TI-Nspire™ CX
Sovelluksen käsikirja

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

Tärkeitä tietoja

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

© 2006 - 2019 Texas Instruments Incorporated

Sisällys

Lausekemallit **1**

Luettelo aakkosjärjestyksessä **7**

A 7
B 15
C 20
D 36
E 45
F 53
G 61
I 71
L 79
M 94
N 103
O 111
P 114
Q 121
R 124
S 139
T 159
U 172
V 173
W 174
X 176
Z 177

Symbolit **183**

TI-Nspire™ CX II – Piirtokomennot **207**

Kuvaajien ohjelmointi 207
Kuvaajanäyttö 207
Oletusnäkymä ja asetukset 208
Kuvaajanäytön virheviestit 209
Virheelliset komennot kuvaajatilassa 209
C 211
D 212
F 215
G 217
P 218
J 220
I 222

Tyhjät elementit **223**

Matemaattisten lausekkeiden syöttäminen pikavalintojen avulla **225**

EOS-järjestelmän (yhtälökäyttöjärjestelmä) hierarkia **226**

TI-Nspire CX II – TI-Basic-ohjelmointiominaisuudet **228**

Automaattinen sisennys Ohjelmointieditorissa 228
TI-Basicin virheviestien parannukset 228

Vakiot ja arvot **231**

Virhekoodit ja viestit **232**

Varoituskoodit ja -viestit **239**

Yleistä **241**

Online-tuki 241
Ota yhteyttä TI-tukeen 241
Huolto- ja takuutiedot 241

Indeksi **242**

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 185.

Esimerkki:

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

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 186.

Esimerkki:

$$2^3 \qquad 8$$

Neliöjuurimalli

[ctrl] [x²] **painikkeet**

Huomaa: Katso myös **√() (neliöjuuri)**, sivu 196.

Esimerkki:

$$\sqrt{4} \qquad 2$$

$$\sqrt{\{9,16,4\}} \qquad \{3,4,2\}$$

N:s juuri -malli

ctrl ^ **painikkeet**

$\sqrt[\square]{\square}$

Huomaa: Katso myös **root()**, sivu 136.

Esimerkki:

$\sqrt[3]{8}$	2
$\sqrt[3]{\{8,27,15\}}$	$\{2,3,2.46621\}$

e eksponenttimalli

e^x **painikkeet**

$e^{\square}$

e-kantainen eksponenttifunktio korotettuna potenssiin

Huomaa: Katso myös **e^()**, sivu 46.

Esimerkki:

e^1	2.71828182846

Logaritmimalli

ctrl 10^x **painike**

$\log_{\square}(\square)$

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

Huomaa: Katso myös **log()**, sivu 90.

Esimerkki:

$\log_4(2.)$	0.5

Paloittain määritellyn funktion malli (2-osainen)

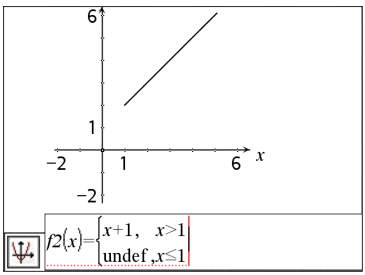
Katalogi >

$\begin{cases}\square, & \square \\ \square, & \square\end{cases}$

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

Huomaa: Katso myös **piecewise()**, sivu 116.

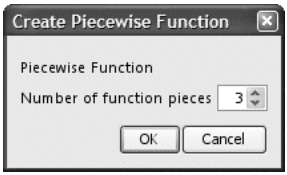
Esimerkki:



Paloittain määritellyn funktion malli (N-osainen)

Katalogi >

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



Huomaa: Katso myös **piecewise()**, sivu 116.

Esimerkki:

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

Yhtälöparin malli

Katalogi >

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

Huomaa: Katso myös **system()**, sivu 159.

Esimerkki:

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

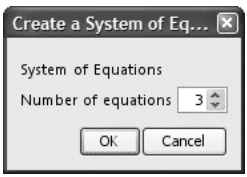
$$\text{solve}\left(\begin{cases} y=x^2-2 \\ x+2\cdot y=-1 \end{cases}, x, y\right)$$

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

N-osaisen yhtälöryhmän malli

Katalogi >

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



Huomaa: Katso myös **system()**, sivu 159.

Esimerkki:

Katso yhtälöparin (2 yhtälöä) mallin esimerkki.

Itseisarvon malli

Katalogi >

|▯| **Huomaa:** Katso myös **abs()**, sivu 7.

Esimerkki:

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

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°15'10" 0.528011

Matriisimalli (2 x 2)

Katalogi >

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

Luo 2 x 2 -matriisin.

Esimerkki:

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 5$ $\begin{bmatrix} 5 & 10 \\ 15 & 20 \end{bmatrix}$

Matriisimalli (1 x 2)

Katalogi >

$\begin{bmatrix} ▯ & ▯ \end{bmatrix}$.

Esimerkki:

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

Matriisimalli (2 x 1)

Katalogi >

$\begin{bmatrix} ▯ \\ ▯ \end{bmatrix}$

Esimerkki:

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

Matriisimalli (m x n)

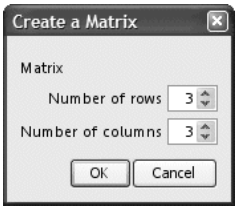
Katalogi >

Malli tulee näkyviin määritettyäsi rivien ja sarakkeiden lukumäärän syöttöruutuun.

Esimerkki:

Matriisimalli (m x n)

Katalogi >



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

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

Summan malli (Σ)

Katalogi >

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

Esimerkki:

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

Huomaa: Katso myös **Σ() (sumSeq)**, sivu 197.

Tulon malli (Π)

Katalogi >

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

Esimerkki:

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

Huomaa: Katso myös **Π() (prodSeq)**, sivu 196.

Ensimmäisen derivaatan malli

Katalogi >

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

Esimerkki:

$$\frac{d}{dx}(|x|)|x=0 \qquad \text{undef}$$

Ensimmäisen derivaatan malli

Katalogi >

Ensimmäisen derivaatan mallia voi käyttää myös laskettaessa numeerisesti ensimmäinen derivaatta pisteessä käyttäen automaattisia derivointimenetelmiä.

Huomaa: Katso myös **d() (derivaatta)**, sivu 194.

Toisen derivaatan malli

Katalogi >

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

Toisen derivaatan mallia voi käyttää myös laskettaessa numeerisesti toinen derivaatta pisteessä käyttäen automaattisia derivointimenetelmiä.

Huomaa: Katso myös **d() (derivaatta)**, sivu 194.

Esimerkki:

$$\frac{d^2}{dx^2}\left(x^3\right)\Big|_{x=3} \qquad 18$$

Määrätyn integraalin malli

Katalogi >

$$\int_{\square}^{\square} \square \, d\square$$

Määrätyn integraalin mallia voi käyttää myös laskettaessa numeerisesti määrätty integraali käyttäen samaa menetelmää kuin nInt().

Huomaa: Katso myös nInt(), sivu 107.

Esimerkki:

$$\int_{0}^{10} x^2 \, dx \qquad 333.333$$

Luettelo aakkosjärjestyksessä

Komennot, joiden nimiä ei voi järjestää aakkosjärjestykseen (esimerkiksi +, ! ja >), on esitetty tämän kappaleen lopussa alkaen sivulta (sivu 183). Ellei toisin ole mainittu, kaikki tämän kappaleen esimerkit on suoritettu laskimen oletustilassa, eikä mitään muuttujia ole määritetty.

A

abs() Katalogi >

abs(*Arvo1***)⇒***arvo*
abs(*Lista1***)⇒***lista*
abs(*Matriisi1***)⇒***matriisi*

Laskee argumentin itseisarvon.

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

Jos argumentti on kompleksiluku, määrittää luvun moduulin.

Huomaa: Kaikkia määrittämättömiä muuttujia käsitellään reaalimuuttujina.

$\left|\left\{\frac{\pi}{2},\frac{-\pi}{3}\right\}\right|$ $\{1.5708,1.0472\}$

$|2-3\cdot i|$ 3.60555

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 170.

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

amortTbl(12,60,10,5000,,,12,12)

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

oletusarvoksi tulee FV=0.

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

pyörArvo määrittää pyöristyksessä käytettävien desimaalien määrän. Oletusarvo=2.

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

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

Voit käyttää tulosmatriisia syötteenä muissa lyhennyslaskutoimituksissa **ΣInt()** ja **ΣPrn()**, sivu 197, sekä **bal()**, sivu 16.

and

BoolenLaus1 **and** *BoolenLaus2*⇒*Boolen lausekeBoolenLista1* **and** *BoolenLista2*⇒*Boolen listaBoolenMatriisi1* **and** *BoolenMatriisi2*⇒*Boolen matriisi*

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

Kokonaisluku1 **and***Kokonaisluku2*⇒*kokonaisluku*

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

Heksadesimaalisessa kantalukutilassa:

0h7AC36 and 0h3D5F	0h2C16

Tärkeää: Nolla, ei O-kirjain.

Binaarisessa kantalukutilassa:

0b100101 and 0b100	0b100

Desimaalisessa kantalukutilassa:

37 and 0b100	4

and

Katalogi >

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

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

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

angle()

Katalogi >

angle(*Arvo1***)⇒***arvo*

Laskee argumentin kulman tulkiten argumentin kompleksiluvuksi.

Astekulmatilassa:

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

Graadikulmatilassa:

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

Radiaanikulmatilassa:

$\text{angle}(1+\boldsymbol{i})$ 0.785398

$\text{angle}(\{1+2\cdot\boldsymbol{i},3+0\cdot\boldsymbol{i},0-4\cdot\boldsymbol{i}\})$
$\{1.10715,0.,-1.5708\}$

angle(*Lista1***)⇒***lista*
angle(*Matriisi1***)⇒***matriisi*

Laskee listan tai matriisin *Lista1*:n tai *Matriisi1*:n elementtien kulmista tulkiten jokaisen elementin kompleksiluvuksi, joka edustaa kaksiulotteista suorakulmakoordinaattipistettä.

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

ANOVA

Katalogi >

ANOVA *Lista1*,*Lista2*[,*Lista3*,...,*Lista20*] [,*Lippu*]

ANOVA

Katalogi >

Suorittaa yksisuuntaisen varianssianalyysin 2-20 perusjoukon keskiarvon vertailua varten. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

Lippu=0 datalle, *Lippu*=1 tilastoille

Tulosmuuttuja	Kuvaus
stat.F	F-tilaston arvo
stat.PVal	Alin merkitsevyystaso, 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

Katalogi >

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

Laskee kaksisuuntaisen varianssianalyysin 2-10 perusjoukon keskiarvojen vertaamiseksi. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

TasoRivi=0 lohkolle

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

```
Tulokset: Lohkomuoto
```

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

SARAKETEKIJÄN tulokset

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

RIVITEKIJÄN tulokset

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

VUOROVAIKUTUKSEN tulokset

Tulosmuuttuja	Kuvaus
stat.F̄Interact	F̄-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*

Näyttää viimeksi sievennetyn lausekkeen tuloksen.

56	56
56+4	60
60+4	64

approx()

Katalogi >

approx(*Arvo1***)**⇒*luku*

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

Tämä vastaa argumentin syöttämistä ja painikkeen ctrl enter painamista.

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

approx(*Lista1***)**⇒*lista*
approx(*Matriisi1***)**⇒*matriisi*

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

approx() Katalogi >

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

▶approxFraction() Katalogi >

Arvo ▶*approxFraction(*[]*Tol)*⇒*arvo*

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

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

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

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

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

0.83333333333333▶approxFraction(5.E-14)

$\frac{5}{6}$

{π,1.5}▶approxFraction(5.E-14)

$\left\{\frac{5419351}{1725033},\frac{3}{2}\right\}$

approxRational() Katalogi >

approxRational(*Arvo*[, *Tol*]**)**⇒*arvo*

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

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

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

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

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

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

arccos() Katso $\cos^{-1}()$, sivu 28.

arccosh() Katso $\cosh^{-1}()$, sivu 29.

arccot() Katso $\cot^{-1}()$, sivu 30.

arccoth() **Katso $\coth^{-1}()$, sivu 31.**

arccsc() **Katso $\csc^{-1}()$, sivu 33.**

arccsch() **Katso $\mathrm{csch}^{-1}()$, sivu 34.**

arcsec() **Katso $\sec^{-1}()$, sivu 140.**

arcsech() **Katso $\mathrm{sech}^{-1}()$, sivu 140.**

arcsin() **Katso $\sin^{-1}()$, sivu 149.**

arcsinh() **Katso $\sinh^{-1}()$, sivu 150.**

arctan() **Katso $\tan^{-1}()$, sivu 160.**

arctanh() **Katso $\tanh^{-1}()$, sivu 162.**

augment() Katalogi >

augment(*Lista1, Lista2***)⇒***lista*

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

augment({1,⁻3,2},{5,4}) {1,⁻3,2,5,4}

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

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

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

$\begin{bmatrix}5\\6\end{bmatrix}\rightarrow m2$ $\begin{bmatrix}5\\6\end{bmatrix}$

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

avgRC() Katalogi >

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

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

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

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

Laskee erotusosamäärän eteenpäin (keskimääräisen muutosnopeuden).

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

Kun *Arvo* määritetään, se ohittaa mahdolliset aikaisemmat muuttujamääritykset tai mahdolliset muuttujan nykyiset “|” -sijoitukset.

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

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

$x:=2$ 2

avgRC(x^2-x+2,x) 3.001

avgRC($x^2-x+2,x,.1$) 3.1

avgRC($x^2-x+2,x,3$) 6

B

bal() Katalogi >

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

bal(*NPmt*,*amortTable***)**⇒*arvo*

Lyhennysfunktio, joka laskee määritetyn maksuerän jälkeen jäljellä olevan velkasaldon.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 170.

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

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 170.

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

pyörArvo määrittää pyöristyksessä käytettävien desimaalien määrän. Oletusarvo=2.

bal(*NPmt*,*amortTable***)** laskee maksueränumeron *NPmt* jälkeen jäljellä olevan velkasaldon lyhennystaulukon *amortTable* perusteella. *amortTable*-argumentin on oltava matriisi, joka on kohdassa **amortTbl()** kuvatun muotoinen, katso sivu 7.

Huomaa: Katso myös **ΣInt()** ja **ΣPrn()**, sivu 197.

```
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
```

▶Base2 (▸Kantaluku2) Katalogi >

Kokonaisluku1 ▶**Base2**⇒*kokonaisluku*

```
256▸Base2                              0b100000000
0h1F▸Base2                                 0b11111
```

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

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

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

Negatiiviset luvut näytetään kahden komplementteina. Esimerkki:

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

-2^{63} näkyy muodossa
0h8000000000000000 heksadesimaalisessa kantalukutilassa
0b100...000 (63 zeros) binaarisessa kantalukutilassa

▶Base2 (▶Kantaluku2) Katalogi >

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) Katalogi >

Kokonaisluku1 ▶**Base10**⇒*kokonaisluku*

0b10011▶Base10	19
0h1F▶Base10	31

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

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

0b *binaariluku*
0h *heksadesimaaliluku*

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

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

▸Base10 (▶Kantaluku10) Katalogi >

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

▶Base16 (▸Kantaluku16) Katalogi >

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 desimaalikokonaisluvun, joka on etumerkillisen, 64 bitin binaarimuodon lukualueen ulkopuolella, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle. Lisätietoja, katso ▸**Base2**, sivu 16.

binomCdf() Katalogi >

binomCdf(*n*,*p***)**⇒*lista*

binomCdf() Katalogi >

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≤X≤*yläraja*) ⇒*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 ≤ *yläraja*), aseta *alaraja*=0

binomPdf() Katalogi >

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() Katalogi >

ceiling(*Arvo1***)**⇒*arvo*

Laskee lähimmän kokonaisluvun, joka on ≥ argumentti.

Argumentti voi olla reaali- tai kompleksiluku.

Huomaa: Katso myös **floor()**.

```
ceiling(.456)                                   1.
```

ceiling(*Lista1***)**⇒*lista*
ceiling(*Matriisi1***)**⇒*matriisi*

Laskee listan tai matriisin jokaisen elementin ylärajasta.

```
ceiling({-3.1,1,2.5})                   {-3.,1,3.}
ceiling([0  -3.2·i; 1.3  4])            [0  -3.·i; 2.  4]
```

centralDiff() Katalogi >

centralDiff(*Laus1***,***Muutt* [=*Arvo*][,*Askel*]**)** ⇒*lauseke*

centralDiff(*Laus1***,***Muutt* [,*Askel*]**)** **|***Muutt*=*Arvo*⇒*lauseke*

centralDiff(*Laus1***,***Muutt* [=*Arvo*][,*Lista*]**)** ⇒*lista*

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

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

Laskee numeerisen derivaatan käyttäen keskeiserotusosamäärän kaavaa.

Kun *Arvo* määritetään, se ohittaa mahdolliset aikaisemmat muuttujamääritykset tai mahdolliset muuttujan nykyiset “|” -sijoitukset.

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

Lista1:tä tai *Matriisi1*:tä käytettäessä operaatio mapataan listan arvojen tai matriisin elementtien suhteen.

Huomaa: Katso myös **avgRC()**.

$\text{centralDiff}(\cos(x),x)\|x=\frac{\pi}{2}$	-1.

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.

char(38)	"&"
char(65)	"A"

$\chi^2$2way Katalogi >

$\chi^2$2way *ObsMatriisi*

chi22way *ObsMatriisi*

$\chi^2$2way

Katalogi >

Laskee χ^2-testin tarkasteltavan matriisin *ObsMatriisi* sisältämän kaksisuuntaisen lukemataulukon arvojen välisestä assosiaatiosta. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

Lisätietoja matriisissa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

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^2Cdf()

Katalogi >

χ^2Cdf(*alaraja***,***yläraja***,***df***)**⇒*luku*, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

chi2Cdf(*alaraja***,***yläraja***,***df***)**⇒*luku*, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

Laskee χ^2-jakauman todennäköisyyden *alaraja*n ja *yläraja*n väliltä määritetylle vapausasteelle *df*.

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

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

χ^2GOF

Katalogi >

χ^2GOF *obsLista*,*expLista*,*df*

chi2GOF *obsLista*,*expLista*,*df*

χ^2GOF

Katalogi >

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 154.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

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

χ^2Pdf()

Katalogi >

χ^2Pdf(*XArvo***,***df***)**⇒*luku*, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

chi2Pdf(*XArvo***,***df***)**⇒*luku*, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

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

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

clearAZ

Katalogi >

clearAZ

Poistaa kaikki yksikirjaimiset muuttujat nykyiseltä tehtäväalueelta.

$5 \rightarrow b$	5
b	5
ClearAZ	*Done*
b	"Error: Variable is not defined"

clearAZ

Katalogi >

Jos yksi tai useampia muuttujia on lukittu, tämä komento aiheuttaa virheilmoituksen ja poistaa vain lukitsemattomat muuttujat. Katso **unLock, sivu 172.**

ClrErr

Katalogi >

ClrErr

Poistaa virhetilan ja nollaa järjestelmän muuttujan *errCode* .

Else-lauseessa lohkossa **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 115, ja **Try**, sivu 166.

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

Esimerkki **ClrErr**-komennosta, katso esimerkki 2 **Try**-komennon kohdalla, sivu 166.

colAugment()

Katalogi >

colAugment(*Matriisi1***,** *Matriisi2***)** ⇒*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.

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

$\begin{bmatrix}5 & 6\end{bmatrix}\rightarrow m2$ $\quad$ $\begin{bmatrix}5 & 6\end{bmatrix}$

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

colDim() Katalogi >

colDim(*Matriisi***)⇒***lauseke*

Laskee *Matriisi*n sisältämien sarakkeiden lukumäärän.

Huomaa: Katso myös **rowDim()**.

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

colNorm() Katalogi >

colNorm(*Matriisi***)⇒***lauseke*

Laskee maksimiarvon *Matriisi*n sarakkeissa olevien elementtien itseisarvojen summista.

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

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

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

conj() Katalogi >

conj(*Arvo1***)⇒***arvo*
conj(*Lista1***)⇒***lista*
conj(*Matriisi1***)⇒***matriisi*

Laskee argumentin liittokompleksiluvun.

Huomaa: Kaikkia määrittämättömiä muuttujia käsitellään reaalimuuttujina.

$$\text{conj}(1+2\cdot i) \qquad 1-2\cdot i$$

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

constructMat() Katalogi >

constructMat
(
Laus
,*Muutt1***,***Muutt2***,***numRivit***,***numSarakkeet***)**
⇒*matriisi*

Laskee matriisin argumentteihin perustuen.

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

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

constructMat() Katalogi >

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

CopyVar Katalogi >

CopyVar *Muutt1*, *Muutt2*

CopyVar *Muutt1*., *Muutt2*.

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

Define $a(x)=\frac{1}{x}$	*Done*
Define $b(x)=x^2$	*Done*
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 määrityksen funktioon *Muutt2*. Funktio *Muutt1* on määritettävä.

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

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

$aa.a:=45$	45
$aa.b:=6.78$	6.78
CopyVar $aa.,bb.$	*Done*
getVarInfo()	{aa.a "NUM" "" 0, aa.b "NUM" "" 0, bb.a "NUM" "" 0, bb.b "NUM" "" 0}

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

corrMat() Katalogi >

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

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

cos() trig painike

cos(*Arvo1***)⇒***arvo*
cos(*Lista1***)⇒***lista*

cos(*Arvo1***)** määrittää argumentin kosinin arvona.

