäinen merkkijono, mid()	
kumulatiivinen summa,		)	123
cumulativeSum()	44	kumulatiivinen summa,	
kun, when()	217	cumulativeSum()	44
kymmenen potenssi, 10 $^{\wedge}()$	251	lajittele laskevaan järjestykseen,	
		SortD	190
		lajittele nousevaan	
		järjestykseen, SortA	189
		lauseke listaksi, exp $\rightarrow$ list()	69
		lisää/ketjuta, augment()	16

listasta matriisiksi, list►mat() ..	111	maksimi, max()	120
maksimi, max()	120	mallit	
matriisi listaksi, mat►list()	119	derivaatta tai n:s derivaatta	6
minimi, min()	123	e eksponentti	2
pistetulo, dotP()	61	eksponentti	1
ristitulo, crossP()	39	ensimmäinen derivaatta	6
tulo, product()	149	itseisarvo	4
tyhjät elementit	272	Logaritmi	2
uusi, newList()	130	määrätty integraali	7
yhteenlasku, sum()	195-196	määrittämätön integraali	7
ln(), luonnollinen logaritmi	112	matriisi (1 × 2)	4
LnReg, logaritminen regressio	112	matriisi (2 × 1)	5
Local, paikallinen muuttuja	114	matriisi (2 × 2)	4
Lock, lukitse muuttuja tai muuttujaryhmä	114	matriisi (m × n)	5
Logaritmi		murtoluku	1
malli	2	n:s juuri	2
logaritminen regressio, LnReg	112	neliöjuuri	1
logaritmit	112	paloittain määritelty funktio (2- osainen)	2
Logistic, logistinen regressio	116	paloittain määritelty funktio (N- osainen)	3
LogisticD, logistinen regressio	117	raja-arvo	7
logistinen regressio, Logistic	116	sum (G)	5
logistinen regressio, LogisticD	117	toinen derivaatta	6
looginen kaksoisseuraus, ⇔	238	tulo (P)	6
looginen seuraus, ⇒	238, 274	yhtälöpari (2 yhtälöä)	3
Loop, silmukka	118	yhtälöryhmä (N yhtälöä)	4
lopeta, Exit	67	mat►list(), matriisi listaksi	119
loppu	94	matriisi (1 × 2)	
loppu jos, EndIf	94	malli	4
LU, matriisin ala-ylä-dekomponointi	119	matriisi (2 × 1)	
lukujono		malli	5
kierrä, rotate()	166	matriisi (2 × 2)	
oikea, right()	164	malli	4
lukujonot		matriisi (m × n)	
oikea, right()	98	malli	5
luo matriisi, constructMat()	30	matriisi listaksi, mat►list()	119
luonnollinen logaritmi, ln()	112	matriisit	
lyhennystaulukko, amortTbl()	8, 17	ala-ylä-dekomponointi, LU	119
M		alimatriisi, subMat()	195, 197
määrätty integraali		determinantti, det()	55
malli	7	diagonaali, diag()	55
määrittä, Define	49	identtinen, identity()	93
määrittämätön integraali		kumulatiivinen summa,	
malli	7	cumulativeSum()	44
määrittäminen		lisää/ketjuta, augment()	16
julkinen funktio tai ohjelma	51	listasta matriisiksi, list►mat() ..	111
yksityinen funktio tai ohjelma ..	51	maksimi, max()	120
		matriisi listaksi, mat►list()	119