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

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

Astekulmatilassa:

$\cos\left(\left(\frac{\pi}{4}\right)^{r}\right)$	0.707107
$\cos(45)$	0.707107
$\cos(\{0,60,90\})$	$\{1.,0.5,0.\}$

Graadikulmatilassa:

$\cos(\{0,50,100\})$	$\{1.,0.707107,0.\}$

Radiaanikulmatilassa:

$\cos\left(\frac{\pi}{4}\right)$	0.707107
$\cos(45°)$	0.707107

cos(*neliömatriisi1***)⇒***neliömatriisi*

Laskee *neliömatriisi1*:n matriisikosinin. Tämä ei ole sama kuin kunkin elementin kosinin laskeminen.

Kun skaalarista funktiota f(A) käytetään *neliömatriisi1*:een (A), tulos lasketaan algoritmilla:

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

neliömatriisi1:n on oltava diagonalisoitavissa. Lisäksi siinä ei voi olla symbolisia muuttujia, joille ei ole määritetty arvoa.

Matriiseista:

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

Radiaanikulmatilassa:

$$\cos\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$$

$$\begin{bmatrix} 0.212493 & 0.205064 & 0.121389 \\ 0.160871 & 0.259042 & 0.037126 \\ 0.248079 & -0.090153 & 0.218972 \end{bmatrix}$$

cos() trig painike

Tällöin A = X B X^{-1}ja f(A) = X f(B) X^{-1}.
Esimerkiksi, cos(A) = X cos(B) X^{-1}, jossa:

cos(B) =

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

Kaikki laskut suoritetaan liukulukuaritmetiikalla.

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

$\cos^{-1}$(*Arvo1***)⇒***arvo***

$\cos^{-1}$(*Lista1***)⇒***lista***

$\cos^{-1}$(*Arvo1***)** laskee kulman, jonka kosini on ***Arvo1*.**

$\cos^{-1}$(*Lista1***)** laskee listan ***Lista1***:n jokaisen elementin käänteiskosineista.

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

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

$\cos^{-1}$(*neliömatriisi1***)⇒***neliömatriisi***

Laskee ***neliömatriisi1***:n matriisin käänteiskosinin. Tämä ei ole sama kuin kunkin elementin käänteiskosinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Astekulmatilassa:

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

Graadikulmatilassa:

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

Radiaanikulmatilassa:

$\cos^{-1}(\{0,0.2,0.5\})$
$\{1.5708,1.36944,1.0472\}$

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

$\cos^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$

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

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

cosh() Katalogi >

cosh($Arvo1$**)⇒**$arvo$
cosh($Lista1$**)⇒**$lista$

cosh($Arvo1$**)** laskee argumentin hyperbolisen kosinin.

cosh($Lista1$**)** määrittää listan $Lista1$:n kunkin elementin hyperbolisista kosineista.

cosh($neliömatriisi1$**)⇒**$neliömatriisi$

Laskee $neliömatriisi1$:n matriisin hyperbolisen kosinin. Tämä ei ole sama kuin kunkin elementin hyperbolisen kosinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Astekulmatilassa:

$$\cosh\left(\left(\frac{\pi}{4}\right)^{r}\right) \qquad 1.74671\text{E}19$$

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 >

$\cosh^{-1}$($Arvo1$**)⇒**$arvo$

$\cosh^{-1}$($Lista1$**)⇒**$lista$

$\cosh^{-1}$($Arvo1$**)** laskee argumentin käänteisen hyperbolisen kosinin.

$\cosh^{-1}$($Lista1$**)** määrittää listan $Lista1$:n kunkin elementin käänteisistä hyperbolisista kosineista.

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

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

$$\cosh^{-1}(\{1,2.1,3\}) \qquad \{0,1.37286,\cosh^{-1}(3)\}$$

$\cosh^{-1}$($neliömatriisi1$**)⇒**$neliömatriisi$

Laskee $neliömatriisi1$:n matriisin käänteisen hyperbolisen kosinin. Tämä ei ole sama kuin kunkin elementin käänteisen hyperbolisen kosinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

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

cosh⁻¹() Katalogi >

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

cot() trig painike

cot(*Arvo1***)** ⇒ *arvo*
cot(*Lista1***)** ⇒ *lista*

Laskee *Arvo1*:n kotangentin tai määrittää listan *Lista1*:n kaikkien elementtien kotangenteista.

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

Astekulmatilassa:

cot(45)	1.

Graadikulmatilassa:

cot(50)	1.

Radiaanikulmatilassa:

cot({1,2.1,3})	{0.642093,⁻0.584848,⁻7.01525}

cot⁻¹() trig painike

cot⁻¹(*Arvo1***)**⇒*arvo*

cot⁻¹(*Lista1***)**⇒*lista*

Laskee kulman, jonka kotangentti on *Arvo1*, tai määrittää listan, joka sisältää *Lista1*: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)	.785398

coth() Katalogi >

coth(*Arvo1***)**⇒*arvo*
coth(*Lista1***)**⇒*lista*

coth(1.2)	1.19954
coth({1,3.2})	{1.31304,1.00333}

coth() Katalogi >

Laskee *Arvo1*:n hyperbolisen kotangentin tai määrittää listan *Lista1*:n kaikkien elementtien hyperbolisista kotangenteista.

coth⁻¹() Katalogi >

coth⁻¹(*Arvo1***)⇒***arvo*

coth⁻¹(*Lista1***)⇒***lista*

Laskee *Arvo1*:n käänteisen hyperbolisen kotangentin tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien käänteiset hyperboliset kotangentit.

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

$\coth^{-1}(3.5)$	0.293893
$\coth^{-1}(\{-2,2.1,6\})$	$\{-0.549306,0.518046,0.168236\}$

count() Katalogi >

count(*Arvo1taiLista1* [,*Arvo2taiLista2* [,...]])**⇒***arvo*

Laskee elementtien kokonaismäärän argumenteille, jotka sieventyvät numeroarvoiksi.

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

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

Listat & Taulukot -sovelluksessa voit käyttää solualueita argumenttien tilalla.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

$\text{count}(2,4,6)$	3
$\text{count}(\{2,4,6\})$	3
$\text{count}\left(2,\{4,6\},\begin{bmatrix}8 & 10\\ 12 & 14\end{bmatrix}\right)$	7

countif() Katalogi >

countif(*Lista,Kriteerit***)⇒***arvo*

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

Kriteeri voi olla:

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

Listat & Taulukot -sovelluksessa voit käyttää solualueita *Lista*n tilalla.

Listassa olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

Huomaa: Katso myös **sumIf()**, sivu 158, ja **frequency()**, sivu 59.

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

Laskee niiden elementtien lukumäärän, jotka ovat yhtä kuin 3.

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

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

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

Laskee lukumäärään 1:n ja 3:n.

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

Laskee lukumäärään 3:n, 5:n ja 7:n.

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

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

cPolyRoots() Katalogi >

cPolyRoots(*Poly,Muutt***)⇒***lista*

cPolyRoots(*Kertoinlista***)⇒***lista*

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

Poly on oltava polynomi laajennetussa muodossa yhdessä muuttujassa. Älä käytä laajentamattomia muotoja, kuten $y^2 \cdot y+1$ *tai* $x \cdot x+2 \cdot x+1$

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

Huomaa: Katso myös **polyRoots()**, sivu 118.

```
polyRoots(y^3+1,y)                              {-1}
cPolyRoots(y^3+1,y)
    {-1,0.5−0.866025·i,0.5+0.866025·i}
polyRoots(x^2+2·x+1,x)                          {-1,-1}
cPolyRoots({1,2,1})                             {-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({0.1,2.2,-5},{1,-0.5,0})
                    {-2.5,-5.,-2.25}
```

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.

```
crossP([1 2 3],[4 5 6])        [-3 6 -3]
crossP([1 2],[3 4])            [0 0 -2]
```

csc() trig painike

csc(*Arvo1***)⇒*arvo***
csc(*Lista1***)⇒*lista***

Laskee *Arvo1*:n kosekantin tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien kosekantit.

Astekulmatilassa:

```
csc(45)                        1.41421
```

Graadikulmatilassa:

```
csc(50)                        1.41421
```

Radiaanikulmatilassa:

```
csc({1,π/2,π/3})               {1.1884,1.,1.1547}
```

csc⁻¹() trig painike

csc⁻¹(*Arvo1***) ⇒ *arvo***

csc⁻¹(*Lista1***) ⇒ *lista***

Laskee kulman, jonka kosekantti on *Arvo1*, tai määrittää listan, joka sisältää *Lista1*:n kunkin elementin käänteiskosekantit.

Astekulmatilassa:

```
csc⁻¹(1)                       90.
```

Graadikulmatilassa:

```
csc⁻¹(1)                       100.
```

csc⁻¹() — trig painike

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

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

Radiaanikulmatilassa:

$\csc^{-1}(\{1,4,6\})$ $\{1.5708,0.25268,0.167448\}$

csch() — Katalogi >

csch(*Arvo1***)** ⇒ *arvo*

csch(*Lista1***)** ⇒ *lista*

Laskee *Arvo1*:n hyperbolisen kosekantin tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien hyperboliset kosekantit.

$\operatorname{csch}(3)$ 0.099822

$\operatorname{csch}(\{1,2.1,4\})$ $\{0.850918,0.248641,0.036644\}$

csch⁻¹() — Katalogi >

csch⁻¹(*Arvo***)** ⇒ *arvo*

csch⁻¹(*Lista1***)** ⇒ *lista*

Laskee *Arvo1*:n käänteisen hyperbolisen kosekantin tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien käänteiset hyperboliset kosekantit.

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

$\operatorname{csch}^{-1}(1)$ 0.881374

$\operatorname{csch}^{-1}(\{1,2.1,3\})$ $\{0.881374,0.459815,0.32745\}$

CubicReg — Katalogi >

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

Laskee 3. asteen polynomiregressiony = a ·x^3+b · x^2+c ·x+dlistoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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

CubicReg

Katalogi >

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 numeeristen tai merkkijonojen 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 223.

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 $Frekv$, $Luokkalista$ ja $Sisällytä\ luokat$ rajoitusten mukaisesti.
stat.YReg	Muokatun $Y\ Lista$:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen $Frekv$, $Luokkalista$ ja $Sisällytä\ luokat$ rajoitusten mukaisesti.
stat.FreqReg	Komentoja $stat.XReg$ ja $stat.YReg$ vastaava frekvenssilista.

cumulativeSum()

Katalogi >

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

Laskee listan $Lista1$:n sisältämien elementtien kumulatiivisista summista alkaen elementistä 1.

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

cumulativeSum() Katalogi >

cumulativeSum(*Matriisi1***)⇒***matriisi*

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

Tyhjä elementti listassa *Lista1* tai matriisissa *Matriisi1* tuottaa tyhjän elementin tuloksena olevaan listaan tai matriisiin. Lisätietoja tyhjistä elementeistä, katso sivu 223.

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

$$\text{cumulativeSum}(m1) \qquad \begin{bmatrix}1 & 2\\4 & 6\\9 & 12\end{bmatrix}$$

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

▸Cylind Katalogi >

Vektori **▸Cylind**

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.

$$[2\ \ 2\ \ 3]\blacktriangleright\text{Cylind} \qquad [2.82843\ \ \angle 0.785398\ \ 3.]$$

D

dbd() Katalogi >

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

▸DD Katalogi >

Laus1 **▸DD⇒***arvoLista1*
▸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 kulmatila-asetus tulkitsee graadeina, radiaaneina tai asteina.

Astekulmatilassa:

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

Graadikulmatilassa:

1▸DD	$\frac{9}{10}$°

Radiaanikulmatilassa:

(1.5)▸DD	85.9437°

▸Decimal

Katalogi >

Luku1▸*Decimal*⇒*arvo*

Lista1 ▸**Decimal**⇒*arvo*

Matriisi1 ▸**Decimal**⇒*arvo*

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

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

$\frac{1}{3}$▸Decimal	0.333333

Define (Määritä)

Katalogi >

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

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

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* → *Funktio*(*Param1,Param2*).

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

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

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

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

$g(3,-7)$	3

Define (Määritä) Katalogi >

EndPrgm

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

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

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

Huomaa: Katso myös **Define LibPriv**, sivu 39, ja **Define LibPub**, sivu 39.

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

Define LibPriv (Määritä LibPriv) Katalogi >

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 38, ja **Define LibPub**, sivu 39.

Define LibPub (Määritä LibPub) Katalogi >

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

Define LibPub (Määritä LibPub) Katalogi >

= *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 38, ja **Define LibPriv**, sivu 39.

deltaList() Katso ΔList(), sivu 87.

DelVar Katalogi >

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

DelVar *Muutt*.

Poistaa määritetyn muuttujan tai muuttujaryhmän muistista.

Jos yksi tai useampia muuttujia on lukittu, tämä komento aiheuttaa virheilmoituksen ja poistaa vain lukitsemattomat muuttujat. Katso **unLock, sivu 172.**

$2 \rightarrow a$	2
$(a+2)^2$	16
DelVar a	*Done*
$(a+2)^2$	"Error: Variable is not defined"

DelVar

Katalogi >

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

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

delVoid()

Katalogi >

delVoid(*Lista1***)⇒***lista*

Antaa tuloksena listan, jossa on listan *Lista1* sisältö, ja kaikki tyhjät elementit on poistettu.

Lisätietoja tyhjistä elementeistä, katso sivu 223.

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

det()

Katalogi >

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

Laskee *neliömatriisi*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 *Tolerassi*a 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:

$\det\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	-2
$\begin{bmatrix} 1.\text{E}20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow mat1$	$\begin{bmatrix} 1.\text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$
det(*mat1*)	0
det(*mat1*,.1)	1.E20

det() Katalogi >

5E-14 ·**max(dim(***neliömatriisi***))** ·
rowNorm(*neliömatriisi***)**

diag() Katalogi >

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

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

$\text{diag}\left(\begin{bmatrix}2 & 4 & 6\end{bmatrix}\right)$ $\begin{bmatrix}2 & 0 & 0\\0 & 4 & 0\\0 & 0 & 6\end{bmatrix}$

diag(*neliömatriisi***)⇒***rivimatriisi*

Laskee rivimatriisin, joka sisältää elementit *neliömatriisi*n päälävistäjästä.

neliömatriisi:n on oltava neliö.

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

$\text{diag}(Ans)$ $\begin{bmatrix}4 & 2 & 9\end{bmatrix}$

dim() Katalogi >

dim(*Lista***)⇒***kokonaisluku*

Laskee *Lista*n mitat.

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

dim(*Matriisi***)⇒***lista*

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

$\text{dim}\left(\begin{bmatrix}1 & -1\\2 & -2\\3 & 5\end{bmatrix}\right)$ $\{3,2\}$

dim(*Merkkijono***)⇒***kokonaisluku*

Laskee merkkijonon *Merkkijono* sisältämien merkkien lukumäärän.

$\text{dim}(\text{"Hello"})$ 5

$\text{dim}(\text{"Hello "\&"there"})$ 11

Disp

Katalogi >

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 määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

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

DispAt

Katalogi >

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

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

Rivinumero voidaan määritellä lausekkeeksi.

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

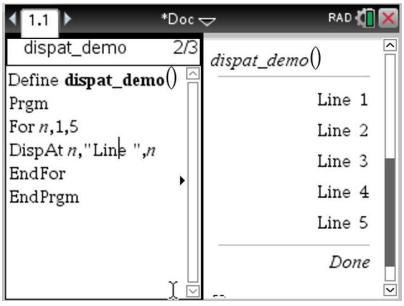
Tämä komento mahdollistaa ohjauspaneelin kaltaisen tuotoksen ohjelmista, joissa lausekkeen arvo tai anturin lukema päivitetään samalle riville.

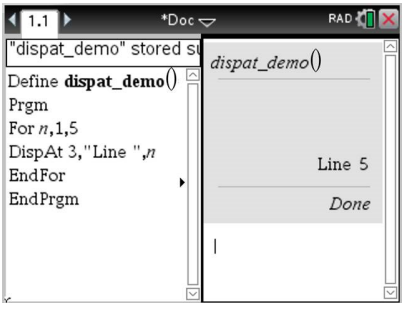
Komentoja **DispAtja Disp** voidaan käyttää samassa ohjelmassa.

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

```
DispAt
```

Esimerkki





Havainnollistavia esimerkkejä:

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

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.

Virheviestit	Kuvaus
Virheellinen tietotyyppi	Ensimmäisen argumentin on oltava numero.
Mitätön: DispAt mitätön	"Hello World" Datatyyppivirhe on mitätöity (jos soittopyyntö on määritelty)

▸DMS

Katalogi >

Lista **▸DMS**

Matriisi **▸DMS**

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 201 .

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

Astekulmatilassa:

(45.371)▸DMS	45°22'15.6"
({45.371,60})▸DMS	{45°22'15.6",60°}

dotP()

Katalogi >

dotP(*Lista1***,** *Lista2***)⇒***lauseke*

Laskee kahden listan “pistetulon”.

dotP(*Vektori1***,** *Vektori2***)⇒***lauseke*

Laskee kahden vektorin “pistetulon”.

Kummankin on oltava rivivektoreita, tai kummankin on oltava sarakevektoreita.

dotP({1,2},{5,6})	17
dotP([1 2 3],[4 5 6])	32

E

e^() — [e^x] painike

e^(*Arvo1***)⇒***arvo*

Laskee **e**:n arvon korotettuna *Arvo1*:n potenssiin.

e^{1}	2.71828
e^{3^2}	8103.08

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

Huomaa: Painikkeen [e^x] painaminen, jotta näkyviin saadaan e^(, on eri asia kuin näppäimistön merkin [E] painaminen.

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

*e***^(***Lista1***)⇒***lista*

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

Laskee **e**:n arvon korotettuna *Lista1*:n jokaisen elementin potenssiin.

*e***^(***neliömatriisi1***)⇒***neliömatriisi*

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

Laskee *neliömatriisi1*:n matriisieksponentin. Tämä ei ole sama kuin laskettaessa e korotettuna kunkin elementin mukaiseen potenssiin. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

eff() — Katalogi >

eff(*nimelliskorko,CpY***)⇒***arvo*

eff(5.75,12)	5.90398

Talouslaskentatoiminto, joka muuntaa nimelliskorkokannan *nimelliskorko* efektiiviseksi vuosikoroksi, kun *CpY* määritetään korkojaksojen lukumääräksi vuodessa.

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

Huomaa: Katso myös **nom()**, sivu 107.

eigVc() Katalogi >

eigVc(*neliömatriisi***)⇒***matriisi*

Laskee matriisin, joka sisältää ominaisvektorit reaaliselle tai kompleksiselle *neliömatriisi*lle, 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, … , x_n], tällöin:

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

*neliömatriisi*a 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.

Suorakulmakompleksimuodossa:

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

$$\text{eigVc}(m1)$$

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

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

eigVl() Katalogi >

eigVl(*neliömatriisi***)⇒***lista*

Laskee listan reaalisen tai kompleksisen *neliömatriisi*n ominaisarvoista.

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

Suorakulmakompleksimuodossa:

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

$$\text{eigVl}(m1)$$

$$\{-4.40941, 2.20471+0.763006\cdot i, 2.20471-0.$$

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

Else Katso If, sivu 72.

ElseIf

Katalogi >

If *BoolenLaus1* **Then**
    Lohko1
ElseIf *BoolenLaus2* **Then**
    Lohko2
    ⋮
ElseIf *BoolenLausN* **Then**
    LohkoN
EndIf
⋮

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

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

EndFor

Katso For, sivu 56.

EndFunc

Katso Func, sivu 60.

EndIf

Katso If, sivu 72.

EndLoop

Katso Loop, sivu 93.

EndPrgm

Katso Prgm, sivu 119.

EndTry

Katso Try, sivu 166.

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

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

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*).

Var on riippumaton muuttuja.

ListOfDepVars on riippuvien muuttujien luettelo.

Differentiaaliyhtälö:
y'=0.001*y*(100-y) ja y(0)=10

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

$$\begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9 & 11.8712 & 12.9174 & 14.042 \end{bmatrix}$$

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

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 $y2(0)=5$

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

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

euler () Katalogi >

{*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 **sign(***VarStep***)** = **sign(***VarMax*-*Var0***)** ja ratkaisut lasketaan *Var0*+i·*VarStep* kaikille i=0,1,2,... niin, että *Var0*+i·*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***)** ⇒ *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 reaaliluvuksi.

Aseta RGB-ledin sininen väri puolelle intensiteetille.

```
lum:=127                                   127
Send "SET COLOR.BLUE eval(lum)"            Done
```

Palauta sininen väri OFF-tilaan.

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

eval()-argumentin on sievennyttävä reaaliluvuksi.

```
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
```

eval() Laitevalikko

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 62), **GetStr** (sivu 69), ja **Send** (sivu 141).

Suorita ohjelma.

fadein()	*Done*

n:=0.25	0.25
m:=8	8
$n \cdot m$	2.
Send "SET COLOR.BLUE ON TIME eval(n·m)"	*Done*
iostr.SendAns	"SET COLOR.BLUE ON TIME 2"

Exit Katalogi >

Exit

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

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

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

Funktion listaus:

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

exp() e^x painike

exp(*Arvo1***)⇒***arvo*

Laskee **e**:n arvon korotettuna *Arvo1*:n potenssiin.

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

e^1	2.71828
e^{3^2}	8103.08

exp() — e^x painike

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

exp(*Lista1***)⇒***lista*

Laskee **e**:n arvon korotettuna *Lista1*:n jokaisen elementin potenssiin.

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

exp(*neliömatriisi1***)⇒***neliömatriisi*

Laskee *neliömatriisi1*:n matriisieksponentin. Tämä ei ole sama kuin laskettaessa **e** korotettuna kunkin elementin mukaiseen potenssiin. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

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

expr() — Katalogi >

expr(*Merkkijono***)⇒***lauseke*

Määrittää *Merkkijono*n sisältämän merkkijonon lausekkeena ja suorittaa toimenpiteen välittömästi.

"Define cube(x)=x^3" → *funcstr*	
	"Define cube(x)=x^3"
expr(*funcstr*)	*Done*
cube(2)	8

ExpReg — Katalogi >

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

Laskee eksponentiaalisen regressiony = a · $(b)^x$listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen 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 223.

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

F

factor() Katalogi >

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 [on]-painiketta painettuna ja paina toistuvasti [enter]-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() Katalogi >

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

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

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

Aseta P($X \le$ *yläraja*):lle *alaraja* = 0.

Fill

Katalogi >

Fill *Arvo, matriisiMuutt⇒matriisi*

Korvaa muuttujan *matriisiMuutt* jokaisen elementin *Arvo*lla.

*matriisiMuutt*ujan on oltava valmiiksi olemassa.

Fill *Arvo, listaMuutt⇒lista*

Korvaa muuttujan *listaMuutt* jokaisen elementin *Arvo*lla.

*listaMuutt*ujan on oltava valmiiksi olemassa.

$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} \rightarrow amatrix$ $\qquad$ $\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$

Fill 1.01,*amatrix* $\qquad$ *Done*

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

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

Fill 1.01,*alist* $\qquad$ *Done*

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

FiveNumSummary

Katalogi >

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

Antaa lyhennetyn version 1 muuttujan tilastoista listalle *X*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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.

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. Lisätietoja tyhjistä elementeistä, katso sivu 223.

Tulosmuuttuja	Kuvaus
stat.MinX	x:n arvojen minimi
stat.Q_1X	x:n ensimmäinen neljännes
stat.MedianX	x:n mediaani
stat.Q_3X	x:n kolmas neljännes
stat.MaxX	x:n arvojen maksimi

floor() Katalogi >

floor(*Arvo1***)⇒***kokonaisluku*

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

Argumentti voi olla reaali- tai kompleksiluku.

```
floor(-2.14)                                  -3.
```

floor(*Lista1***)⇒***lista*
floor(*Matriisi1***)⇒***matriisi*

Määrittää listan tai matriisin jokaisen elementin alarajasta.

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

```
floor({3/2,0,-5.3})                           {1,0,-6.}
floor([1.2  3.4; 2.5  4.8])                   [1.  3.; 2.  4.]
```

For Katalogi >

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

Suorittaa *Lohko*n sisältämät lausekkeet iteratiivisesti jokaiselle muuttujan *Muutt* arvolle, *Matala*sta *Korkea*aan 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 (:).

```
Define g()=Func                               Done
    Local tempsum,step,i
    0→tempsum
    1→step
    For i,1,100,step
    tempsum+i→tempsum
    EndFor
    EndFunc
g()                                           5050
```

For

Katalogi >

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

format()

Katalogi >

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

Määrittää *Arvo*n merkkijonona muotoilumallin perusteella.

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

F[n]: Kiinteä muoto. n on desimaalipisteen jälkeen näytettävien numeroiden lukumäärä.

S[n]: Kymmenpotenssimuoto. n on desimaalipisteen jälkeen näytettävien numeroiden lukumäärä.

E[n]: Tekninen esitystapa. n on ensimmäisen merkitsevän numeron jälkeen näytettävien numeroiden lukumäärä. Eksponentti säätyy kolmella kerrolliseksi, 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.

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

fPart() Katalogi >

fPart(*Laus1***)⇒***lauseke*
fPart(*Lista1***)⇒***lista*
fPart(*Matriisi1***)⇒***matriisi*

Laskee argumentin murtolukuosan.

Kun kyseessä on lista tai matriisi, laskee elementtien murtolukuosat.

Argumentti voi olla reaali- tai kompleksiluku.

```
fPart(-1.234)                    -0.234
fPart({1,-2.3,7.003})      {0,-0.3,0.003}
```

FPdf() Katalogi >

FPdf(*XArvo***,***dfOsoitt***,***dfNimitt***)⇒***luku*, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

Laskee F-jakauman todennäköisyyden *XArvo*n kohdalle määritetyille *dfOsoitt*ajalle (vapausasteet) ja *dfNimitt*äjälle.

freqTable▸list() Katalogi >

freqTable▸list (*Lista1***,***frekvKokonaislukuLista***)⇒***lista*

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

Lista1 voi olla mikä tahansa kelvollinen lista.

*frekvKokonaislukuLista*n 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.

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

freqTable▸list() Katalogi >

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

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

frequency() Katalogi >

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

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

Jos *lokerotLista* on {b(1), b(2), …, b(n)}, määritetyt alueet ovat {**?**≤b(1), b(1)<**?**≤b(2),…,b(n-1)<**?**≤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 **countIf()** termeillä ilmaistuna vastaus on { countIf(list, **?**≤b(1)), countIf (list, b(1)<**?**≤b(2)), …, countIf(list, b(n-1)<**?** ≤b(n)), countIf(list, b(n)>**?**)}.

Niitä *Lista1*:n elementtejä, joita ei voi "lokeroida", ei huomioida. Tyhjiä elementtejä ei myöskään huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

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

Huomaa: Katso myös **countIf()**, sivu 32.

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}

Vastauksen selitys:

2 *Datalista*n elementtiä on ≤2.5

4 *Datalista*n elementtiä on >2.5 ja ≤4.5

3 *Datalista*n elementtiä on >4.5

Elementti "hei" on merkkijono, jota ei voi sijoittaa mihinkään määritetyistä lokeroista.

FTest_2Samp Katalogi >

FTest_2Samp *Lista1*,*Lista2*[,*Frekv1* [,*Frekv2*[,*Hypot*]]]

FTest_2Samp *Lista1*,*Lista2*[,*Frekv1*

FTest_2Samp Katalogi >

[,*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 154.)

Kun H_a: σ1 > σ2, aseta *Hypot*>0
Kun H_a: σ1 ≠ σ2 (oletus), aseta *Hypot* =0
Kun H_a: σ1 < σ2, aseta *Hypot*<0

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

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 *Lista 1*:n ja *Lista 2*:n sisältämille datasekvensseille
stat.x1_bar stat.x2_bar	Otoksen keskiarvot *Lista 1*:n ja *Lista 2*:n sisältämille datasekvensseille
stat.n1, stat.n2	Otosten koko

Func Katalogi >

Func
Lohko
EndFunc

Malli käyttäjän määrittämän funktion luomista varten.

Määritä paloittain määritelty funktio:

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

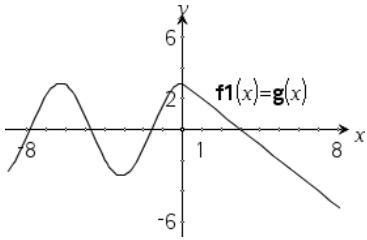
Func

Katalogi >

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

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

Funktion g(x) kuvaajan piirtämisen tulos



G

gcd()

Katalogi >

gcd(*Arvo1, Arvo2***)⇒***lauseke*

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.

$\gcd(18,33)$ 3

gcd(*Lista1, Lista2***)⇒***lista*

Laskee *Lista1*:n ja *Lista2*:n toisiaan vastaavien elementtien suurimmat yhteiset jakajat.

$\gcd(\{12,14,16\},\{9,7,5\})$ $\{3,7,1\}$

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

Laskee *Matriisi1* :n ja *Matriisi2*:n toisiaan vastaavien elementtien suurimmat yhteiset jakajat.

$\gcd\left(\begin{bmatrix}2 & 4\\6 & 8\end{bmatrix},\begin{bmatrix}4 & 8\\12 & 16\end{bmatrix}\right)$ $\begin{bmatrix}2 & 4\\6 & 8\end{bmatrix}$

geomCdf()

Katalogi >

geomCdf(*p,alaraja,yläraja***)⇒***luku*, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

geomCdf(*p,yläraja***)**kun P(1≤X≤*yläraja*) ⇒*luku*, jos *yläraja* on luku, *lista*, jos *yläraja* on lista

geomCdf() Katalogi >

Laskee kumulatiivisen geometrisen todennäköisyyden *alaraja*lta *yläraja*lle määritetyllä onnistumistodennäköisyydellä *p*.

Kun P(X ≤ *yläraja*), aseta *alaraja* = 1.

geomPdf() Katalogi >

geomPdf(*p***,***XArvo***)⇒***luku*, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

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

Get Laitevalikko

Get [*kehotemerkkijono*,] *var*[, *statusVar*]

Get [*kehotemerkkijono*,] *func*(*arg1*, ...*argn*)[, *statusVar*]

Ohjelmointikomento: Noutaa arvon liitetystä TI-Innovator™ Hub ja sijoittaa arvon muuttujaan *var*.

Arvo täytyy kysyä:

- Etukäteen **Send "READ ..."** - komennolla.

  — tai —

- Upottamalla **"READ ..."** kysy valinnaisena *kehotemerkkijono*argumenttina. Tämä menetelmä antaa sinun käyttää yksittäistä komentoa kysyäksesi arvoa ja hakeaksesi sen.

Tapahtuu implisiittinen yksinkertaistus. Esimerkiksi vastaanotettu merkkijono "123" tulkitaan numeeriseksi arvoksi. Säilyttääksesi merkkijonon käytä toimintoa **GetStr** toiminnon **Get** sijaan.

Esimerkki: Kysy laitteen sisäänrakennetun valaistusanturin tämänhetkinen arvo. Käytä komentoa **Get** noutaaksesi arvon ja sijoittaaksesi se muuttujaan *lightval*.

Send "READ BRIGHTNESS"	*Done*
Get *lightval*	*Done*
lightval	0.347922

Upota READ-kysely komentoon **Get**.

Get "READ BRIGHTNESS",*lightval*	*Done*
lightval	0.378441

Get — Laitevalikko

Jos sisällytät valinnaisen argumentin *statusVar*, sille määrätään arvo toimenpiteen onnistumisen perusteella. Arvo nolla merkitsee, ettei tietoa ole vastaanotettu.

Järjestyksessä toisessa syntaksissa *func*()-argumentti sallii ohjelman tallentaa vastaanotetun merkkijonon funktiomääritelmänä. Tämä syntaksi toimii ikään kuin ohjelma suorittaisi komennon:

Määrittele *func*(*arg1*, ...*argn*) = *vastaanotettu merkkijono*

Sen jälkeen ohjelma voi käyttää määriteltyä funktiota *func*().

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

Huomio: Katso myös **GetStr**, sivu 69 ja **Send** sivu 141.

getDenom() — Katalogi >

getDenom(*Murtoluku1***)⇒***arvo*

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

$x:=5:\ y:=6$	6
$\text{getDenom}\left(\frac{x+2}{y-3}\right)$	3
$\text{getDenom}\left(\frac{2}{7}\right)$	7
$\text{getDenom}\left(\frac{1}{x}+\frac{y^2+y}{y^2}\right)$	30

getKey() — Katalogi >

getKey([0|1]) ⇒ returnString

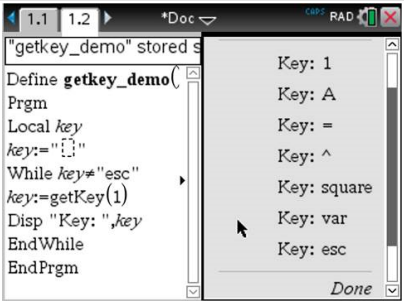
Kuvaus:getKey() – mahdollistaa TI-Basic -ohjelmalle näppäimistön – kannettava laite, pöytätietokone ja emulaattori pöytätietokoneella.

Esimerkki:

getKey()

Esimerkki:

- alas painettu näppäin := **getKey()** palauttaa näppäimen tai tyhjän merkkijonon, jos mitään näppäintä ei ole painettu. Tämä komento palaa välittömästi.
- alas painettu näppäin := **getKey(1)** odottaa, kunnes jotakin näppäintä painetaan. Tämä komento pysäyttää ohjelman suorituksen, kunnes jotakin näppäintä on painettu.



Näppäimen pitäminen painettuna:

Kannettava laite/Emulaattorinäppäin	Pöytätietokone	Palauta arvo
Esc	Esc	"esc"
Kosketuslevy – Yläpainallus	N/A	"ylös"
Päällä	N/A	"päävalikko"
Scratchapps	N/A	muistilehtiö
Kosketuslevy – Vasen painallus	N/A	"vasen"
Kosketuslevy – 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"

Kannettava laite/Emulaattorinäppäin	Pöytätietokone	Palauta arvo
=	=	"="
Trigonometria	N/A	Trigonometria
0:sta 9:ään	0-9	"0" ... 9
Sapluunat	N/A	"sapluuna"
Lista	N/A	"lista"
^	^	"^"
X^2	N/A	"neliö"
/ (jakonäppäin)	/	"/"
* (kertonäppäin)	*	"*"
e^x	N/A	"eksponentti"
10^x	N/A	"10voima"
+	+	"+"
-	-	"-"
(	(	"("
)	)	")"
.	.	"."
(-)	N/A	"-" (negatiivinen merkki)
Syötä	Syötä	"syötä"
ee	N/A	"E" (tieteellinen merkintä E)
a – z	a–z	alfa = kirjainmerkki painettu (pieni kirjain) ("a" – "z")
vaihto a–z	vaihto a–z	alfa = kirjainmerkki painettu "A" - "Z"
		Huomaa: Ctrl-vaihto lukitsee isot kirjaiment
?!	N/A	"?!"
pii	N/A	"pii"
Lippu	N/A	ei palautusta

Kannettava laite/Emulaattorinäppäin	Pöytätietokone	Palauta arvo
,	,	","
Palautus	N/A	Palautus
Välilyönti	Välilyönti	Välilyönti
Ei pääsyä	Näppäimet erikoismerkeille, kuten @,!,^, etc.	Kirjainmerkki on palautettu
N/A	Toimintonäppäimet	Kirjainmerkkejä ei ole palautettu
N/A	Erityiset näytön kontrollinäppäimet	Kirjainmerkkejä ei ole palautettu
Ei pääsyä	Muita näytön näppäimiä, jotka eivät ole käytettävissä laskimessa, kun getkey() odottaa näppäimen painallusta. ({, },;, :, ...)	Sama kirjainmerkki, jonka saat Notesista (ei matemaattisessa kentässä)

Huomaa: On tärkeää huomata, että ohjelman **getKey ()** läsnäolo muuttaa ohjelmaa tiettyjen tapahtumien käsittelyä. Joitakin näistä kuvataan jäljempänä.

Lopeta ohjelma ja käsittele tapahtuma – aivan kuin jos käyttäjä poistuisi ohjelmasta painamalla **ON** -näppäintä

"**Tuki** " jäljempänä tarkoittaa – Järjestelmä toimii odotetusti – ohjelma jatkuu.

Tapahtuma	laite	Työpöytä – TI-Nspire ™ Ohjelmisto opiskelijoille
Pikakysely	Lopeta ohjelma, käsittele tapahtuma	Sama kuin kannettava laite (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software – vain)
Etätiedoston hallinta (Sisältää lähetyksen "Exit Press 2 Test" -tiedoston toisesta kämmenlaitteesta tai työpöydän kannettavasta laitteesta)	Lopeta ohjelma, käsittele tapahtuma	Sama kuin kannettava laite. (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-vain)
Lopeta oppitunti	Lopeta ohjelma, käsittele tapahtuma	Tuki (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-vain)

Tapahtuma	Laite	Työpöytä – TI-Nspire ™ Kaikki versiot
TI-Innovator™ Hub Yhdistä / katkaise	Tuki – onnistuu komennolla TI-Innovator™ Hub. Kun olet lopettanut ohjelman, TI-Innovator™ Hub työskentelee edelleen kämmenlaitteen kanssa.	Sama kuin kannettava laite

getLangInfo() Katalogi >

getLangInfo()⇒*merkkijono*

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”

getLangInfo()	"en"

getLockInfo() Katalogi >

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 90, ja unLock, sivu 172.**

a:=65	65
Lock *a*	*Done*
getLockInfo(*a*)	1
a:=75	"Error: Variable is locked."
DelVar *a*	"Error: Variable is locked."
Unlock *a*	*Done*
a:=75	75
DelVar *a*	*Done*

getMode(*TilanNimiKokonaisluku***)⇒***arvo*

getMode(0)⇒*lista*

getMode(*TilanNimiKokonaisluku***)** laskee arvon, joka vastaa *TilanNimiKokonaisluku*-tilan nykyistä asetusta.

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 143.

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

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
Vektorimuoto	6	**1**=Suorakulma, **2**=Sylinteri, **3**=Pallo
Kantaluku	7	**1**=Desimaali, **2**=Heksagonaalinen, **3**=Binaarinen

getNum() Katalogi >

getNum(*Murtoluku1***)⇒***arvo*

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

$x:=5: y:=6$	6
$\text{getNum}\left(\frac{x+2}{y-3}\right)$	7
$\text{getNum}\left(\frac{2}{7}\right)$	2
$\text{getNum}\left(\frac{1}{x}+\frac{1}{y}\right)$	11

GetStr Hub-valikko

GetStr [*kehotemerkkijono*,] *var*[, *statusVar*]

GetStr [*kehotemerkkijono*,] *func*(*arg1*, ...*argn*)[, *statusVar*]

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 62 ja **Send** sivu 141.

Katso esimerkit kohdasta **Get**.

getType() Katalogi >

getType(*var***)⇒***merkkijono*

Antaa tulokseksi merkkijonon, joka ilmoittaa muuttujan *var* datatyypin.

Jos muuttujaa *var* 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"
DelVar *temp*	*Done*
$\text{getType}(temp)$	"NONE"

getVarInfo()⇒*matriisi* tai *merkkijono*

getVarInfo(*LibNameString***)**⇒*matriisi* tai *merkkijono*

getVarInfo() laskee tietomatriisin (muuttujan nimi, tyyppi, kirjaston käytettävyys ja lukittu/lukitsematon-tila) kaikille nykyisessä tehtävässä määritetyille muuttujille ja kirjasto-objekteille.

Jos yhtään muuttujaa ei ole määritetty, **getVarInfo()** antaa vastauksena merkkijonon "NONE".

getVarInfo(*KirjNimiMerkkijono***)**antaa tuloksena tietomatriisin kaikista kirjastossa *KirjNimiMerkkijono* määritetyistä kirjasto-objekteista. *KirjNimiMerkkijono*n on oltava merkkijono (lainausmerkkien sisällä oleva teksti) tai merkkijonomuuttuja.

Jos kirjastoa *KirjNimiMerkkijono* ei ole olemassa, esiintyy virhe.

Huomaa vasemmanpuoleinen esimerkki, jossa funktion **getVarInfo()** vastaus on määritetty muuttujaan *vs*. Jos muuttujan *vs* riviä 2 tai riviä 3 yritetään näyttää, tuloksena on “Kelpaamaton lista tai matriisi” -virhe, koska vähintään yksi näiden rivien elementeistä (esimerkiksi muuttuja *b*) sieventyy uudelleen matriisiksi.

Tämä virhe voi esiintyä myös käytettäessä *Ans*-muuttujaa funktion **getVarInfo()** tuloksen uudelleenlaskennassa.

Järjestelmä antaa edellä mainitun virheen, koska ohjelmiston nykyinen versio ei tue yleistettyä matriisirakennetta, jossa matriisin elementti voi olla joko matriisi tai lista.

getVarInfo()	"NONE"
Define x=5	*Done*
Lock x	*Done*
Define LibPriv y={1,2,3}	*Done*
Define LibPub $z(x)$=3·x^2−x	*Done*
getVarInfo()	$\begin{bmatrix} x & \text{"NUM"} & \text{"}\square\text{"} & 1 \\ y & \text{"LIST"} & \text{"LibPriv "} & 0 \\ z & \text{"FUNC"} & \text{"LibPub "} & 0 \end{bmatrix}$
getVarInfo(*tmp3*)	"Error: Argument must be a string"
getVarInfo("tmp3")	[*volcyl2* "NONE" "LibPub " 0]

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

Goto

Katalogi >

Goto *tunnusnimi*

Siirtää ohjauksen tunnukseen *tunnusnimi*.

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

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

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

▶Grad

Katalogi >

Laus1 ▶ **Grad**⇒*lauseke*

Muuttaa *Laus1*:n graadikulmaan.

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

Astekulmatilassa:

```
(1.5)▶Grad                    (1.66667)g
```

Radiaanikulmatilassa:

```
(1.5)▶Grad                    (95.493)g
```

I

identity()

Luettelo >

identity(*kokonaisluku***)** ⇒ *matriisi*

Laskee identiteettimatriisin, jonka koko on *kokonaisluku*.

Kokonaisluvun on oltava positiivinen kokonaisluku.

```
identity(4)
[1 0 0 0
 0 1 0 0
 0 0 1 0
 0 0 0 1]
```

Jos*BoolenLaus*
Ilmaisut

Jos *BoolenLaus* **Niin**
Lohko
EndIf

Jos *BoolenLaus* on tosi, suorittaa yhden lausekkeen *Lauseke* tai lausekkeiden lohkon *Lohko* ennen suorituksen jatkamista.

Jos *BoolenLaus* 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.