minimi, min()	123	keskimmäinen merkkijono, mid()	123
mitta, dim()	56	lauseke merkkijonoksi, string()	195
ominaisarvo, eigVl()	63	liitä, &	239
ominaisvektori, eigVc()	63	merkkijono lausekkeeksi, expr()	71, 115
piste-erotus, -	233	merkkijono, char()	24
pisteosamäärä, ./	233	merkkikoodi, ord()	141
pistepotenssi, .^	234	muoto, format()	78
pistesumma, +	232	muotoilu	78
pistetulo, *	233	siirrä, shift()	177
QR-hajottaminen, QR	150	sisällä, inString()	97
rivi-echelon-muoto, ref()	159	vasen, left()	102
rivien kertominen ja yhteenlasku, mRowAdd()	125	mid(), keskimmäinen merkkijono	123
rivien yhteenlasku, rowAdd()	168	min(), minimi	123
rivikoko, rowDim()	168	minimi, min()	123
rivinormi, rowNorm()	168	minuuttimuoto,	248
rivinvaihto, rowSwap()	169	mirr(), modifioitu sisäinen korkokanta	124
rivioperaatio, mRow()	125	mitta, dim()	56
sarakemitat, colDim()	28	mod(), modulo	125
sarakenormi, colNorm()	28	modifioitu sisäinen korkokanta, mirr()	124
satunnainen, randMat()	156	modulo, mod()	125
täyttäminen, Fill	75	Moninkertaisen lineaarisen regression t-testi	127
transponoi, T	197	mRow(), matriisin rivioperaatio	125
tulo, product()	149	mRowAdd(), matriisin rivien kertominen ja yhteenlasku	125
uusi, newMat()	130	MultReg	125
yhteenlasku, sum()	195-196	MultRegIntervals()	126
max(), maksimi	120	MultRegTests()	127
mean(), keskiarvo	120	murtoluvut	
mediaani, median()	121	malli	1
mediaani-mediaani-suoran regressio, MedMed	121	propFrac	149
median(), mediaani	121	muunna	
MedMed, mediaani-mediaani-suoran regressio	121	4Grad	93
merkit		yksiköt	250
merkkijono, char()	24	muuntaminen	
numeerinen koodi, ord()	141	►Rad	155
merkkijono		muutoin, Else	94
mitta, dim()	56	muuttuja	
pituus	56	nimen luominen merkkijonosta	277
merkkijono, char()	24	muuttajat	
merkkijonon muoto, format()	78	paikallinen, Local	114
merkkijonon pituus	56	poista kaikki yksikirjaimiset	27
merkkijonon sisällä, inString()	97	poista, DelVar	52
merkkijonot		muuttajat ja funktiot	
epäsuora operaattori, #	245	kopioiminen	31
käyttö muuttujanimien luomisessa	277	muuttajat, lukitseminen ja	89, 114, 214

vapauttaminen	114
muuttujien ja muuttujaryhmien lukitseminen	214
muuttujien ja muuttujaryhmien vapauttaminen	

N

n:s juuri	
malli	2
näppäimistön pikavalinnat	274
näytä	
asteina/minuutteina/sekunteina,	
►DMS	58
binaarisena, 4Kantaluku2	18
desimaalikokonaislukuna,	
4Kantaluku10	19
desimaalikulmana, ►DD	49
heksadesimaalilukuna,	
4Kantaluku16	20
pallonmuotoinen vektori,	
►Sphere	190
polaarinen vektori, 4Polaarinen	143
suorakulmavektorina, ►Rect	158
vektorina	
sylinterikoordinaatistos	
sa, 4Cylind	45
näytä tiedot, Disp	56, 171
näyttö	
asteina/minuutteina/sekun	
teina, ►DMS	58
näyttö pallonmuotoisena vektorina,	
►Sphere	190
näyttö vektorina	
sylinterikoordinaatistossa,	
►Cylind	45
nCr(), kombinaatiot	129
nDerivative(), numeerinen	
derivaatta	130
negatio, negatiivisten lukujen	
syöttäminen	277
neliöjuuri	
malli	1
neliöjuuri, #()	191, 242
nettonykyarvo, npv()	136
newList(), uusi lista	130
newMat(), uusi matriisi	130
nfMax(), numeerisen funktion	131

maksimi	
nfMin(), numeerisen funktion	
minimi	131
nimelliskorko, nom()	132
nimittäjä	28
nInt(), numeerinen integraali	131
nollakohdat, zeroes()	219
nom), muuta efektiivinen korko	
nimelliskoroksi	132
norm(), Frobeniusin normi	134
normaalijakauman todennäköisyys,	
normCdf()	134
normaalisuora, normalLine()	134
normalLine()	134
normCdf()	134
normPdf()	134
nPr(), permutaatiot	136
npv(), nettonykyarvo	136
nSolve(), numeerinen ratkaisu	137
numeerinen	
derivaatta, nDeriv()	131
derivaatta, nDerivative()	130
integraali, nInt()	131
ratkaisu, nSolve()	137