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

Jos *BoolenLaus*, **niin**
Lohko1
Tai
Lohko2
EndIf

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

Jos *BoolenLaus* 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
              Else
              Return x
              EndIf
              EndFunc
g(12)                                 12
g(-12)                                12
```

Jos

Luettelo >

Jos *BoolenLaus1*, **niin**
    Lohko1
Jos taas *BoolenLaus2*, **niin**
    Lohko2
⋮
Jos taas *BoolenLausN*, **niin**
    LohkoN
EndIf

Sallii haarautumisen Jos *BoolenLaus1* on tosi, suorittaa *Lohko1*:n Jos *BoolenLaus1* on epätosi, laskee *BoolenLaus2*:n jne.

```
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
g(⁻4)                           4
g(10)                           3
```

ifFn()

Luettelo >

ifFn(*BoolenLaus*,*Arvo_Jos_tosi* [,*Arvo_Jos_epätosi* [,*Arvo_Jos_tuntematon*]]**) ⇒** *lauseke, lista tai matriisi*

Laskee *BoolenLaus* (jokaiselle *BoolenLaus*) elementille) ja antaa tuloksen noudattaen seuraavia sääntöjä:

- *BoolenLaus* voi testata yksittäisen arvon, listan tai matriisin.
- Jos jokin *BoolenLaus* elementti on tosi, laskee vastaavan elementin lausekkeesta *Arvo_Jos_tosi*.
- Jos jokin *BoolenLaus* elementti on epätosi, laskee vastaavan elementin lausekkeesta *Arvo_Jos_epätosi*. Jos jätät pois lausekkeen *Arvo_Jos_epätosi*, laskee määrittelemättömäksi.
- Jos *BoolenLaus* 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, Boolen testiä sovelletaan jokaiseen sijaintiin Boolen lausekkeessa*BoolenLaus*.

```
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.

```
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.

```
ifFn({1,2,3}<2.5,{5,6,7})           {5,6,undef}
```

Arvoa *Arvo_Jos_epätosi* ei ole määritelty. Käytetään merkintää EiMäär

ifFn() Luettelo >

Huomaa: Jos sievennetty *BoolenLaus* ilmaisu sisältää listan tai matriisin, kaikkien muiden lista- tai matriisiargumenttien on oltava samansuuruisia, ja myös tuloksen on oltava samansuuruinen.

```
ifFn({2,"a"}<2.5,{6,7},{9,10},"err")
                                {6,"err"}
```

Yksi elementti valittu lausekkeesta *Arvo_Jos_tosi*. Yksi elementti valittu lausekkeesta *Arvo_Jos_epätosi*.

imag() Luettelo >

imag(*Arvo1*) ⇒ *arvo*

Laskee argumentin imaginaarisen osan.

```
imag(1+2·i)                                  2
```

imag(*Lista1*) ⇒ *lista*

Laskee listan alkutekijöiden imaginaarisista osista.

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

imag(*Matriisi1*) ⇒ *matriisi*

Laskee matriisin alkutekijöiden imaginaarisista osista.

```
imag([1   2 ])                        [0  0]
     [i·3 i·4]                        [3  4]
```

Epäsuora operaattori Katso #(), sivu 199.

inString() Luettelo >

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

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.

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

int() Luettelo >

int(*Arvo***)** ⇒ *kokonaisluku*
int(*Lista1***)** ⇒ *lista*
int(*Matriisi1***)** ⇒ *matriisi*

Laskee suurimman kokonaisluvun, joka on pienempi tai yhtä suuri kuin argumentti. Tämä funktio on identtinen funktion pohjan **floor()** kanssa.

Argumentti voi olla reaali- tai kompleksiluku.

Kun kyseessä on lista tai matriisi, laskee kunkin elementin suurimman kokonaisluvun.

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

intDiv() Luettelo >

intDiv(*Luku1***,** *Luku2***)** ⇒ *kokonaisluku*
intDiv(*Lista1***,** *Lista2***)** ⇒ *lista*
intDiv(*Matriisi1***,** *Matriisi2***)** ⇒ *matriisi*

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

Laskee sarjoille ja matriiseille lausekkeen (argumentti 1 ÷ argumentti 2) etumerkillisen kokonaislukuosan kullekin elementtiparille.

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

interpoloi () Luettelo >

Interpoloi(*xArvo***,** *xList***,** *yList***,** *yPrimeList***)** ⇒ *lista*

Tällä toiminnolla suoritetaan seuraavaa:

Differentiaaliyhtälö:
y'=-3•y+6•t+5 and y(0)=5

```
rk:=rk23(-3·y+6·t+5,t,y,{0,10},5,1)
[0.     1.        2.        3.        4.
 5.   3.19499  5.00394  6.99957  9.00593  1(
```

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

interpoloi () Luettelo >

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 luku tai lukuluettelo.

Käytä interpolate()-funktiota laskeaksesi funktion arvot x-arvolistalle:

```
xvaluelist:=seq(i,i,0,10,0.5)
{0,0.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,▸
xlist:=mat▸list(rk[1])
{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.}
ylist:=mat▸list(rk[2])
{5.,3.19499,5.00394,6.99957,9.00593,10.9978▸
yprimelist:=⁻3·y+6·t+5|y=ylist and t=xlist
{⁻10.,1.41503,1.98819,2.00129,1.98221,2.006▸
interpolate(xvaluelist,xlist,ylist,yprimelist)
{5.,2.67062,3.19499,4.02782,5.00394,6.00011▸
```

invχ^2() Luettelo >

invχ^2(*Ala***,***df***)**

invChi2(*Ala***,***df***)**

Laskee käänteisen kumulatiivisen χ^2 (chi-neliö) -todennäköisyysfunktion, joka on määritelty 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() Luettelo >

invBinom
(*CumulativeProb*,*NumTrials*,*Prob*,
OutputForm)⇒ *asteikkomuoto* tai
matriisi

Käänteinen binomi. Johtuen kokeiden (*NumTrials*) ja kunkin kokeen todennäköisyydestä onnistua (*Prob*), tämä toiminto laskee onnistumisten minimimäärän, *k*siten, että arvo *k* on suurempi tai yhtä suuri kuin kumuloituva todennäköisyys (*CumulativeProb*).

OutputForm=**0**, näyttää tuloksen asteikkomuodossa (oletus).

OutputForm=**1**, näyttää tuloksen matriisina.

Esimerkki: Mary ja Kevin pelaavat noppapeliä. Maryn on arvattava, miten monta kertaa numero 6 enintään esiintyy 30 heittoa kohti. Jos numero 6 esiintyy yhtä monta kertaa tai vähemmän, Mary voittaa. Lisäksi, mitä pienempi on hänen arvaamansa määrä, sitä suuremmat ovat hänen voittonsa. Mikä on pienin määrä, jonka Mary voi arvata, jos hän haluaa voittamisen todennäköisyyden olevan suurempi kuin 77 %?

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

invBinomN() Luettelo >

invBinomN(*CumulativeProb*,*Prob*,
NumSuccess,*OutputForm*)
⇒ *asteikkomuoto* tai *matriisi*

N:n suhteen käänteisesti binominen. Johtuen menestyksen todennäköisyydestä kussakin kokeessa (*Prob*), onnistumisten määrä (*NumSuccess*), tämä funktio laskee kokeiden vähimmäismäärän *N* siten, että arvo, *N*, on vähemmän tai yhtä suuri kuin annettu kumulatiivinen todennäköisyys (*CumulativeProb*).

OutputForm=**0**, näyttää tuloksen asteikkomuodossa (oletus).

OutputForm=**1**, näyttää tuloksen matriisina.

Esimerkki: Monique harjoittelee koriin heittoa koripallossa. Hän tietää kokemuksesta, että hänen mahdollisuutensa tehdä kori on 70 %. Hän suunnittelee harjoittelevansa, kunnes hän on tehnyt 50 koria. Kuinka montaa koria hänen on yritettävä varmistaakseen, että todennäköisyys tehdä ainakin 50 koria on enemmän kuin 0,99?

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

invNorm() Luettelo >

invNorm(*Ala*[,μ[,σ]])

Laskee käänteisen kumulatiivisen normaalijakaumafunktion annetulle alalle *Ala, joka on normaalijakaumakäyrän alapuolella ja jonka määräävät* μ ja σ.

invt() Luettelo >

invt(*Ala***,***df***)**

Laskee käänteisen kumulatiivisen student t -todennäköisyysfunktion, jonka määräävät vapausaste, *df* ja annettu alue *Ala* käyrän alapuolella.

iPart() Luettelo >

iPart(*Luku***)** ⇒ *kokonaisluku*
iPart(*Lista1***)** ⇒ *lista*
iPart(*Matriisi1***)** ⇒ *matriisi*

Laskee argumentin kokonaisosan.

Laskee sarjoille ja matriiseille kunkin elementin kokonaisosan.

Argumentti voi olla reaali- tai kompleksiluku.

iPart(⁻1.234)	⁻1.
iPart({3/2,⁻2.3,7.003})	{1,⁻2.,7.}

irr() Luettelo >

irr(*CF0***,***CFLista* [,*CFFrekv*]**)** ⇒ *arvo*

Talouslaskentatoiminto, joka laskee investoinnin sisäisen korkokannan.

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

CFLista on lista kassavirtamääristä alun kassavirran CF0 jälkeen.

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

Huomaa: Katso myös **mirr()**, sivu 99.

list1:={6000,⁻8000,2000,⁻3000}	{6000,⁻8000,2000,⁻3000}
list2:={2,2,2,1}	{2,2,2,1}
irr(5000,*list1*,*list2*)	⁻4.64484

isPrime() Luettelo >

isPrime(*Luku***)** ⇒ *Boolen vakiolauseke*

isPrime(5)	true
isPrime(6)	false

isPrime() Luettelo >

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.

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

Funktio, jolla etsitään seuraava jaoton luku määrätyn luvun jälkeen_

```
Define nextprim(n)=Func                    Done
                  Loop
                  n+1→n
                  If isPrime(n)
                  Return n
                  EndLoop
                  EndFunc
nextprim(7)                                11
```

isVoid() Luettelo >

isVoid(*Muutt***)** ⇒ *Boolen vakiolauseke*
isVoid(*Laus***)** ⇒ *Boolen vakiolauseke*
isVoid(*Lista***)** ⇒ *lista Boolen vakiolausekkeista*

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

Lisätietoja tyhjistä elementeistä, katso sivu 223.

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

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                            Done
           Local temp,i
           0→temp
           1→i
           Lbl top
           temp+i→temp
           If i<10 Then
           i+1→i
           Goto top
           EndIf
           Return temp
           EndFunc
g()                                        55
```

lcm() 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*

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

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

left("Hello",2) → "He"

left(*Lista1***[, *Num*])⇒***lista*

Määrittää vasemmanpuoleisimmat *Num*-elementit, jotka sisältyvät listaan *Lista1*.

Jos jätät pois komennon *Num*, määrittää kaiken listasta *Lista1*.

left({1,3,-2,4},3) → {1,3,-2}

left(*Vertailu***)⇒***lauseke*

Laskee yhtälön tai epäyhtälön vasemman puolen.

libShortcut() Katalogi >

libShortcut(*KirjNimiMerkkijono***,**
PikavalNimiMerkkijono
[, *KirjYksitLippu*])⇒*muuttujalista*

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

libShortcut() Katalogi >

Luo muuttujaryhmän nykyiseen ongelmaan, joka sisältää viittauksia kaikkiin määritetyn kirjastoasiakirjan *kirjNimiMerkkijono* sisältämiin objekteihin. Lisää myös ryhmän jäsenet Muuttujat-valikkoon. Tällöin voit viitata kuhunkin objektiin käyttäen sen komentoa *PikavalNimimerkkijono*.

Aseta *KirjYksitLippu*=**0**, kun haluat sulkea pois yksityiset kirjasto-objektit (oletusarvo)
Aseta *KirjYksitLippu*=**1**, kun haluat sisällyttää yksityiset kirjasto-objektit

Muuttujaryhmän kopioiminen, katso **CopyVar** sivulla sivu 26.
Muuttujaryhmän poistaminen, katso **DelVar** sivulla sivu 40.

```
getVarInfo("linalg2")
        [clearmat  "FUNC"  "LibPub "
         gauss1    "PRGM"  "LibPriv "
         gauss2    "FUNC"  "LibPub "]
libShortcut("linalg2","la")
                {la.clearmat,la.gauss2}
libShortcut("linalg2","la",1)
        {la.clearmat,la.gauss1,la.gauss2}
```

LinRegBx Katalogi >

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

Laskee lineaarisen regressiony = a+b ·xlistoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen 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.

LinRegBx

Katalogi >

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: a+b ·x
stat.a, stat.b	Regressiokertoimet
stat.r^2	Määrityskerroin
stat.r	Korrelaatiokerroin
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg vastaava frekvenssilista*

LinRegMx

Katalogi >

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

Laskee lineaarisen regression y = m ·x+b listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen luokkakoodien lista vastaavalle *X*- ja *Y* -datalle.

LinRegMx

Katalogi >

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 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: m ·x+b
stat.m, stat.b	Regressiokertoimet
stat.r^2	Määrityskerroin
stat.r	Korrelaatiokerroin
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg vastaava frekvenssilista*

LinRegtIntervals

Katalogi >

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 154.)**

Kaikkien listojen on oltava samankokoisia.

X ja *Y* ovat riippumattomien ja riippuvien muuttujien listoja.

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

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	**Kuvaus**
stat.RegEqn	Regressioyhtälö: a+b ·x
stat.a, stat.b	Regressiokertoimet
stat.df	Vapausasteet
stat.r^2	Määrityskerroin
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
stat.SEPred	Ennusteen keskivirhe

Tulosmuuttuja	Kuvaus
stat.$\hat{y}$	a + b ·XArvo

LinRegtTest

Katalogi >

LinRegtTest X,Y[,$Frekv$[,$Hypot$]]

Laskee lineaarisen regression X- ja Y-listoista ja suorittaa t-testin kulmakertoimen β ja korrelaatiokertoimen ρ arvosta yhtälölle y=α+βx. Testaa nollahypoteesia H_0:β=0 (vastaavasti, ρ=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:β=ρ=0) testataan.

Kun H_a: β≠0 ja ρ≠0 (oletus), aseta $Hypot$=0
Kun H_a: β<0 ja ρ<0, aseta $Hypot$<0
Kun H_a: β>0 ja ρ>0, aseta $Hypot$>0

Tulosten yhteenveto tallentuu $stat.results$-muuttujaan. (Katso sivu sivu 154.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: a + b ·x
stat.t	t-tilasto merkitsevyystestille
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat.df	Vapausasteet

Tulosmuuttuja	Kuvaus
stat.a, stat.b	Regressiokertoimet
stat.s	Keskivirhe suoran ympärillä
stat.SESlope	Kulmakertoimen keskivirhe
stat.r^2	Määrityskerroin
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.

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

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

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

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

ΔList() Katalogi >

ΔList(*Lista1***)⇒***lista*

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

Määrittää listan, joka sisältää *Lista1*:n peräkkäisten elementtien väliset erotukset. Jokainen *Lista1*:n elementti vähennetään *Lista1*:n seuraavasta elementistä. Tuloksena oleva lista on aina yhden elementin lyhyempi kuin alkuperäinen *Lista1*.

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

list▸mat() Katalogi >

list▸mat(*Lista* **[,** *elementtiäRivillä***])** ⇒*matriisi*

Laskee matriisin, joka on täytetty rivi riviltä *Lista*n elementeillä.

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

Jos *Lista* ei täytä tulosmatriisia, siihen lisätään nollia.

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

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

ln() ctrl e^x painikkeet

ln(*Arvo1***)⇒***arvo*
ln(*Lista1***)⇒***lista*

Määrittää argumentin luonnollisen logaritmin.

Jos kyseessä on lista, määrittää elementtien luonnolliset logaritmit.

```
ln(2.)                     0.693147
```

Jos kompleksilukumuodon tila on Reaali:

```
ln({-3,1.2,5})
"Error: Non-real calculation"
```

Jos kompleksilukumuodon tila on Suorakulma:

ln()

ctrl e^x **painikkeet**

ln(*neliömatriisi1***)⇒***neliömatriisi*

Määrittää *neliömatriisi1*:n matriisin luonnollisen logaritmin. Tämä ei ole sama kuin kunkin elementin luonnollisen logaritmin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

$$\ln(\{-3,1.2,5\})$$
$$\{1.09861+3.14159\cdot i,0.182322,1.60944\}$$

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

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

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

LnReg

Katalogi >

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

Laskee logaritmisen regression y = a+b ·ln(x) listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen 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.

LnReg

Katalogi >

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: a+b ·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 *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg vastaava frekvenssilista*

Local

Katalogi >

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

Määrittää määritetyt *muuttujat* paikallisina muuttujina. Nämä muuttujat ovat olemassa vain funktion sievennyksen aikana, ja ne poistetaan, kun funktion suoritus päättyy.

Huomaa: Paikalliset muuttujat säästävät muistia, koska ne ovat olemassa vain väliaikaisesti. Lisäksi ne eivät häiritse mitään olemassa olevia globaalien muuttujien arvoja. Paikallisia muuttujia on käytettävä **For**-silmukoissa sekä tallennettaessa arvoja väliaikaisesti monirivisessä funktiossa, sillä globaalien muuttujien modifiointeja ei sallita funktiossa.

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

```
Define rollcount()=Func
                   Local i
                   1→i
                   Loop
                   If randInt(1,6)=randInt(1,6)
                   Goto end
                   i+1→i
                   EndLoop
                   Lbl end
                   Return i
                   EndFunc
                                          Done
rollcount()                                 16
rollcount()                                  3
```

Lock

Katalogi >

Lock*Muutt1*[, *Muutt2*] [, *Muutt3*] ...
Lock*Muutt*.

Lukitsee määritetyt muuttujat tai muuttujaryhmän. Lukittuja muuttujia ei voi muokata eikä poistaa.

Et voi lukita tai vapauttaa järjestelmän muuttujaa *Ans*, etkä voi lukita järjestelmän muuttujaryhmiä *stat*. tai *tvm*.

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

Katso **unLock, sivu 172, ja getLockInfo(), sivu 67.**

a:=65	65
Lock a	*Done*
getLockInfo(a)	1
a:=75	"Error: Variable is locked."
DelVar a	"Error: Variable is locked."
Unlock a	*Done*
a:=75	75
DelVar a	*Done*

log()

ctrl 10^x painikkeet

log(*Arvo1*[,*Arvo2*]**)**⇒*arvo***log(***Lista1*[,*Arvo2*]**)**⇒*lista*

Laskee ensimmäisen argumentin kantaluku--*Arvo2*:n logaritmin.

Huomaa: Katso myös **Logaritmimalli**, sivu 2.

Kun kyseessä on lista, laskee elementtien kantaluku--*Arvo2*:n logaritmin.

Jos toinen argumentti jätetään pois, kantalukuna käytetään lukua 10.

$\log_{10}(2.)$	0.30103
$\log_{4}(2.)$	0.5
$\log_{3}(10)-\log_{3}(5)$	0.63093

Jos kompleksilukumuodon tila on Reaali:

$\log_{10}(\{-3,1.2,5\})$	"Error: Non−real calculation"

Jos kompleksilukumuodon tila on Suorakulma:

$\log_{10}(\{-3,1.2,5\})$	$\{0.477121+1.36438\cdot i,0.079181,0.69897\}$

log(*neliömatriisi1*[,*Arvo*]**)**⇒*neliömatriisi*

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

log() ctrl 10^x painikkeet

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

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

Jos kantalukuargumentti jätetään pois, kantalukuna käytetään lukua 10.

$$\log_{10}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$$

$$\begin{bmatrix} 0.795387+0.753438\cdot i & 0.003993-0.6474 \\ 0.194895-0.315095\cdot i & 0.462485+0.2707 \\ -0.115909-0.904706\cdot i & 0.488304+0.7774 \end{bmatrix}$$

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

Logistic Katalogi >

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

Laskee logistisen regressiony = (c/(1+a ·e^-bx))listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen 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 223.

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 *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg vastaava frekvenssilista*

LogisticD

Katalogi >

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 154.)

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 suoritusaika 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 numeeristen tai merkkijonojen luokkakoodien lista vastaavalle *X*- ja *Y* -datalle.

LogisticD

Katalogi >

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 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: $c/(1+a \cdot e^{-bx})+d)$
stat.a, stat.b, stat.c, stat.d	Regressiokertoimet
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun *X Lista*n sisältämä datapisteiden lista, jota käytetään todellisesti regressiossa komentojen *Frekv*, *Luokka Lista* ja *Sisällytä Luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*n sisältämä datapisteiden lista, jota käytetään todellisesti regressiossa komentojen *Frekv*, *Luokka Lista* ja *Sisällytä Luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg vastaava frekvenssilista*

Loop (Silmukka)

Katalogi >

Loop
Lohko
EndLoop

Suorittaa toistuvasti *Lohko*n sisältämät lausekkeet. Huomaa, että silmukkaa suoritetaan loputtomasti, ellei **Goto**- tai **Exit**-ohjetta suoriteta *Lohko*n sisällä.

Lohko on lausekkeiden sarja, jotka on erotettu toisistaan kaksoispisteellä (:).

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

```
Define rollcount()=Func
                Local i
                1→i
                Loop
                If randInt(1,6)=randInt(1,6)
                Goto end
                i+1→i
                EndLoop
                Lbl end
                Return i
                EndFunc
                                        Done
rollcount()                              16
rollcount()                               3
```

LU Katalogi >

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 · *uMatriisi* = *pMatriisi* · *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 ·max(dim(*Matriisi*)) ·riviNorm (*Matriisi*)

LU-dekomponointialgoritmi käyttää osittaista rivien vaihtoa (pivoting).

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

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

lower $\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$

upper $\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$

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

M

mat▸list() Katalogi >

mat▸list(*Matriisi***)⇒***lista*

Luo listan, joka on täytetty *Matriisi*n elementeillä. Elementit kopioidaan *Matriisi*sta rivi riviltä.

mat▸list([1 2 3]) — {1,2,3}

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

mat▸list(*m1*) — {1,2,3,4,5,6}

mat▸list() Katalogi >

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

max() Katalogi >

max(*Arvo1***, *Arvo2***)⇒***lauseke***
max(*Lista1***, *Lista2***)⇒***lista***
max(*Matriisi1***, *Matriisi2***)⇒***matriisi***

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

$\max(2.3,1.4)$ → 2.3

$\max(\{1,2\},\{-4,3\})$ → $\{1,3\}$

max(*Lista***)⇒***lauseke***

Laskee *lista*:n maksimielementin.

$\max(\{0,1,-7,1.3,0.5\})$ → 1.3

max(*Matriisi1***)⇒***matriisi***

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

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

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

Huomaa: Katso myös **min()**.

mean() Katalogi >

mean(*Lista***[,***frekvLista***])⇒***lauseke***

Laskee *Lista*n sisältämien elementtien keskiarvon.

$\mathrm{mean}(\{0.2,0,1,-0.3,0.4\})$ → 0.26

$\mathrm{mean}(\{1,2,3\},\{3,2,1\})$ → $\frac{5}{3}$

Jokainen *frekvLista*n elementti näyttää *Lista*n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

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

Luo rivivektorin kaikkien *Matriisi1*:n sarakkeiden keskiarvoista.

Suorakulmavektorimuodossa:

mean() Katalogi >

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

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

$$\text{mean}\left(\begin{bmatrix}0.2 & 0\\ -1 & 3\\ 0.4 & -0.5\end{bmatrix}\right) \qquad \begin{bmatrix}-0.133333 & 0.833333\end{bmatrix}$$

$$\text{mean}\left(\begin{bmatrix}\frac{1}{5} & 0\\ -1 & 3\\ \frac{2}{5} & \frac{-1}{2}\end{bmatrix}\right) \qquad \begin{bmatrix}\frac{-2}{15} & \frac{5}{6}\end{bmatrix}$$

$$\text{mean}\left(\left\{\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix},\begin{bmatrix}5 & 3\\ 4 & 1\\ 6 & 2\end{bmatrix}\right\}\right) \qquad \begin{bmatrix}\frac{47}{15} & \frac{11}{3}\end{bmatrix}$$

median() Katalogi >

median(*Lista***[, *frekvLista*])⇒*lauseke***

Laskee *Lista*n elementtien mediaanin.

Jokainen *frekvLista*n elementti näyttää *Lista*n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

$$\text{median}(\{0.2,0,1,-0.3,0.4\}) \qquad 0.2$$

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

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

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

$$\text{median}\left(\begin{bmatrix}0.2 & 0\\ 1 & -0.3\\ 0.4 & -0.5\end{bmatrix}\right) \qquad \begin{bmatrix}0.4 & -0.3\end{bmatrix}$$

Huomaa:

- Kaikkien listan tai matriisin syötteiden tulee sieventyä luvuiksi.
- Listassa tai matriisissa olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

MedMed Katalogi >

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

Laskee mediaani-mediaani-suorany = (m ·x+b)listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen 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 223.

Tulosmuuttuja	**Kuvaus**
stat.RegEqn	Mediaani-mediaani-suoran yhtälö: m ·x+b
stat.m, stat.b	Mallin kertoimet
stat.Resid	Mediaani-mediaani-suoran jäännökset
stat.XReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg 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 *lähdemerkkijono*sta aloittaen merkistä numero *Alku*.

Count:n on oltava ≥ 0. Jos *Count* = 0, antaa vastauksena tyhjän merkkijonon.

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

mid(*lähdelista***,** *Alku* **[,** *Count***])**⇒*lista*

Laskee *Count*:n elementit *lähdelista*sta aloittaen elementistä numero *Alku*.

Jos *Count* jätetään pois, tai jos se on suurempi kuin *lähdelista*, laskee kaikki elementit *lähdelista*sta aloittaen elementistä numero *Alku*.

Count:n on oltava ≥ 0. Jos Count = 0, antaa vastauksena tyhjän listan.

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(*lähdeMerkkijonoLista***,** *Alku***[,** *Count***])** ⇒*lista*

Laskee *Count*:n merkkijonot merkkijonolistasta *lähdeMerkkijonoLista* aloittaen elementistä numero *Alku*.

mid({"A","B","C","D"},2,2)	{"B","C"}

min() Katalogi >

min(*Arvo1***,** *Arvo2***)**⇒*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(2.3,1.4)	1.4
min({1,2},{-4,3})	{-4,2}

min(*Lista***)**⇒*lauseke*

Laskee *Lista*n minimielementin.

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

min() Katalogi >

min(*Matriisi1***)⇒***matriisi*

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

Huomaa: Katso myös **max().**

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

mirr() Katalogi >

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

Talouslaskentatoiminto, joka laskee investoinnin modifioidun sisäisen korkokannan.

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 ryhmitetylle (peräkkäiselle) kassavirtamäärälle, joka on *CFLista*n vastaava elementti. Oletusarvo on 1; jos syötät arvoja, niiden on oltava positiivisia kokonaislukuja < 10,000.

Huomaa: Katso myös **irr()**, sivu 78.

$list1:=\{6000,-8000,2000,-3000\}$	$\{6000,-8000,2000,-3000\}$
$list2:=\{2,2,2,1\}$	$\{2,2,2,1\}$
$\text{mirr}(4.65,12,5000,list1,list2)$	13.41608607

mod() Katalogi >

mod(*Arvo1***,** *Arvo2***)⇒***lauseke*
mod(*Lista1***,** *Lista2***)⇒***lista*
mod(*Matriisi1***,** *Matriisi2***)⇒***matriisi*

Laskee ensimmäisen argumentin modulo toinen argumentti identtisten yhtälöiden määrittelyn mukaisesti:

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

mod() Katalogi >

mod(x,0) = x
mod(x,y) = x − y floor(x/y)

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

Jos argumentteina on kaksi listaa tai matriisia, laskee listan tai matriisin, joka sisältää vastaavien elementtien kunkin parin modulon (jakojäännöksen).

Huomaa: Katso myös **remain()**, sivu 131

mRow() Katalogi >

mRow(*Arvo***,** *Matriisi1***,** *Indeksi***)⇒***matriisi*

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

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

mRowAdd() Katalogi >

mRowAdd(*Arvo***,** *Matriisi1***,** *Indeksi1***,** *Indeksi2***) ⇒***matriisi*

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

Arvo · *rivi Indeksi1* + *rivi Indeksi2*

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

MultReg Katalogi >

MultReg *Y***,** *X1*[**,***X2*[**,***X3***,…**[,***X10*]]]

Laskee listan *Y* moninkertaisen lineaarisen regression listojen *X1*, *X2*, …, *X10* suhteen. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

Kaikkien listojen on oltava samankokoisia.

MultReg

Katalogi >

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: b0+b1 ·x1+b2 ·x2+ ...
stat.b0, stat.b1, ...	Regressiokertoimet
stat.R^2	Moninkertaisen määrityksen kerroin
stat.ŷLista	ŷLista = b0+b1 ·x1+ ...
stat.Resid	Regressioyhtälön jäännökset

MultRegIntervals

Katalogi >

MultRegIntervals *Y*, *X1*[,*X2*[,*X3*,...[,*X10*]]],*XArvoLista*[,*CTaso*]

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

Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

Kaikkien listojen on oltava samankokoisia.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: b0+b1 ·x1+b2 ·x2+ ...
stat.ŷ	Pisteen A arvio: ŷ = b0 + b1 · xl + ... *XArvoListalle*
stat.dfError	Virheen vapausasteet
stat.CLower, stat.CUpper	Keskiarvovasteen luottamusväli
stat.ME	Luottamusvälin virhemarginaali
stat.SE	Keskiarvovasteen keskivirhe
stat.LowerPred, stat.UpperrPred	Yhden havainnon ennusteväli

Tulosmuuttuja	Kuvaus
stat.MEPred	Ennustevälin virhemarginaali
stat.SEPred	Ennusteen keskivirhe
stat.bList	Regressiokertoimien lista, {b0,b1,b3,...}
stat.Resid	Regressioyhtälön jäännökset

MultRegTests

Katalogi >

MultRegTests Y, $X1$[,$X2$[,$X3$,...[,$X10$]]]

Moninkertaisen lineaarisen regression testi laskee lineaarisen regression tietystä datasta ja määrittää kertoimille globaalin F-testin tilastot sekä t-testin tilastot.

Tulosten yhteenveto tallentuu $stat.results$-muuttujaan. (Katso sivu sivu 154.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

`Tulokset`

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: b0+b1 ·x1+b2 ·x2+ ...
stat.F	Globaalin F-testin tilasto
stat.PVal	Globaaliin F-tilastoon liittyvä P-arvo
stat.R^2	Moninkertaisen määrityksen kerroin
stat.AdjR^2	Moninkertaisen määrityksen säädetty kerroin
stat.s	Virheen keskihajonta
stat.DW	Durbin-Watsonin tilasto; käytetään määritettäessä, esiintyykö mallissa ensimmäisen asteen automaattista korrelaatiota
stat.dfReg	Regression vapausasteet
stat.SSReg	Regression neliöiden summa
stat.MSReg	Regression keskineliö
stat.dfError	Virheen vapausasteet
stat.SSError	Virheen neliöiden summa

Tulosmuuttuja	Kuvaus
stat.MSError	Virheen keskineliö
stat.bList	{b0,b1,…} Kertoimien lista
stat.tList	t-tilastojen lista, yksi kullekin bListan sisältämälle kertoimelle
stat.PList	P-arvojen lista kullekin t-tilastolle
stat.SEList	Keskivirheiden lista bListan sisältämille kertoimille
stat.ŷ Lista	ŷ Lista = b0+b1 ·x1+ . . .
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

BoolenLaus1 **nand** *BoolenLaus2* antaa vastauksena *Boolen lausekkeenBoolenList1* **nand** *BoolenList2* antaa vastauksena *Boolen listanBoolenMatriisi1* **nand** *BoolenMatriisi2* antaa vastauksena *Boolen 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.

Kokonaisluku1 **nand** *Kokonaisluku2⇒kokonaisluku*

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}

nand (ei-ja)

ctrl = näppäimet

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

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

nCr()

Katalogi >

nCr(*Arvo1***,** *Arvo2***)⇒***lauseke*

Kokonaisluvulle *Arvo1* ja *Arvo2* sekä *Arvo1* ≥ *Arvo2* ≥ 0, **nCr()** on *Arvo1*:n asioiden kombinaatioiden lukumäärä, joita otetaan *Arvo2*:n verran kerrallaan. (Tästä käytetään myös nimitystä binomikerroin.)

$\mathrm{nCr}(z,3)\|z=5$	10
$\mathrm{nCr}(z,3)\|z=6$	20

nCr(*Arvo***, 0)⇒1**

nCr(*Arvo***,** *negKokonaisluku***)⇒0**

nCr(*Arvo***,** *posKokonaisluku***)⇒** *Arvo* · **(***Arvo***−1)... (***Arvo***−***posKokonaisluku***+1)/** *posKokonaisluku***!**

nCr(*Arvo***,** *eiKokonaisluku***)⇒***lauseke***!/ ((** *Arvo***−***eiKokonaisluku***)! ·***eiKokonaisluku***!)**

nCr(*Lista1***,** *Lista2***)⇒***lista*

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

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

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

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

nCr() Katalogi >

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

nDerivative() Katalogi >

nDerivative(*Laus1***,***Muutt=Arvo***[,***Aste***])** ⇒*arvo*

nDerivative(*Laus1***,***Muutt***[,***Aste***])** | *Muutt=Arvo*⇒*arvo*

Laskee numeerisen derivaatan käyttäen automaattisia derivointimenetelmiä.

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

Jos muuttuja *Muutt* ei sisällä numeerista arvoa, *Arvo* on annettava.

*Derivaatan Aste*en on oltava **1** tai **2**.

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

Huomaa: nDerivaatta()-algoritmiin liittyy rajoitus: se laskee rekursiivisesti sieventämättömästä lausekkeesta ensimmäisen (ja toisen, mikäli mahdollista) derivaatan numeerisen arvon ja sieventää jokaisen alalausekkeen, mistä voi olla tuloksena odottamaton vastaus.

Tarkastele oikealla olevaa esimerkkiä. Yhtälön x·(x^2+x)^(1/3), kun x=0, ensimmäinen derivaatta on yhtä kuin 0. Koska alalausekkeen (x^2+x)^(1/3) ensimmäinen derivaatta kuitenkin on määrittämätön pisteessä x=0, ja tällä arvolla lasketaan koko lausekkeen derivaatta, **nDerivaatta()** ilmaisee vastauksen määrittämättömänä ja näyttää varoitusviestin.

Jos tämä rajoitus esiintyy, varmista ratkaisu graafisesti. Voit myös kokeilla funktiota **centralDiff()**.

$\text{nDerivative}\left(x\cdot\left(x^2+x\right)^{\frac{1}{3}}, x, 1\right)\lvert x=0$	undef
$\text{centralDiff}\left(x\cdot\left(x^2+x\right)^{\frac{1}{3}}, x\right)\lvert x=0$	0.000033

newList() Katalogi >

newList(*numElementit***)⇒***lista*

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

newList(4)	{0,0,0,0}

newMat() Katalogi >

newMat(*numRivit***, *numSarakkeet***)** ⇒*matriisi*

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

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

nfMax() Katalogi >

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

Laskee muuttujan *Muutt* numeerisen ehdotusarvon, jossa lausekkeen *Laus* paikallinen maksimi esiintyy.

Jos ilmoitat *alarajan* ja *ylärajan*, funktio etsii suljetulta väliltä [*alaraja*,*yläraja*] paikallista maksimia.

$\text{nfMax}(-x^2-2\cdot x-1,x)$	-1.
$\text{nfMax}(0.5\cdot x^3-x-2,x,-5,5)$	5.

nfMin() Katalogi >

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

Laskee muuttujan *Muutt* numeerisen ehdotusarvon, jossa lausekkeen *Laus* paikallinen minimi esiintyy.

Jos ilmoitat *alarajan* ja *ylärajan*, funktio etsii suljetulta väliltä [*alaraja*,*yläraja*] paikallista minimia.

$\text{nfMin}(x^2+2\cdot x+5,x)$	-1.
$\text{nfMin}(0.5\cdot x^3-x-2,x,-5,5)$	-5.

nInt() Katalogi >

nInt(*Laus1, Muutt, Ala, Ylä***)⇒***lauseke*

Jos integroitava funktio *Laus1* ei sisällä muita muuttujia kuin *Muutt*, ja jos *Ala* ja *Ylä* ovat vakioita, positiivinen ∞ tai negatiivinen ∞, tällöin **nInt()** laskee likiarvon lausekkeesta ∫(*Laus1*, *Var*, *Ala*, *Ylä*). Tämä likiarvo on integrandin välillä *Ala*<*Muutt*<*Ylä* olevien joidenkin otosarvojen painotettu keskiarvo.

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

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

$\text{nInt}\left(\cos(x),x,-\pi,\pi+1.\text{E}-12\right)$ $-1.04144\text{E}-12$

Näkyviin tulee viesti (Questionable accuracy (Kyseenalainen tarkkuus)), kun näyttää siltä, että tavoitetta ei ole saavutettu.

Sijoita **nInt()**-komentoa sisäkkäin, jos haluat suorittaa moninkertaisen numeerisen integroinnin. Integroinnin raja-arvot voivat riippua niiden ulkopuolella olevista integrointimuuttujista.

$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x\cdot y}}{\sqrt{x^2-y^2}},y,-x,x\right),x,0,1\right)$ 3.30423

nom() Katalogi >

nom(*efektiivinenKorko,CpY***)⇒***arvo*

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

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

$\text{nom}(5.90398,12)$ 5.75

Huomaa: Katso myös **eff()**, sivu 46.

nor (eikä) ctrl = näppäimet

BoolenLaus1 **nor** *BoolenLaus2* antaa vastauksena *Boolen*

lausekkeenBoolenList1 **nor** *BoolenList2* antaa vastauksena *Boolen*

nor (eikä)

ctrl = näppäimet

listanBoolenMatriisi1 **nor***BoolenMatriisi2* antaa vastauksena *Boolen matriisin*

Antaa vastauksena loogisen **or** operaation negaation kahdesta argumentista. Antaa vastauksena totuusarvon tosi, epätosi tai yhtälön sievennetyn muodon.

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Kokonaisluku1 **nor***Kokonaisluku2*⇒*kokonaisluku*

Vertaa kahta reaalikokonaislukua bitti bitiltä **nor**-operaation avulla. Sisäisesti kumpikin kokonaisluku muunnetaan etumerkilliseksi, 64 bitin binaariluvuksi. Kun vastaavia bittejä verrataan, tulos on 1, jos kumpikin bitti on 1. Muussa tapauksessa tulos on 0. Laskettu arvo edustaa bittituloksia ja se näkyy 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).

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

norm()

Katalogi >

norm(*Matriisi***)**⇒*lauseke*

norm(*Vektori***)**⇒*lauseke*

Laskee Frobeniusin normin.

$\text{norm}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	5.47723
$\text{norm}\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right)$	2.23607
$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	2.23607

normCdf()

Katalogi >

normCdf(*alaraja***,***yläraja***[,**μ**[,**σ**]])**⇒*luku*, jos *yläraja* ja *alaraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

normCdf() Katalogi >

Laskee normaalijakauman todennäköisyyden *alaraja*n ja *yläraja*n välillä määritetylle μ:lle (oletus=0) ja σ:lle (oletus=1).

Kun P(X ≤ *yläraja*), aseta *alaraja* = **−9E**999.

normPdf() Katalogi >

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

Laskee normaalijakauman pistetodennäköisyysfunktion määritetyssä *XArvo*ssa määritetylle μ:lle ja σ:lle.

not (ei) Katalogi >

not *BoolenLaus*⇒*Boolen lauseke*

Määrittää totuusarvon tosi, epätosi tai argumentin sievennetyn muodon.

not *Kokonaisluku1*⇒*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 16.

not (2≥3)	true
not 0hB0▶Base16	0hFFFFFFFFFFFFFF4F
not not 2	2

Heksadesimaalisessa kantalukutilassa:

not 0h7AC36	0hFFFFFFFFFFF853C9

Binaarisessa kantalukutilassa:

0b100101▶Base10	37
not 0b100101	
0b1111111111111111111111111111111▶	
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(*Arvo1***,** *Arvo2***)⇒***lauseke*

Kokonaisluvulle *Arvo1* ja *Arvo2* sekä *Arvo1* ≥ *Arvo2* ≥ 0, **nPr()** on *Arvo1*:n asioiden permutaatioiden lukumäärä, joita otetaan *Arvo2*:n verran kerrallaan.

$\text{nPr}(z,3)\|z=5$	60
$\text{nPr}(z,3)\|z=6$	120
$\text{nPr}(\{5,4,3\},\{2,4,2\})$	$\{20,24,6\}$
$\text{nPr}\left(\begin{bmatrix}6&5\\4&3\end{bmatrix},\begin{bmatrix}2&2\\2&2\end{bmatrix}\right)$	$\begin{bmatrix}30&20\\12&6\end{bmatrix}$

nPr(*Arvo***, 0)⇒1**

nPr(*Arvo***,** *negKokonaisluku***)⇒ 1/ ((***Arvo***+1) ·(***Arvo***+2)...** **(***Arvo***−***negKokonaisluku***))**

nPr(*Arvo***,** *posKokonaisluku***)⇒** *Arvo* **·** **(***Arvo***−1)... (***Arvo***−***posKokonaisluku***+1)**

nPr(*Arvo***,** *eiKokonaisluku***)⇒***Arvo***! /** **(***Arvo***−***eiKokonaisluku***)!**

nPr(*Lista1***,** *List2***)⇒***lista*

$\text{nPr}(\{5,4,3\},\{2,4,2\})$	$\{20,24,6\}$

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

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

$\text{nPr}\left(\begin{bmatrix}6&5\\4&3\end{bmatrix},\begin{bmatrix}2&2\\2&2\end{bmatrix}\right)$	$\begin{bmatrix}30&20\\12&6\end{bmatrix}$

Luo matriisin permutaatioista kahden matriisin sisältämiin vastaaviin elementtipareihin perustuen. Argumenttien on oltava samankokoisia matriiseja.

npv() Katalogi >

npv(*Korkoprosentti***,***CFO***,***CFLista*
[**,***CFFrekv***])**

$list1:=\{6000,^-8000,2000,^-3000\}$	$\{6000,^-8000,2000,^-3000\}$
$list2:=\{2,2,2,1\}$	$\{2,2,2,1\}$
$\text{npv}(10,5000,list1,list2)$	4769.91

Talouslaskentatoiminto, joka laskee nettonykyarvon; tulevien ja poistuvien kassavirtojen nykyisten arvojen summan. Jos npv:n tulos on positiivinen, investointi on kannattava.