O

objektit	
luo pikavalinnat kirjastoon	103
Odota-komento	215
ohita virhe, PassErr	142
ohjelmat	
julkisen kirjaston määrittäminen	51
yksityisen kirjaston	
määrittäminen	51
ohjelmat ja ohjelmointi	
kokeilun loppu, EndTry	207
lopeta ohjelma, EndPrgm	148
näytä I/O-näyttö, Disp	56, 171
poista virhe, ClrErr	27
try, Try	207
ohjelmointi	
määritä ohjelma, Prgm	148
näytä tiedot, Disp	56, 171
ohita virhe, PassErr	142
oikea, right()	98, 164
ominaisarvo, eigVl()	63
ominaisvektori, eigVc()	63
on yhtä kuin, =	235

OneVar, yhden muuttujan tilastot ..	138	polyGcd()	145-146
operaattorit		polynomit	
sievennysjärjestys	276	laske, polyEval()	145
ord(), numeerinen merkkikoodi	141	satunnainen, randPoly()	157
P		PolyRoots()	147
P►Rx(), x:n suorakulmakoordinaatti	141	potenssi, ^	231
P►Ry(), y:n suorakulmakoordinaatti	141	potenssiregressio,	
pääoman maksuerien summa	244	PowerReg	147, 161, 163, 203
paikallinen muuttuja, Local	114	PowerReg, potenssiregressio	147
paikallinen, Local	114	Prgm, määritä ohjelma	148
päivämäärien väliset päivät, dbd() ..	48	prodSeq()	149
Palaa, return	164	product(), tulo	149
paloittain määritelty funktio (2- osainen)		propFrac, varsinainen murtoluku ...	149
malli	2	prosentti, %	234
paloittain määritelty funktio (N- osainen)		pyöristä, round()	167
malli	3	PyydäMerkkij	163
PassErr, ohita virhe	142	Pyyntö	161
Pdf()	80	Q	
permutaatiot, nPr()	136	QR, QR-hajottaminen (tekijöihin jako)	150
piecewise()	142	QR-hajottaminen, QR	150
pienempi tai yhtä suuri kuin, {	236	QuadReg, 2. asteen regressio	151
pienin yhteinen jaettava, lcm	102	QuartReg, 4. asteen regressio	153
piirrä	261-263	R	
pikavalinnat, näppäimistö	274	R, radiaani	246
piste		R►Pr(), polaarinen koordinaatti	154
erotus, -	233	R►Pθ(), polaarinen koordinaatti	154
osamäärä, ./	233	radiaani, R	246
potenssi, ^	234	rahan aika-arvo, Korko	210
summa, .+	232	rahan aika-arvo, maksuerä	210
tulo, *	233	rahan aika-arvo, maksuerien lukumäärä	210
tulo, dotP()	61	rahan aika-arvo, nykyarvo	211
poissCdf()	143	rahan aika-arvo, Tuleva arvo	210
poissPdf()	143	raja-arvo	
poista		lim()	104
poista, ClrErr	27	limit()	104
tyhjät elementit listasta	52	malli	7
poistaminen		rajoittava operaattori "I"	252
muuttuja, DelVar	52	rajoittava operaattori,	
polaarinen		laskemisjärjestys	276
koordinaatti, R►Pr()	154	rand(), satunnaisluku	155
koordinaatti, R►Pθ()	154	randBin, satunnaisluku	155
vektorinäyttö, 4Polaarinen	143	randInt(), satunnainen kokonaisluku	155
polyCoef()	144	randMat(), satunnaismatriisi	156
polyDegree()	145		
polyEval(), laske polynomi	145		