Korkoprosentti on prosentti, jolla kassavirtoja (rahan kustannusta) vähennetään yhden jakson aikana.

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

npv() Katalogi >

CFLista on lista kassavirtamääristä ensimmäisen kassavirran *CF0* jälkeen.

CFFrekv on lista, jossa jokainen elementti määrittää esiintymisfrekvenssin ryhmitetylle (peräkkäiselle) kassavirtamäärälle, joka on *CFLista*n vastaava elementti. Oletusarvo on 1; jos syötät arvoja, niiden on oltava positiivisia kokonaislukuja < 10,000.

nSolve() Katalogi >

nSolve(*Yhtälö*,*Muutt***[=***Arvaus***])**⇒*luku tai virhe_merkkijono*

nSolve(*Yhtälö*,*Muutt***[=***Arvaus***]**,*alaraja***)** ⇒*luku tai virhe_merkkijono*

nSolve(*Yhtälö*,*Muutt* **[=***Arvaus***]**,*alaraja*,*yläraja***)** ⇒*luku tai virhe_merkkijono*

nSolve(*Yhtälö*,*Muutt***[=***Arvaus***]) |** *alaraja* ≤*Muutt*≤*yläraja* ⇒*luku tai virhe_merkkijono*

Etsii iteratiivisesti yhtä likimääräistä numeerista ratkaisua *Yhtälö*n yhdelle muuttujalle. Määritä muuttuja seuraavasti:

muuttuja
– tai –
muuttuja = *reaaliluku*

Esimerkiksi x kelpaa ja samoin x=3.

$\text{nSolve}(x^2+5\cdot x-25=9,x)$	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.

nSolve() yrittää määrittää joko yhden pisteen, jossa jäännös on nolla, tai kaksi toisiaan suhteellisen lähellä olevaa pistettä, jossa jäännöksen etumerkki on vastakkainen ja jäännöksen magnitudi ei ole liian suuri. Jos funktio ei pysty määrittämään tätä kohtuullisella otospisteiden määrällä, se antaa vastauksena merkkijonon “no solution found” (yhtään ratkaisua ei löydy).

$\text{nSolve}(x^2+5\cdot x-25=9,x)\|x<0$	-8.84429
$\text{nSolve}\left(\frac{(1+r)^{24}-1}{r}=26,r\right)\|r>0 \text{ and } r<0.25$	0.006886
$\text{nSolve}(x^2=-1,x)$	"No solution found"

O

OneVar [**1,**]*X*[**,**[*Frekv*][**,***Luokka***,***Sisällytä*]]

OneVar [*n***,**]*X1***,***X2*[*X3*[**,**...[**,***X20*]]]

Laskee yhden muuttujan tilaston enintään 20 listasta. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 tyhjistä elementeistä, katso sivu 223.

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

Tulosmuuttuja	Kuvaus
stat.Q_1X	x:n ensimmäinen neljännes
stat.MedianX	x:n mediaani
stat.Q_3X	x:n kolmas neljännes
stat.MaxX	x:n arvojen maksimi
stat.SSX	x:n keskiarvon poikkeamien neliöiden summa

or (tai)

Katalogi >

BoolenLaus1 **or** *BoolenLaus2* antaa vastauksena *Boolen lausekkeen* *BoolenList1* **or** *BoolenList2* antaa vastauksena *Boolen listan* *BoolenMatriisi1* **or** *BoolenMatriisi2* antaa vastauksena *Boolen matriisin*

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

Antaa vastauksen tosi, jos jompikumpi tai molemmat lausekkeet ovat tosia. Antaa vastauksen epätosi, jos kumpikin lauseke on epätosi.

Huomaa: Katso **xor**.

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

```
Define g(x)=Func                    Done
            If x≤0 or x≥5
            Goto end
            Return x·3
            Lbl end
            EndFunc
g(3)                                9
g(0)        A function did not return a value
```

Kokonaisluku1 **or** *Kokonaisluku2* ⇒ *kokonaisluku*

Vertaa kahta reaalikokonaislukua bitti bitiltä or-operaation avulla. Sisäisesti kumpikin kokonaisluku muunnetaan etumerkilliseksi, 64 bitin binaariluvuksi. Kun vastaavia bittejä verrataan, tulos on 1, jos jompikumpi bitti on 1; tulos on 0 vain, jos kumpikin bitti on 0. Laskettu arvo edustaa bittituloksia, ja se näkyy kantalukutilan mukaisesti.

Heksadesimaalisessa kantalukutilassa:

```
0h7AC36 or 0h3D5F                   0h7BD7F
```

Tärkeää: Nolla, ei O-kirjain.

Binaarisessa kantalukutilassa:

```
0b100101 or 0b100                   0b100101
```

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

or (tai) Katalogi >

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.

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 16.

Huomaa: Katso **xor.**

ord() Katalogi >

ord(*Merkkijono***)⇒***kokonaisluku*
ord(*Lista1***)⇒***lista*

Määrittää merkkijonon *Merkkijono* ensimmäisen merkin numerokoodin tai luo listan jokaisen listaelementin ensimmäisistä merkeistä.

ord("hello")	104
char(104)	"h"
ord(char(24))	24
ord({"alpha","beta"})	{97,98}

P

P▶Rx() Katalogi >

P▶Rx(*rLaus***, θ***Laus***)⇒***lauseke*
P▶Rx(*rLista***, θ***Lista***)⇒***lista*
P▶Rx(*rMatriisi***, θ***Matriisi***)⇒***matriisi*

Määrittää paria (r, θ) vastaavan x-koordinaatin.

Radiaanikulmatilassa:

P▶Rx(4,60°)	2.
P▶Rx({-3,10,1.3},{π/3,-π/4,0})	{-1.5,7.07107,1.3}

P▶Rx() Katalogi >

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(...)**.

P▶Ry() Katalogi >

P▶Ry(*rArvo***, θ***Arvo***)⇒***arvo***
P▶Ry(*rLista***, θ***Lista***)⇒***lista***
P▶Ry(*rMatriisi***, θ***Matriisi***)⇒***matriisi***

Määrittää paria (r, θ) vastaavan y-koordinaatin.

Huomaa: Argumentti θ tulkitaan joko aste-, radiaani- tai graadikulmaksi valittuna olevan kulmatilan mukaisesti.

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

Radiaanikulmatilassa:

$$P\blacktriangleright Ry(4,60°) \qquad 3.4641$$

$$P\blacktriangleright Ry\left(\{-3,10,1.3\},\left\{\frac{\pi}{3},\frac{-\pi}{4},0\right\}\right)$$
$$\{-2.59808,-7.07107,0\}$$

PassErr Katalogi >

PassErr

Ohittaa virheen siirtyen seuraavalle tasolle.

Jos järjestelmän muuttuja *errCode* on nolla, **PassErr** ei tee mitään.

Else-lauseessa lohkossa **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.

Esimerkki **PassErr**-komennosta, katso esimerkki 2 **Try**-komennon kohdalla, sivu 166.

PassErr

Katalogi >

Huomaa: Katso myös **ClrErr**, sivu 24, ja **Try**, sivu 166.

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

piecewise()

Katalogi >

piecewise(*Laus1* [**,** *Ehto1* [**,** *Laus2* [**,** *Ehto2* [**,** ...]]]]**)**

Laatii määritelmät paloittain määritellylle funktiolle listan muodossa. Voit luoda paloittain määriteltyjen funktioiden määrityksiä myös mallin avulla.

Huomaa: Katso myös **Paloittain määritellyn funktion malli**, sivu 3.

Define $p(x)=\begin{cases} x, & x>0 \\ \text{undef}, & x\le 0 \end{cases}$	*Done*
$p(1)$	1
$p(-1)$	undef

poissCdf()

Katalogi >

poissCdf(λ**,***alaraja***,***yläraja***)**⇒*luku*, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

poissCdf(λ**,***yläraja***)**(kun P(0≤X≤*yläraja*) ⇒*luku*, jos *yläraja* on luku, lista, jos *yläraja* on lista

Laskee kumulatiivisen todennäköisyyden diskreetille Poissonin jakaumalle, jolla on määritetty keskiarvo λ.

Kun P(X ≤ *yläraja*), aseta *alaraja*=0

poissPdf()

Katalogi >

poissPdf(λ**,***XArvo***)**⇒*luku*, jos *XArvo* on luku, *lista*, jos *XArvo* on lista

Laskee todennäköisyyden diskreetille Poissonin jakaumalle, jolla on määritetty keskiarvo λ.

▶Polar (▶Polaarinen) Katalogi >

Vektori ▶**Polar**

[1 3.]▶Polar	[3.16228 ∠71.5651]

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

Näyttää *vektorin* polaarisessa muodossa [r ∠θ]. 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 128.

kompleksiArvo ▶**Polaarinen**

Näyttää *kompleksiVektori*n polaarisessa muodossa.

- Astekulmatilassa vastauksena on (r∠θ).
- Radiaanikulmatilassa vastauksena on $re^{i\theta}$.

*kompleksiArvo*lla voi olla mikä tahansa kompleksilukumuoto. Syöte $re^{i\theta}$ aiheuttaa kuitenkin virheen astekulmatilassa.

Huomaa: Polaarisessa syötteessä (r∠θ) on käytettävä sulkeita.

Radiaanikulmatilassa:

(3+4·*i*)▶Polar	$e^{.927295\cdot i}\cdot 5$
$\left(\left\{4\angle\frac{\pi}{3}\right\}\right)$▶Polar	$e^{1.0472\cdot i}\cdot 4.$

Graadikulmatilassa:

(4·*i*)▶Polar	(4∠ 100.)

Astekulmatilassa:

(3+4·*i*)▶Polar	(5 ∠ 53.1301)

polyEval() Katalogi >

polyEval(*Lista1***,** *Laus1***)⇒***lauseke*
polyEval(*Lista1***,** *Lista2***)⇒***lauseke*

polyEval({1,2,3,4},2)	26
polyEval({1,2,3,4},{2,-7})	{26,-262}

Tulkitsee ensimmäisen argumentin laskeva-asteisen polynomin kertoimeksi ja antaa vastauksena polynomin, josta on laskettu toisen argumentin arvo.

polyRoots() Katalogi >

polyRoots(*Poly*,*Muutt***)** ⇒*lista*

polyRoots(*Kertoinlista***)** ⇒*lista*

Ensimmäinen syntaksi, **polyRoots (***Poly*,*Muutt***)**, laskee polynomin *Poly* reaalilukujuurten listan muuttujan *Muutt* suhteen. Jos reaalilukujuurta ei ole, tuloksena on tyhjä lista: { }.

Poly on oltava polynomi laajennetussa muodossa yhdessä muuttujassa. Älä käytä laajentamattomia muotoja, kuten $y^2 \cdot y+1$ *tai* $x \cdot x+2 \cdot x+1$

Toinen syntaksi, **polyRoots(***Kertoinlista***)**, askee reaalilukujuurten listan kertoimille, jotka sisältyvät *Kertoinlistaan.*

Huomaa: Katso myös **cPolyRoots()**, sivu 32.

polyRoots(y^3+1,y)	{-1}
cPolyRoots(y^3+1,y)	{-1,0.5−0.866025·*i*,0.5+0.866025·*i*}
polyRoots($x^2+2 \cdot x+1,x$)	{-1,-1}
polyRoots({1,2,1})	{-1,-1}

PowerReg Katalogi >

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

Laskee potenssiregressiony = (a ·(x)^b) listoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen luokkakoodien lista vastaavalle *X*- ja *Y* -datalle.

PowerReg

Katalogi >

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 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: a ·(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 *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg 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.

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

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 >

$proggcd(4560,450)$

```
450 60
60 30
30 0
GCD=30
```

Done

prodSeq() — Katso Π(), sivu 196.

Product (PI) — Katso Π(), sivu 196.

product() — Katalogi >

product(*Lista***[, *Alku*[, *loppu*]])⇒*lauseke***

Laskee *Lista*n sisältämien elementtien tulon. *Alku* ja *Loppu* ovat valinnaisia. Ne määrittävät elementtien alueen.

$product(\{1,2,3,4\})$ → 24

$product(\{4,5,8,9\},2,3)$ → 40

product(*Matriisi1***[, *Alku*[, *loppu*]])⇒*matriisi***

Laskee rivivektorin, joka sisältää *Matriisi1*:n sarakkeiden elementtien tulot. *Alku* ja *Loppu* ovat valinnaisia. Ne määrittävät rivialueen.

$product\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\right)$ → $\begin{bmatrix}28&80&162\end{bmatrix}$

$product\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},1,2\right)$ → $\begin{bmatrix}4&10&18\end{bmatrix}$

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

propFrac() — Katalogi >

propFrac(*Arvo1***[, *Muutt*])⇒*arvo***

propFrac(*rationaali_luku***)** laskee *rationaali_luvun* kokonaisluvun summana ja murtolukuna, jolla on sama etumerkki ja suurempi nimittäjä kuin osoittaja.

$propFrac\left(\frac{4}{3}\right)$ → $1+\frac{1}{3}$

$propFrac\left(\frac{-4}{3}\right)$ → $-1-\frac{1}{3}$

propFrac() Katalogi >

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* samanlaiset 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.

propFrac()-funktion avulla voit esittää sekalukuja ja suorittaa sekalukujen yhteen- ja vähennyslaskua.

$\text{propFrac}\left(\frac{11}{7}\right)$	$1+\frac{4}{7}$
$\text{propFrac}\left(3+\frac{1}{11}+5+\frac{3}{4}\right)$	$8+\frac{37}{44}$
$\text{propFrac}\left(3+\frac{1}{11}-\left(5+\frac{3}{4}\right)\right)$	$^-2-\frac{29}{44}$

Q

QR Katalogi >

QR *Matriisi*, *qMatNimi*, *rMatNimi*[, *Tol*]

Suorittaa Householderin QR-dekomponoinnin reaaliluku- tai kompleksilukumatriisista. Tuloksena olevat Q- ja R-matriisit tallentuvat määritettyihin *MatNimi*-muuttujiin. Q-matriisi on jakamaton. R-matriisi on yläkolmiomatriisi.

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

Liukuluku (9.) muuttujassa m1 aiheuttaa sen, että vastaukset lasketaan liukulukumuodossa.

$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9.\end{bmatrix}\rightarrow m1$	$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9.\end{bmatrix}$
QR *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	$\begin{bmatrix}8.12404&9.60114&11.0782\\0.&0.904534&1.80907\\0.&0.&0.\end{bmatrix}$

QR

Katalogi >

- 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 ·max(dim(*Matriisi*)) ·rowNorm (*Matriisi*)

QR-hajotus lasketaan numeerisesti Householderin transformaatioiden avulla. Symbolinen ratkaisu lasketaan Gram-Schmidtin menetelmällä. Matriisin *qMatNimi* sarakkeet ovat ne ortonormitetut perusvektorit, jotka sijoittuvat *matriisi*n määrittämään tilaan.

QuadReg

Katalogi >

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

Laskee 2. asteen polynomiregressiony = a ·x^2+b ·x+clistoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen 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.

QuadReg

Katalogi >

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.RegEqn	Regressioyhtälö: a ·x^2+b ·x+c
stat.a, stat.b, stat.c	Regressiokertoimet
stat.R^2	Määrityskerroin
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg vastaava frekvenssilista*

QuartReg

Katalogi >

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

Laskee 4. asteen polynomiregression y = a ·x^4+b ·x^3+c · x^2+d ·x+elistoista *X* ja *Y* frekvenssillä *Frekv*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu sivu 154.)

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 numeeristen tai merkkijonojen luokkakoodien lista vastaavalle *X*- ja *Y* -datalle.

QuartReg Katalogi >

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 223.

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	Määrityskerroin
stat.Resid	Regressioyhtälön jäännökset
stat.XReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg vastaava frekvenssilista*

R

R▶Pθ() Luettelo >

R▶Pθ (*xArvo***, *yArvo***)** ⇒ *arvo*
R▶Pθ (*xLista***, *yLista***)** ⇒ *lista*
R▶Pθ (*xMatriisi***, *yMatriisi***)** ⇒ *matriisi*

Laskee yhtäpitävän θ-koordinaatin argumenttiparille (*x*,*y*).

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

Huomaa: Voit syöttää tämän funktion tietokoneen näppäimistöltä kirjoittamalla **`R@>Ptheta(...)`**.

Astekulmatilassa:

$R\blacktriangleright P\theta(2,2)$ 45.

Graadikulmatilassa:

$R\blacktriangleright P\theta(2,2)$ 50.

Radiaanikulmatilassa:

$R\blacktriangleright P\theta(3,2)$ 0.588003

$R\blacktriangleright P\theta\left(\left[3\ \ -4\ \ 2\right],\left[0\ \ \frac{\pi}{4}\ \ 1.5\right]\right)$

$\left[0.\ \ 2.94771\ \ 0.643501\right]$

R▶Pr() Luettelo >

R▶Pr (*xArvo***, *yArvo*) ⇒ *arvo***
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:

R▶Pr(3,2)	3.60555
R▶Pr([3 -4 2],[0 π/4 1.5])	[3 4.07638 5/2]

▶Rad Luettelo >

Arvo1▶*Rad* ⇒ *arvo*

Muuntaa argumentin radiaanikulmayksikköön.

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

Astekulmatilassa:

(1.5)▶Rad	$(0.02618)^r$

Graadikulmatilassa:

(1.5)▶Rad	$(0.023562)^r$

rand() Luettelo >

rand() ⇒ *lauseke*
rand(*Kokeiden lkm*) ⇒ *lista*

rand() laskee satunnaisen arvon välillä 0 ja 1.

rand(*Kokeiden lkm*) laskee listan, joka sisältää *Kokeiden lkm* satunnaista arvoa välillä 0 ja 1.

Aseta satunnainen siemenluku.

RandSeed 1147	*Done*
rand(2)	{0.158206,0.717917}

randBin() Luettelo >

randBin(*n*, *p*) ⇒ *lauseke*
randBin(*n*, *p*, *Kokeiden lkm*) ⇒ *lista*

randBin(*n*, *p*) laskee satunnaisen reaaliluvun määrätystä binomijakaumasta.

randBin(*n*, *p*, *Kokeiden lkm*) laskee listan, joka sisältää *Kokeiden lkm* satunnaista reaalilukua määrätystä binomijakaumasta.

randBin(80,0.5)	46.
randBin(80,0.5,3)	{43.,39.,41.}

randInt() Luettelo >

randInt(*Muutt.alaraja* **,** *Muutt.yläraja* **) ⇒** *lauseke*

randInt(*Muutt.alaraja* **,** *Muutt.yläraja* **,** *Kokeiden lkm* **) ⇒** *lista*

randInt(3,10)	3.
randInt(3,10,4)	{9.,3.,4.,7.}

randInt(*Muutt.alaraja* **,** *Muutt.yläraja* **)** laskee satunnaisen kokonaisluvun rajojen *Muutt.alaraja* ja *Muutt.yläraja* määräämällä kokonaislukuvälillä.

randInt(*Muutt.alaraja* **,** *Muutt.yläraja* **,** *Kokeiden lkm* **)** laskee listan, joka sisältää *Kokeiden lkm* satunnaista kokonaislukua määrätyllä välillä.

randMat() Luettelo >

randMat(*numRivit* **,** *numSarakkeet* **) ⇒** *matriisi*

Laskee välillä -9 ja 9 määrätylle mitalle kokonaislukumatriisin.

Kummankin argumentin on sievennyttyvä kokonaisluvuksi.

RandSeed 1147	*Done*
randMat(3,3)	$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$

Huomaa: Tämän matriisin arvot muuttuvat aina, kun painat enter.

randNorm() Luettelo >

randNorm(μ, σ**)** ⇒ *lauseke*
randNorm(μ, σ, *Kokeiden lkm***)** ⇒ *lista*

randNorm(μ, σ**)** laskee desimaaliluvun määrätystä normaalijakaumasta. Luku voi olla mikä tahansa reaaliluku, mutta se keskittyy voimakkaasti välille [μ−3•σ, μ+3•σ].

randNorm(μ, σ, *Kokeiden lkm***)** listan (list), joka sisältää *Kokeiden lkm* desimaalilukua määrätystä normaalijakaumasta.

RandSeed 1147	*Done*
randNorm(0,1)	0.492541
randNorm(3,4.5)	−3.54356

randPoly() Luettelo >

randPoly(*Muutt***,** *Aste***)** ⇒ *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*
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***])** ⇒ *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 *list3*={1,2,3,4,5}	*Done*
Define *list4*=randSamp(*list3*,6)	*Done*
list4	{1.,3.,3.,1.,3.,1.}

RandSeed Luettelo >

RandSeed *Luku*

Jos *Luku* = 0, tämä asettaa siemenluvut satunnaislukugeneraattorin tehdasasetuksiin. Jos *Luku* ≠ 0, sitä käytetään luomaan kaksi siemenlukua, jotka tallennetaan järjestelmän muuttujiin siemen1 ja siemen2

RandSeed 1147	*Done*
rand()	0.158206

real() Luettelo >

real(*Arvo1***)** ⇒ *arvo*

Laskee argumentin reaalilukuosan.

$\text{real}(2+3\cdot i)$ → 2

real(*List1***)** ⇒ *lista*

Laskee kaikkien alkutekijöiden reaaliosat

$\text{real}(\{1+3\cdot i,3,i\})$ → $\{1,3,0\}$

real(*Matriisi1***)** ⇒ *matriisi*

Laskee kaikkien alkutekijöiden reaaliosat.