randNorm(), satunnainen		S	
normaalijakauma	156		
randPoly(), satunnaispolynomi	157	samanaikaiset yhtälöt, simult()	179
randSamp()	157	sarja, series()	174
RandSeed, satunnaisluvun		satunnainen	
siemenluku	157	matriisi, randMat()	156
ratkaise, solve()	185	normaalijakauma, randNorm()	156
ratkaisu, deSolve()	53	polynomi, randPoly()	157
reaali, real()	157	siemenluku, RandSeed	157
real(), reaali	157	satunnaisotos	157
ref(), rivi-echelon-muoto	159	sec(), sekantti	169
RefreshProbeVars	160	sec ⁻¹ (), käänteissekantti	170
regressiot		sech(), hyperbolinen sekantti	171
2. asteen, QuadReg	151	sech ⁻¹ (), käänteinen hyperbolinen	
3. asteen, CubicReg	43	sekantti	171
4. asteen, QuartReg	153	sekaluvut, käyttäen funktiota	
eksponentiaalinen, ExpReg	71	propFrac()	149
lineaarinen regressio, LinRegAx	106	sekuntimuoto, "	248
lineaarinen regressio, LinRegBx	105, 107	sekvenssi, seq()	172
logaritminen, LnReg	112	seq(), sekvenssi	172
Logistinen	116	seqGen()	172
logistinen, Logistic	117	seqn()	173
mediaani-mediaani-suora,		sequence, seq()	172-173
MedMed	121	series(), sarja	174
MultReg	125	setMode(), aseta tila	176
potenssiregressio,		shift(), siirrä	177
PowerReg	147, 161, 163, 203	sievennetty rivi-echelon-muoto, rref	
sinimuotoinen, SinReg	184	()	169
remain(), jäännös	161	sievennysjärjestys	276
return, Palaa	164	sign(), etumerkki	179
right(), oikea	164	siirry, Goto	93
right, right()	29, 65, 216	silmukan loppu, EndLoop	118
ristitulo, crossP()	39	silmukka, Loop	118
rivi-echelon-muoto, ref()	159	simult(), samanaikaiset yhtälöt	179
rk23(), Runge Kutta -funktio	164	sin(), sini	181
rotate(), kierrä	166	sin ⁻¹ (), arkussini	182
round(), pyöristä	167	sinh(), hyperbolinen sini	183
rowAdd(), matriisin rivien		sinh ⁻¹ (), hyperbolinen arkussini	183
yhteenlasku	168	sini	
rowDim(), matriisin rivikoko	168	näytä kulman	180
rowNorm(), matriisin rivinormi	168	sini, sin()	181
rowSwap(), matriisin rivinvaihto	169	siniregressio, SinReg	184
rref(), sievennetty rivi-echelon-		SinReg, siniregressio	184
muoto	169	solve(), ratkaise	185
ryhmät, lukitseminen ja		SortA, lajittele nousevaan	
vapauttaminen	114, 214	järjestykseen	189
ryhmät, lukitustilan testaaminen	89	SortD, lajittele laskevaan	
		järjestykseen	190
		sqrt(), neliöjuuri	191

stat.results	191	tCdf(), studentint- todennäköisyysjakauma ...	201
stat.values	193	tCollect(), trigonometrinen keruu ..	201
stdDevPop(), perusjoukon keskihajonta	193	tekijä, factor()	72
stdDevSamp(), otoksen keskihajonta	194	Teksti-komento	203
Stop-komento	194	Test_2S, 2 otoksen F-testi	81
string(), lauseke merkkijonoksi	195	tExpand(), trigonometrinen laajennus	202
strings right, right()	29, 65, 216	tila-asetukset, getMode()	89
studentint-todennäköisyysjakauma, tCdf()	201	tilastot kahden muuttujan tulokset, TwoVar	211
studentint-todennäköisyysstiheys, tPdf()	206	kertoma, !	239
subMat(), alimatriisi	195, 197	keskiarvo, mean()	120
sum(), yhteenlasku	195	keskihajonta, stdDev()	193-194, 214
sumlf()	196	kombinaatiot, nCr()	129
summa (G) malli	5	mediaani, median()	121
summa, S()	243	permutaatiot, nPr()	136
sumSeq()	197	satunnainen normaalijakauma, randNorm()	156
suorakulmavektorinäyttö, ►Rect	158	satunnainen siemenluku, RandSeed	157
suurempi kuin, >	237	varianssi, variance()	215
suurempi tai yhtä suuri kuin, 	237	yhden muuttujan tilastot, OneVar	138
suurin yhteinen jakaja, gcd()	83	tilat asetus, setMode()	176
syöte, Input	96	TInterval, t-luottamusväli	203
		TInterval_2Samp, kahdenotoksen t- luottamusvälin	204
T		tmpCnv()	205
T, transponoi	197	todennäköisyysstiheys, normPdf() ..	134
tai (Boolean), tai	139	toinen derivaatta malli	6
tai, Boolean operaattori	139	tPdf(), studentint- todennäköisyysstiheys	206
tallentaminen symboli, &	253-254	trace()	207
talouselaskentafunktiot, tvml()	210	transponoi, T	197
talouselaskentafunktiot, tvml()	210	trigonometrinen keruu, tCollect() ..	201
talouselaskentafunktiot, tvmlN()	210	trigonometrinen laajennus, tExpand()	202
talouselaskentafunktiot, tvmlPmt() ..	210	Try, virheenkäsitelykomento	207
talouselaskentafunktiot, tvmlPV()	211	tTest, t-testi	208
tan(), tangentti	197	tTest_2Samp, kahden otoksen t-testi	209
tan ⁻¹ (), arkustangentti	198	t-testi, tTest	208
tangentLine()	199	tulo (P) malli	6
tangentti, tan()	197	tulo, P()	242
tangenttisuora, tangentLine()	199	tulo, product()	149
tanh(), hyperbolinen tangentti	199		
tanh ⁻¹ (), hyperbolinen areatangentti	200		
täsmällinen, exact()	67		
taylor(), Taylorin polynomi	201		
Taylorin polynomi, Taylor()	201		
täyttö	264-265		

tulokset, tilastot	191	vektorit	
tulosarvot, tilastot	193	näyttö vektorina	
tunnus, Lbl	102	sylinterikoordinaattistos	
TVM-argumentit	211	sa, ►Cylind	45
TVM-funktioiden argumentit	211	pistetulo, dotP()	61
tvmFV()	210	ristitulo, crossP()	39
tvmI()	210	yksikkö, unitV()	213
tvmN()	210	virheet ja vianmääritys	
tvmPmt()	210	ohita virhe, PassErr	142
tvmPV()	211	poista virhe, ClrErr	27
TwoVar, kahden muuttujan tulokset	211	virhekoodit ja -viestit	290
tyhjä, testi	101		
tyhjän testi, isVoid()	101		
tyhjä elementit	272		
tyhjä elementit, poista	52		
Tyhjennä	260		
		W	
		warnCodes(), Warning codes	216
		when(), kun	217
		While, while	217
		while, While	217
		X	
unitV(), yksikkövektori	213	x:n suorakulmakoordinaatti, P►Rx()	141
unLock, vapauta muuttuja tai		x ² , neliö	232
muuttujaryhmä	214	XNOR	238
uusi		xor, Boolean eksklusiivinen tai (or) ...	218
lista, newList()	130		
matriisi, newMat()	130		
		Y	
		y:n suorakulmakoordinaatti, P►Ry()	141
		yhden muuttujan tilastot, OneVar ..	138
		Yhtälökäyttöjärjestelmä (EOS)	276
		yhtälöpari (2 yhtälöä)	
		malli	3
		yhtälöryhmä (N yhtälöä)	
		malli	4
		yhteenlasku, +	228
		yhteenlasku, sum()	195
		yhteinen nimittäjä, comDenom() ...	28
		yksikkövektori, unitV()	213
		yksiköt	
		muunna	250
		yläraja, ceiling()	21-22, 38
		Z	
		zeroes(), nollakohdat	219
		zInterval, z-luottamusväli	221
		zInterval_1Prop, yhden osuuden z-	
		luottamusväli	222
		zInterval_2Prop, kahden osuuden z-	223

luottamusväli	
zInterval_2Samp, kahden näytteen	
z-luottamusväli	223
zTest	224
zTest_1Prop, yhden osuuden z-testi	225
zTest_2Prop, kahden osuuden z-testi	225
zTest_2Samp, kahden otoksen z-	
testi	226

Δ

Δlist(), listaerotus	111
ΔtmpCnv() [tmpCnv]	205

X

χ^2 2way	25
χ^2 Cdf()	25
χ^2 GOF	26
χ^2 Pdf()	26