$\text{real}\left(\begin{bmatrix}1+3\cdot i & 3\\ 2 & i\end{bmatrix}\right)$ → $\begin{bmatrix}1 & 3\\ 2 & 0\end{bmatrix}$

▶Rect Luettelo >

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.

$\left(\begin{bmatrix}3 & \angle\frac{\pi}{4} & \angle\frac{\pi}{6}\end{bmatrix}\right)\blacktriangleright\text{Rect}$

$\begin{bmatrix}1.06066 & 1.06066 & 2.59808\end{bmatrix}$

Huomaa: ▶**Rect** on näyttömuodon ohje, ei muunnosfunktio. Voit käyttää komentoa ainoastaan syöterivin lopussa, eikä se päivitä arvoa *ans*.

Huomaa: Katso myös ▶**Polar**, sivu 117.

kompleksiArvo **▶Rect**

Näyttää kompleksisen arvon *complexValue*suorakulmaisessa muodossa a+bi. Arvolla *complexValue* voi olla mikä tahansa kompleksinen muoto. Syöte $re^{i\theta}$aiheuttaa kuitenkin virheen astekulmatilassa.

Huomaa: Polaarisessa syötteessä (r∠ θ) on käytettävä sulkeita.

Radiaanikulmatilassa:

$\left(4\cdot e^{\frac{\pi}{3}}\right)\blacktriangleright\text{Rect}$ → 11.3986

$\left(\left(4\angle\frac{\pi}{3}\right)\right)\blacktriangleright\text{Rect}$ → $2.+3.4641\cdot i$

Graadikulmatilassa:

$((1\angle 100))\blacktriangleright\text{Rect}$ → i

Astekulmatilassa:

$((4\angle 60))\blacktriangleright\text{Rect}$ → $2.+3.4641\cdot i$

Huomaa: Näppäilläksesi merkin ∠ valitse se Luettelon symboliluettelosta.

ref(*Matriisi1*[, *Tol*]**)** ⇒ *matriisi*

Laskee matriisin *Matriisi1* 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 •max(dim(*Matriisi1*)) •rowNorm (*Matriisi1*)

Vältä määrittelemättömiä alkutekijöitä matriisissa *Matriisi1*. Tuloksena saattaa olla odottamattomia vastauksia.

Jos esimerkiksi a on määrittelemätön seuraavassa lausekkeessa, näkyviin tulee varoitusviesti, ja vastaus näyttää seuraavalta:

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \qquad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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.

$$\text{ref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) \qquad \begin{bmatrix} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

ref() Luettelo >

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \mid a=0 \qquad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Huomaa: Katso myös **rref()**, page 139.

RefreshProbeVars Luettelo >

RefreshProbeVars

Pääset mittausarvoon kaikista TI-Basic-ohjelmaan kytketyistä antureista.

StatusVar Value	Tila
statusVar =0	Normaali (jatka ohjelmalla)
statusVar =1	Vernier DataQuest™ on tiedonkeruutilassa. **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 määritys()=
Prgm
© Tarkista, onko järjestelmä valmiina
RefreshProbeVars-tila
Jos tila=0, niin
Käyttötila "valmis"
Muuttujalle n,1,50
RefreshProbeVars-tila
lämpotila:=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
```

RefreshProbeVars

Luettelo >

Huomaa: Tätä voi käyttää myös TI-Innovator™i-laitteen keskiössä.

remain()

Luettelo >

remain(*Arvo1***,** *Arvo2***)** ⇒ *arvo*
remain(*Lista1***,** *Lista2***)** ⇒ *lista*
remain(*Matriisi1***,** *Matriisi2***)** ⇒ *matriisi*

Laskee ensimmäisen argumentin jäännöksen toisen argumentin suhteen seuraavien identtisten yhtälöiden määritelmien mukaisesti:

remain(x,0) x
remain(x,y) x−y•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 99.

remain(7,0)	7
remain(7,3)	1
remain(⁻7,3)	⁻1
remain(7,⁻3)	1
remain(⁻7,⁻3)	⁻1
remain({12,⁻14,16},{9,7,⁻5})	{3,0,1}

$$\text{remain}\left(\begin{bmatrix} 9 & -7 \\ 6 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 3 \\ 4 & -3 \end{bmatrix}\right) \qquad \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$$

Pyydä

Luettelo >

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*.

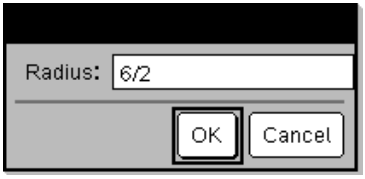
Jos käyttäjä klikkaa **Cancel**, ohjelma etenee hyväksymättä mitään syötteitä. Ohjelma käyttää muuttujan *muutt* aikaisempaa arvoa, jos *var* on jo määritelty.

Määrittele ohjelma:

```
Määrittele pyydä_demo()=Ohjelma
    Pyydä "Säde: ",r
    Disp "Ala = ",pi*r²
EndPrgm
```

Aja ohjelma ja kirjoita vastaus:

```
Pyydä_demo()
```



Vastaus **OK**-näppäimen painamisen jälkeen:

Valinnainen *NäytäLippu* -argumentti voi olla mikä tahansa lauseke.

- 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 muuttuja*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 [on]-painiketta painettuna ja paina toistuvasti [enter]-painiketta.
- **Windows®:** Pidä **F12**-näppäintä pohjassa ja paina toistuvasti **Enter**.
- **Macintosh®:** Pidä **F5**-näppäintä pohjassa

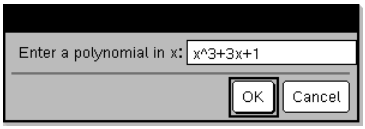
```
Säde: 6/2
Ala = 28.2743
```

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}
```

ja paina toistuvasti **Enter**.

- **iPad®:** Sovellus näyttää kehotuksen. Voit jatkaa odottamista tai peruuttaa.

Huomaa: Katso myös **RequestStr**, page 133.

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 [on]-painiketta painettuna ja paina toistuvasti [enter]-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 131.

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.
```

Laske — Luettelo >

Return [*Laus*]

Laskee lausekkeen *Laus*funktion tuloksena Käytetään lohkon **Func**...**EndFunc** sisällä.

Huomaa: Käytä **Return** ilman argumenttia lohkossa **Prgm**...**EndPrgm** poistuaksesi 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() — Luettelo >

oikea(*List1***[, *Num*]) ⇒ *lista***

Laskee oikeanpuoleisimmat *Num*-alkutekijät, jotka sisältyvät listaan *List1*.

Jos jätät pois komennon *Num*, laskee kaiken listasta *List1*.

```
right({1,3,-2,4},3)              {3,-2,4}
```

right(*lädemerkkijono***[, *Num*]) ⇒ *merkkijono***

Laskee oikeanpuoleisimmat *Num*-merkit, jotka sisältyvät merkkijonoon *lähdemerkkijono*

Jos jätät pois komennon*Num*, laskee kaiken *lähdemerkkijonosta*.

```
right("Hello",2)                 "lo"
```

right(*Vertailu***) ⇒ *lauseke***

Laskee yhtälön tai epäyhtälön oikean puolen.

rk23 () — Luettelo >

rk23(*Laus***, *Var*, *depVar*, {*Var0*, *VarMax*}, *depVar0*, *VarStep* [, *diftol*]) ⇒ *matriisi***

rk23(*SystemOfExpr***, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep*[, *diftol*]) ⇒ *matriisi*c**

rk23(*ListOfExpr***, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep*[, *diftol*]) ⇒ *matriisi***

Differentiaaliyhtälö:

y'=0.001*y*(100-y) ja y(0)=10

```
rk23(0.001·y·(100−y),t,y,{0,100},10,1)
[ 0.    1.        2.        3.      4
  10.   10.9367   11.9493   13.042  14.2
```

Käyttää Runge-Kutta-menetelmää järjestelmän ratkaisuun

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

funktiolla *depVar*(*Var0*)=*depVar0* välillä [*Var0*,*VarMax*]. Laskee tulokseksi matriisin, 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 *VarStep* on arvoltaan nollasta eroava numero: merkki(*VarStep*) = merkki (*VarMax*-*Var0*) ja ratkaisut lasketaan muuttujalla *Var0*+i**VarStep* kaikille i=0,1,2,... niin, että *Var0*+i**VarStep* on alueella [*muutt0*,*VarMax*] (muuttujalla *VarMax* ei ehkä ole ratkaisuarvoa).

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

Sama yhtälö, jossa *diftol* on asetettu 1.E−6

rk23(0.001·y·(100−y),t,y,{0,100},10,1,1.E-6)

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

Yhtälöryhmä:

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

kun *y1*(0)=2 and *y2*(0)=5

rk23({−y1+0.1·y1·y2, 3·y2−y1·y2},t,{y1,y2},{0,5},{2,5},1)

0.	1.	2.	3.	4.
2.	1.94103	4.78694	3.25253	1.82848
5.	16.8311	12.3133	3.51112	6.27245

rk23 ()

Luettelo >

jos *VarStep* sievenee nollaksi, ratkaisut lasketaan "Runge-Kutta" -muuttujan *Muutt* arvoilla.

diftol on virhetoleranssi (oletuksena 0.001).

root()

Luettelo >

root(*Arvo***)** ⇒ *juuri*
root(*Arvo1***, *Arvo2***)** ⇒ *juuri*

juuri(*Arvo*) laskee arvon *Arvo*neliöjuuren.

juuri(*Arvo1***, *Arvo2***)** laskee arvona *Arvo2* arvon*Arvo1* juuren. *Arvo1* voi olla reaalinen tai kompleksinen liukulukuvakio, kokonaisluku tai kompleksinen rationaalilukuvakio

Huomaa: Katso myös **N:s juuri -malli**, sivu 2.

$\sqrt[3]{8}$	2
$\sqrt[3]{3}$	1.44225

rotate()

Luettelo >

rotate(*Kokonaisluku1***[,***KiertojenLkm***])** ⇒ *kokonaisluku*

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 lisätietoa kohdasta ▶**Kantaluku2**, sivu 16.

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.

0b0000000000000011110101100001101 01

Oikeanpuoleisin bitti siirtyy vasemmaisimmaksi.

tuottaa:

Binaarisessa kantalukutilassa:

rotate(0b11111111111111111111111111111111)	
0b1000000000000000000000000000000011▸	
rotate(256,1)	0b1000000000

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

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 O-kirjain).

rotate() Luettelo >

0b10000000000000011110101100001101 0

Vastaus näkyy kantalukutilan mukaisesti.

rotate(*Lista1*[**,***KiertojenLkm*]**)** ⇒ *lista*

Laskee *Lista1:n*kopion, jota on kierretty oikealle tai vasemmalle *KiertojenLkm*:n elementtien määrittämän määrän. Ei muuta *Listaa1*.

Jos *KiertojenLkm* on positiivinen, kierto on vasemmalle. Jos *KiertojenLkm* on negatiivinen, kierto on oikealle. Oletusarvo on −1 (kierrä oikealle yksi elementti).

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(*Merkkijono1*[**,***KiertojenLkm*]**)** ⇒ *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).

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"bcda"

round() Luettelo >

round(*Arvo1*[**,** *numeroa*]**)** ⇒ *lauseke*

Laskee argumentin pyöristettynä jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

numeroiden on oltava kokonaisluku välillä 0–12. Jos *numeroita* ei sisällytetä, laskee argumentin pyöristettynä 12 merkitykselliseen numeroon.

Huomaa: Näytettävät numerot -tila saattaa vaikuttaa näyttöön.

round(1.234567,3)	1.235

round(*Lista1*[**,** *numerot*]**)** ⇒ *lista*

Pyöristää alkutekijöiden listan jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

round({π,√2,ln(2)},4)	{3.1416,1.4142,0.6931}

round() Luettelo >

round(*Matriisi1*[, *numerot*]) ⇒ *matriisi*

Pyöristää alkutekijöiden matriisin jättäen desimaalipisteen jälkeen määrätyn määrän numeroita.

$$\text{round}\left(\begin{bmatrix}\ln(5) & \ln(3)\\ \pi & e^{1}\end{bmatrix},1\right) \qquad \begin{bmatrix}1.6 & 1.1\\ 3.1 & 2.7\end{bmatrix}$$

rowAdd() Luettelo >

rowAdd(*Matriisi1*, *rIndeksi1*, *rIndeksi2*) ⇒ *matriisi*

Laskee matriisin *Matriisi1* kopion rivi*rIndeksi2* korvattuna rivien summalla*rIndeksi1* ja *rIndeksi2*.

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

rowDim() Luettelo >

rowDim(*Matriisi***)** ⇒ *lauseke*

Laskee *Matriisin* sisältämien rivien lukumäärän.

Huomaa: Katso myös **colDim()**, sivu 25.

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

$$\text{rowDim}(m1) \qquad 3$$

rowNorm() Luettelo >

rowNorm(*Matriisi***)** ⇒ *lauseke*

Laskee *Matriisin* riveillä olevien elementtien itseisarvojen summien maksimin.

Huomaa: Kaikkien matriisien alkutekijöiden on sievennyttävä luvuiksi. Katso myös **colNorm()**, sivu 25.

$$\text{rowNorm}\left(\begin{bmatrix}-5 & 6 & -7\\ 3 & 4 & 9\\ 9 & -9 & -7\end{bmatrix}\right) \qquad 25$$

rowSwap() Luettelo >

rowSwap(*Matriisi1*, *rIndeksi1*, *rIndeksi2*) ⇒ *matriisi*

Laskee matriisin *Matriisi1*, jossa rivien *rIndeksi1* ja *rIndeksi2* paikkoja on vaihdettu.

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

$$\text{rowSwap}(mat,1,3) \qquad \begin{bmatrix}5 & 6\\ 3 & 4\\ 1 & 2\end{bmatrix}$$

rref() Luettelo >

rref(*Matriisi1*[, *Tol*]**)** ⇒ *matriisi*

Laskee matriisin *Matriisi1* pelkistetyn rivi-echelon-muodon.

$$\text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) \qquad \begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

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 määritelty arvoa. Muussa tapauksessa komentoa *Tol* ei huomioida.

- Jos käytät painikkeita ctrl enter 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 •max(dim(*Matriisi1*)) •rowNorm (*Matriisi1*)

Huomaa: Katso myös **ref()**, page 129.

S

sec() trig -painike

sec(*Arvo1***)** ⇒ *arvo*

sec(*Lista1***)** ⇒ *lista*

Laskee *Arvo1*:n sekantin tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien sekantit.

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

Astekulmatilassa:

sec(45)	1.41421
sec({1,2.3,4})	{1.00015,1.00081,1.00244}

sec⁻¹() trig -painike

sec⁻¹(*Arvo1*) ⇒ *arvo*

sec⁻¹(*Lista1*) ⇒ *lista*

Määrittää kulman, jonka sekantti on *Arvo1*, tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien käänteissekantit.

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

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla `arcsec(...)`.

Astekulmatilassa:

$\sec^{-1}(1)$ 0.

Graadikulmatilassa:

$\sec^{-1}(\sqrt{2})$ 50.

Radiaanikulmatilassa:

$\sec^{-1}(\{1,2,5\})$ $\{0,1.0472,1.36944\}$

sech() Katalogi >

sech(*Arvo1*) ⇒ *arvo*

sech(*Lista1*) ⇒ *lista*

Laskee *Arvo1*:n hyperbolisen sekantin tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien hyperboliset sekantit.

$\mathrm{sech}(3)$ 0.099328

$\mathrm{sech}(\{1,2.3,4\})$ $\{0.648054,0.198522,0.036619\}$

sech⁻¹() Katalogi >

sech⁻¹(*Arvo1*) ⇒ *arvo*

sech⁻¹ (*Lista1*) ⇒ *lista*

Laskee *Arvo1*:n käänteisen hyperbolisen sekantin tai määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien käänteiset hyperboliset sekantit.

Radiaanikulma- ja suorakulmakompleksitilassa:

$\mathrm{sech}^{-1}(1)$ 0

$\mathrm{sech}^{-1}(\{1,-2,2.1\})$ $\{0,2.0944\cdot i,8.\mathrm{E}^{-}15+1.07448\cdot i\}$

sech⁻¹() Katalogi >

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

Send Laite-valikko

Send *lausTaiMerkkijono1* [, *lausTaiMerkkijono2*] ...

Ohjelmointikomento: Lähettää yhden tai useamman TI-Innovator™ Hub komennon liitettyyn laitteeseen.

lausTaiMerkkijonon täytyy olla validi TI-Innovator™ Hub Komento. Tavallisesti *lausTaiMerkkijono* sisältää **"SET ..."**komento laitteen tai **"READ ..."** hallitsemiseksi komento tiedon kysymiseksi.

Argumentit lähetetään laitteeseen peräkkäisinä.

Huomio: Komentoa **Send** voi käyttää käyttäjän määrittelemän ohjelman sisällä mutta ei funktion sisällä.

Huomio: Katso myös **Get** (sivu 62), **GetStr** (sivu 69), ja **eval()** (sivu 50).

Esimerkki: Sytytä sisäänrakennetun RGB-ledin sininen väri 0,5 sekunniksi.

Send "SET COLOR.BLUE ON TIME .5"	*Done*

Esimerkki: Kysy laitteen sisäänrakennetun valotasoanturin tämänhetkinen arvo. **Get**-komento noutaa arvon ja sijoittaa sen muuttujaan *lightval*.

Send "READ BRIGHTNESS"	*Done*
Get *lightval*	*Done*
lightval	0.347922

Esimerkki: Lähetä laskettu taajuus laitteen sisäänrakennettuun kaiuttimeen. Käytä erikoismuuttujaa *iostr.SendAns* näyttääksesi laitekomennon ilmaus arvioituna.

$n:=50$	50
$m:=4$	4
Send "SET SOUND eval(m· n)"	*Done*
iostr.SendAns	"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.

$\text{seq}(n^2,n,1,6)$	$\{1,4,9,16,25,36\}$
$\text{seq}\left(\frac{1}{n},n,1,10,2\right)$	$\left\{1,\frac{1}{3},\frac{1}{5},\frac{1}{7},\frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2},n,1,10,1\right)\right)$	$\frac{1968329}{1270080}$

Huom: Vastauksen pakottaminen likimääräiseksi:

seq() (sekv) Katalogi >

Kämmenlaite: Paina ctrl enter.
Windows®: Paina **Ctrl+Enter**.
Macintosh®: Paina ⌘+**Enter**.
iPad®: Pidä **enter** ja valitse ≈.

$$\text{sum}\left(\text{seq}\left(\frac{1}{n^2},n,1,10,1\right)\right) \qquad 1.54977$$

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

Luo sekvenssin 5 ensimmäistä termiä $u(n)$ = $u(n\text{-}1)^2/2$, kun $u(1)$=**2** ja *VarStep*=**1**.

$$\text{seqGen}\left(\frac{(u(n-1))^2}{n},n,u,\{1,5\},\{2\}\right)$$
$$\left\{2,2,\frac{4}{3},\frac{4}{9},\frac{16}{405}\right\}$$

Esimerkki, jossa Var0=2:

$$\text{seqGen}\left(\frac{u(n-1)+1}{n},n,u,\{2,5\},\{3\}\right)$$
$$\left\{3,\frac{4}{3},\frac{7}{12},\frac{19}{60}\right\}$$

seqGen(*ListOrSystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*} [, *MatrixOfInitTerms* [, *VarStep* [, *CeilingValue*]]]**)** ⇒*matrix*

Luo termimatriisin sekvenssien *ListOfDepVars*(*Var*) =*ListOrSystemOfExpr* ryhmälle (tai listalle) seuraavasti: Lisää riippumattoman muuttujan *Var* arvoa arvosta *Var0* arvoon *VarMax* portailla *VarStep*, laskee *ListOfDepVars*(*Var*) muuttujan *Var* vastaaville arvoille käyttäen *Expr*-kaavaa ja *MatrixOfInitTerms* -luetteloa, ja antaa tulokset matriisina.

Kahden sekvenssin ryhmä:

$$\text{seqGen}\left(\left\{\frac{1}{n},\frac{u2(n-1)}{2}+u1(n-1)\right\},n,\{u1,u2\},\{1,5\},\left[\begin{matrix}_\\2\end{matrix}\right]\right)$$
$$\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5}\\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24}\end{bmatrix}$$

Huomaa: Muuttujaa Tyhjä (_) yllä olevassa aloitustermimatriisissa käytetään ilmoittamaan, että u1(n):n aloitustermi on laskettu käyttämällä täsmällistä sekvenssikaavaa u1(n)=1/n.

Muuttujan *Var* alkuperäinen sisältö on edelleen tallessa funktion **seqGen()** suorittamisen jälkeen

Oletusarvo *VarStep* = **1**.

seqn() Katalogi >

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 termiluettelon ei-rekursiiviselle sekvenssille *u*(*n*)=*Expr*(*u*, n) seuraavasti: Lisää muuttujan *n* arvoa arvosta 1 arvoon *nMax* 1 välein, laskee lausekkeen *u*(*n*) muuttujan *n* vastaaville arvoille käyttäen *Expr*(*n*) -kaavaa ja antaa tulokset luettelona.

Jos arvo *nMax* puuttuu, *nMax* asetetaan arvoon 2500.

Jos *nMax*=0, *nMax* asetetaan arvoon 2500.

Huomaa: seqn() hakee funktion **seqGen( )** kun *n0*=**1** ja *nstep* =**1**

Luo sekvenssin 6 ensimmäistä termiä *u*(*n*) = *u*(*n*-1)/2, kun *u*(1)=**2**.

$$\text{seqn}\left(\frac{u(n-1)}{n},\{2\},6\right) \qquad \left\{2,1,\frac{1}{3},\frac{1}{12},\frac{1}{60},\frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2},6\right) \qquad \left\{1,\frac{1}{4},\frac{1}{9},\frac{1}{16},\frac{1}{25},\frac{1}{36}\right\}$$

setMode() Katalogi >

setMode(*tilaNimiKokonaisluku***, *asetusKokonaisluku*) ⇒*kokonaisluku***
setMode(*lista***) ⇒*kokonaislukulista***

Toimii vain funktiossa tai ohjelmassa.

setMode(*tilaNimiKokonaisluku***, *asetusKokonaislkuku*)** asettaa tilan *tilaNimiKokonaisluku* väliaikaisesti uuteen asetukseen *asetusKokonaisluku* ja määrittää kokonaisluvun, joka vastaa kyseisen tilan alkuperäistä asetusta. Muutos on rajoitettu ohjelman/funktion suorittamisen ajalle.

Näytä π:n likiarvo käyttäen Näytettävät numerot -tilan oletusasetusta ja näytä sen jälkeen π asetuksella Kiinteä2. Tarkista, että oletusarvo palautuu ohjelman suorittamisen jälkeen.

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)** → *muutt*, voit palauttaa nämä asetukset komennolla **setMode(***muutt***)** aina funktion tai ohjelman sulkemiseen saakka. Katso **getMode()**, sivu 68.

Huomaa: Nykyiset tila-asetukset siirtyvät haettuihin alarutiineihin. Jos jokin alarutiini muuttaa tila-asetusta, tilamuutos häviää, kun ohjaus palautuu hakurutiiniin.

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

```
Define prog1()=Prgm                    Done
               Disp π
               setMode(1,16)
               Disp π
               EndPrgm
prog1()
                                  3.14159
                                  3.14
                                     Done
```

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

Tilan nimi	Tilaa vastaava kokonaisluku	Asetuksia vastaavat kokonaisluvut
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
Vektorimuoto	**6**	**1**=Suorakulma, **2**=Sylinteri, **3**=Pallo
Kantaluku	**7**	**1**=Desimaali, **2**=Heksagonaalinen, **3**=Binaarinen

shift() Katalogi >

shift(*Kokonaisluku1*[**,***SiirtojenLkm*]**)** ⇒*kokonaisluku*

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 16.

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.

0b0000000000000111101011000011010

Lisää 0:n, jos vasemmanpuoleisin bitti on 0, tai 1:n, jos vasemmanpuoleisin bitti on 1.

Binaarisessa kantalukutilassa:

shift(0b1111010110000110101)	0b1111010110000011010
shift(256,1)	0b1000000000

Heksadesimaalisessa kantalukutilassa:

shift(0h78E)	0h3C7
shift(0h78E,⁻2)	0h1E3
shift(0h78E,2)	0h1E38

Tärkeää: Binaariluvun edelle tulee aina merkitä etumerkki 0b ja heksadesimaaliluvun edelle 0h (nolla, ei O-kirjain).

shift() Katalogi >

tuottaa:

0b00000000000000011110101100001101 0

Vastaus näkyy kantalukutilan mukaisesti. Alkunollia ei näytetä.

shift(*Lista1* [**,***SiirtojenLkm*]**)⇒***lista*

Luo *Lista1*:n kopion, jota on siirretty oikealle tai vasemmalle *SiirtojenLkm*:n elementtien määrittämän määrän. Ei muuta *Lista1*:ä.

Jos *SiirtojenLkm* on positiivinen, siirto tapahtuu vasemmalle. Jos *SiirtojenLkm* on negatiivinen, siirto tapahtuu oikealle. Oletusarvo on ⁻1 (siirrä yksi elementti oikealle).

Elementit, jotka siirto vie *lista*n alkuun tai loppuun, asettuvat symboliksi “undef”.

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(*Merkkijono1* [**,***SiirtojenLkm*]**)**
⇒*merkkijono*

Luo *Merkkijono1*:n kopion, jota on siirretty oikealle tai vasemmalle *SiirtojenLkm*:n merkkien määrittämän määrän. Ei muuta *Merkkijono1*:ä.

Jos *SiirtojenLkm* on positiivinen, siirto tapahtuu vasemmalle. Jos *SiirtojenLkm* on negatiivinen, siirto tapahtuu oikealle. Oletusarvo on ⁻1 (siirrä yksi merkki oikealle).

Merkit, jotka siirto vie *merkkijono*n alkuun tai loppuun, muuttuvat välilyönneiksi.

shift("abcd")	" abc"
shift("abcd",⁻2)	" ab"
shift("abcd",1)	"bcd "

sign() (etumerkki) Katalogi >

sign(*Arvo1***)⇒***arvo*

sign(*Lista1***)⇒***lista*

sign(*Matriisi1***)⇒***matriisi*

sign(⁻3.2)	⁻1
sign({2,3,4,⁻5})	{1,1,1,⁻1}

Jos kompleksilukutila on Reaali:

sign([⁻3 0 3])	[⁻1 undef 1]

sign() (etumerkki) Katalogi >

Kun kyseessä on reaali- tai kompleksilukuarvo *Arvo1*, antaa vastauksena *Arvo1* / **abs(***Arvo1***)**, kun *Arvo1* ≠ 0.

Vastaus on 1, jos *Arvo1* on positiivinen.Vastaus on ⁻1, jos *Arvo1* on negatiivinen. **sign(0)** antaa vastauksen ±1, jos kompleksilukutilan asetus on Reaali; muussa tapauksessa antaa vastauksena itsensä.

sign(0) edustaa kompleksialueen yksikköpiiriä.

Kun kyseessä on lista tai matriisi, määrittää kaikkien elementtien etumerkit.

simult() Katalogi >

simult(*kerroinMatriisi***,** *vakioVektori***[,** *tol***])⇒***matriisi*

Määrittää sarakevektorin, joka sisältää lineaarisen yhtälöryhmän ratkaisut.

Huomaa: Katso myös **linSolve()**, sivu 86.

*kerroinMatriisi*n on oltava neliömatriisi, joka sisältää yhtälöiden kertoimet.

*vakioVektori*ssa on oltava sama rivimäärä (sama koko) kuin *kerroinMatriisi*ssa 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

Ratkaise yhtälöstä x ja y:
x + 2y = 1
3x + 4y = ⁻1

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

Ratkaisu on x=⁻3 and y=2.

Ratkaise:
ax + by = 1
cx + dy = 2

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\rightarrow matx1 \qquad \begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$$

$$\text{simult}\left(matx1,\begin{bmatrix}1\\2\end{bmatrix}\right) \qquad \begin{bmatrix}0\\\frac{1}{2}\end{bmatrix}$$

suoritetaan liukulukuaritmetiikalla.

- Jos *Tol* jätetään pois tai sitä ei käytetä, oletusarvoinen toleranssi lasketaan seuraavasti:
  5E⁻14 ·max(dim(*kerroinMatriisi*))
  ·rowNorm(*kerroinMatriisi*)

simult(*kerroinMatriisi*, *vakioMatriisi*[, *tol*]**)⇒***matriisi*

Ratkaisee lineaarisia yhtälöryhmiä, joissa jokaisessa ryhmässä on samat yhtälöiden kertoimet mutta eri vakiot.

Jokaisen *vakioMatriisi*n sarakkeen tulee sisältää jonkin yhtälöryhmän vakiot. Jokainen tulosmatriisin sarake sisältää vastaavan ryhmän ratkaisun.

Ratkaise:
$x + 2y = 1$
$3x + 4y = -1$

$x + 2y = 2$
$3x + 4y = -3$

$$\text{simult}\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix},\begin{bmatrix}1 & 2\\-1 & -3\end{bmatrix}\right) \qquad \begin{bmatrix}-3 & -7\\2 & \frac{9}{2}\end{bmatrix}$$

Ensimmäisessä ryhmässä x=⁻3 ja y=2.
Toisessa ryhmässä x=⁻7 ja y=9/2.

sin() trig -painike

sin(*Arvo1***)⇒***arvo*

sin(*Lista1***)⇒***lista*

sin(*Arvo1***)** määrittää argumentin sinin.

sin(*Lista1***)** määrittää listan, joka sisältää *Lista1*:n kaikkien elementin sinit.

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

Astekulmatilassa:

$\sin\left(\left(\frac{\pi}{4}\right)^r\right)$	0.707107
$\sin(45)$	0.707107
$\sin(\{0,60,90\})$	$\{0.,0.866025,1.\}$

Graadikulmatilassa:

$\sin(50)$	0.707107

Radiaanikulmatilassa:

$\sin\left(\frac{\pi}{4}\right)$	0.707107
$\sin(45°)$	0.707107

sin(*neliömatriisi1***)⇒***neliömatriisi*

Radiaanikulmatilassa:

sin() trig -painike

Määrittää *neliömatriisi1*:n matriisin sinin. Tämä ei ole sama kuin kunkin elementin sinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

$$\sin\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$$

$$\begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

$\sin^{-1}()$ trig -painike

$\sin^{-1}$(*Arvo1*)⇒*arvo*

$\sin^{-1}$(*Lista1*)⇒*lista*

$\sin^{-1}$(*Arvo1*) määrittää kulman, jonka sini on *Arvo1*.

$\sin^{-1}$(*Lista1*) määrittää listan, joka sisältää *Lista1*:n kaikkien elementtien käänteissinit.

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

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

$\sin^{-1}$(*neliömatriisi1*)⇒*neliömatriisi*

Laskee *neliömatriisi1*:n matriisin käänteissinin. Tämä ei ole sama kuin kunkin elementin käänteissinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

neliömatriisi1: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() Katalogi >

sinh(*Numver1*)⇒*arvo*

sinh(*Lista1*)⇒*lista*

sinh (*Arvo1*) laskee argumentin hyperbolisen kosinin.

$\sinh(1.2)$ 1.50946

$\sinh(\{0,1.2,3.\})$ $\{0,1.50946,10.0179\}$

sinh() Katalogi >

sinh (*Lista1***)** määrittää listan, joka sisältää *Lista1*:n jokaisen elementin hyperbolisen sinin.

sinh(*neliömatriisi1***)⇒***neliömatriisi*

Laskee *neliömatriisi1*:n matriisin hyperbolisen sinin. Tämä ei ole sama kuin kunkin elementin hyperbolisen sinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Radiaanikulmatilassa:

$$\sinh\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}360.954 & 305.708 & 239.604\\ 352.912 & 233.495 & 193.564\\ 298.632 & 154.599 & 140.251\end{bmatrix}$$

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

$\sinh^{-1}$(*Arvo1***)⇒***arvo*

$\sinh^{-1}$(*Lista1***)⇒***lista*

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

$$\sinh^{-1}(\{0,2.1,3\}) \qquad \{0,1.48748,1.81845\}$$

$\sinh^{-1}$(*Arvo1***)** laskee argumentin käänteisen hyperbolisen sinin.

$\sinh^{-1}$(*Lista1***)** määrittää listan, joka sisältää *Lista1*:n jokaisen elementin käänteiset hyperboliset sinit.

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

$\sinh^{-1}$(*neliömatriisi1***)⇒***neliömatriisi*

Laskee *neliömatriisi1*:n matriisin käänteisen hyperbolisen sinin. Tämä ei ole sama kuin kunkin elementin käänteisen hyperbolisen sinin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Radiaanikulmatilassa:

$$\sinh^{-1}\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}0.041751 & 2.15557 & 1.1582\\ 1.46382 & 0.926568 & 0.112557\\ 2.75079 & -1.5283 & 0.57268\end{bmatrix}$$

SinReg Katalogi >

SinReg *X*, *Y* [, [*Iteraatiot*] ,[*Jakso*] [,

Luokka, *Sisällytä*]]

Laskee siniregression listoista X ja Y. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 suoritusaika on pitempi ja päin vastoin.

Jakso määrittää arvioidun jakson. Mikäli sitä ei käytetä, X:n arvojen välisen eron tulisi olla sama ja arvojen tulisi olla peräkkäisessä järjestyksessä. Jos määrität *Jakso*n, x:n arvojen väliset erot voivat olla erisuuria.

Luokka on numeeristen tai merkkijonojen 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 elementitsivulla sivu 223.

Tulosmuuttuja	**Kuvaus**
stat.RegEqn	Regressioyhtälö: a ·sin(bx+c)+d
stat.a, stat.b, stat.c, stat.d	Regressiokertoimet
stat.Resid	Regressioyhtälön jäännökset

Tulosmuuttuja	Kuvaus
stat.XReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.YReg	Muokatun *Y Lista*:n sisältämä datapisteiden lista, jota käytetään regressiossa komentojen *Frekv*, *Luokkalista* ja *Sisällytä luokat rajoitusten mukaisesti*
stat.FreqReg	Komentoja *stat.XReg* ja *stat.YReg vastaava frekvenssilista*

SortA

Katalogi >

SortA *Lista1*[, *Lista2*] [, *Lista3*] ...
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 tyhjistä elementeistä, katso sivu 223.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
SortA *list1*	*Done*
list1	$\{1,2,3,4\}$
$\{4,3,2,1\} \rightarrow list2$	$\{4,3,2,1\}$
SortA *list2*,*list1*	*Done*
list2	$\{1,2,3,4\}$
list1	$\{4,3,2,1\}$

SortD

Katalogi >

SortD *Lista1*[, *Lista2*] [, *List3*] ...
SortD *Vektori1*[,*Vektori2*] [,*Vektori3*] ...

Muuten samanlainen kuin **SortA** paitsi, että **SortD** lajittelee elementit laskevaan järjestykseen.

Ensimmäisen argumentin sisältämät tyhjät elementit siirtyvät alas. Lisätietoja tyhjistä elementeistä, katso sivu 223.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
SortD *list1*,*list2*	*Done*
list1	$\{4,3,2,1\}$
list2	$\{3,4,1,2\}$

▸Sphere Katalogi >

Vektori **▸Sphere**

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

Näyttää rivi- tai sarakevektorin pallonmuotoisena [ρ ∠θ ∠ϕ].

*Vektori*n on oltava kooltaan 3, ja se voi olla rivi- tai sarakevektori.

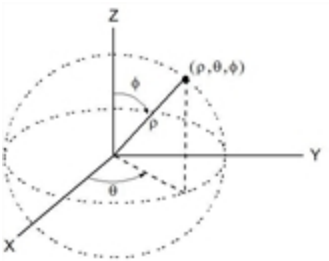
Huomaa: ▸Sphere on näyttömuodon ohje, ei muunnosfunktio. Voit käyttää komentoa ainoastaan syöterivin lopussa.

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina [ctrl] [enter].
Windows®: Paina **Ctrl+Enter**.
Macintosh®: Paina ⌘+**Enter**.
iPad®: Pidä **enter** ja valitse [≈].

$$[1 \quad 2 \quad 3]\blacktriangleright\text{Sphere}$$
$$[3.74166 \quad \angle 1.10715 \quad \angle 0.640522]$$

$$\left(\begin{bmatrix} 2 & \angle\frac{\pi}{4} & 3 \end{bmatrix}\right)\blacktriangleright\text{Sphere}$$
$$[3.60555 \quad \angle 0.785398 \quad \angle 0.588003]$$



sqrt() Katalogi >

sqrt(*Arvo1***)⇒***arvo*

sqrt(*Lista1***)⇒***lista*

Laskee argumentin neliöjuuren.

Kun kyseessä on lista, laskee kaikkien *Lista1*:n elementtien neliöjuuret.

Huomaa: Katso myös **Neliöjuurimalli**, sivu 1.

$\sqrt{4}$	2
$\sqrt{\{9,2,4\}}$	$\{3,1.41421,2\}$

stat.results

Näyttää tilastollisen laskutoimituksen tulokset.

Vastaukset näytetään nimiarvoparien sarjana. Näytetyt nimenomaiset nimet riippuvat viimeksi sievennetystä tilastofunktiosta tai komennosta.

Voit kopioida nimen tai arvon ja liittää sen muihin paikkoihin.

xlist:={1,2,3,4,5} {1,2,3,4,5}

ylist:={4,8,11,14,17} {4,8,11,14,17}

LinRegMx *xlist*,*ylist*,1: *stat.results*

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r²"	0.996109
"r"	0.998053
"Resid"	"{...}"

stat.values

"Linear Regression (mx+b)"
"m*x+b"
3.2
1.2
0.996109
0.998053
"{⁻0.4,0.4,0.2,0.,⁻0.2}"

Huomaa: Älä määritä muuttujia, joilla on sama nimi kuin tilastoanalyyseissä käytettävillä muuttujilla. Joissakin tapauksissa tästä voi olla seurauksena virhetilanne. Tilastoanalyyseissä 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.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.σx	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.σy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.σx1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.σx2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.$\bar{x}$
stat.b9	stat.FBlock	Stat.$\hat{p}$	stat.Σx^2	stat.$\bar{x}$1
stat.b10	stat.Fcol	stat.$\hat{p}$1	stat.Σxy	stat.$\bar{x}$2
stat.bList	stat.FInteract	stat.$\hat{p}$2	stat.Σy	stat.$\bar{x}$Diff
stat.χ^2	stat.FreqReg	stat.$\hat{p}$Diff	stat.Σy^2	stat.$\bar{x}$List
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg
stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat.$\bar{y}$
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat.$\hat{y}$
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat.$\hat{y}$List

stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

Huomaa: Aina kun Listat & Taulukot -sovellus laskee tilastolaskujen vastauksia, se kopioi “stat.” -ryhmän muuttujat “stat#.” -ryhmään, jossa # on luku, jota lisätään automaattisesti. Tällä tavoin voit säilyttää aikaisemmat tulokset suorittaessasi useita laskutoimituksia.

stat.values

Katalogi >

stat.values

Näyttää matriisin viimeksi sievennetylle tilastofunktiolle tai -komennolle lasketuista arvoista.

Toisin kuin **stat.results**, **stat.values** jättää pois arvoihin liittyvät nimet.

Voit kopioida arvon ja liittää sen muihin paikkoihin.

```
Katso esimerkki kohdassa
stat.results.
```

stDevPop()

Katalogi >

stDevPop(*Lista***[,** *frekvLista***])⇒***lauseke*

Laskee *Lista*:n sisältämien elementtien perusjoukon keskihajonnan.

Jokainen *frekvLista*n elementti näyttää *Lista*n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Lista*ssa tulee olla vähintään kaksi elementtiä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

Radiaanikulma- ja automaattisissa tiloissa:

$\text{stDevPop}(\{1,2,5,-6,3,-2\})$	3.59398
$\text{stDevPop}(\{1.3,2.5,-6.4\},\{3,2,5\})$	4.11107

stDevPop() Katalogi >

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

Laskee rivivektorin *Matriisi1*:n sarakkeiden perusjoukon keskihajonnoista.

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

Huomaa: *Matriisi1* :ssä on oltava vähintään kaksi riviä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

$$\text{stDevPop}\left(\begin{bmatrix}1 & 2 & 5\\ -3 & 0 & 1\\ 5 & 7 & 3\end{bmatrix}\right)$$
$$\begin{bmatrix}3.26599 & 2.94392 & 1.63299\end{bmatrix}$$

$$\text{stDevPop}\left(\begin{bmatrix}-1.2 & 5.3\\ 2.5 & 7.3\\ 6 & -4\end{bmatrix},\begin{bmatrix}4 & 2\\ 3 & 3\\ 1 & 7\end{bmatrix}\right)$$
$$\begin{bmatrix}2.52608 & 5.21506\end{bmatrix}$$

stDevSamp() Katalogi >

stDevSamp(*Lista***[,** *frekvLista***])**⇒*lauseke*

Laskee *Lista*n sisältämien elementtien otoksen keskihajonnan.

Jokainen *frekvLista*n elementti näyttää *Lista*n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Lista*ssa tulee olla vähintään kaksi elementtiä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

$\text{stDevSamp}(\{1,2,5,-6,3,-2\})$ 3.937

$\text{stDevSamp}(\{1.3,2.5,-6.4\},\{3,2,5\})$ 4.33345

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

Laskee rivivektorin *Matriisi1*:n sarakkeiden otosten keskihajonnoista.

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

Huomaa: *Matriisi1* :ssä on oltava vähintään kaksi riviä. Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

$$\text{stDevSamp}\left(\begin{bmatrix}1 & 2 & 5\\ -3 & 0 & 1\\ 5 & 7 & 3\end{bmatrix}\right)$$
$$\begin{bmatrix}4. & 3.60555 & 2.\end{bmatrix}$$

$$\text{stDevSamp}\left(\begin{bmatrix}-1.2 & 5.3\\ 2.5 & 7.3\\ 6 & -4\end{bmatrix},\begin{bmatrix}4 & 2\\ 3 & 3\\ 1 & 7\end{bmatrix}\right)$$
$$\begin{bmatrix}2.7005 & 5.44695\end{bmatrix}$$

Stop

Katalogi >

Stop

Ohjelmointikomento: Pysäyttää ohjelman.

Stop ei ole sallittu funktioissa.

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

i:=0	0
Define $prog1()$=Prgm For i,1,10,1 If i=5 Stop EndFor EndPrgm	*Done*
$prog1()$	*Done*
i	5

Store

Katso → (tallenna), sivu 205.

string() (merkkijono)

Katalogi >

string(*Laus***)⇒***merkkijono*

Sieventää lausekkeen *Laus* ja antaa vastauksen merkkijonona.

string(1.2345)	"1.2345"
string(1+2)	"3"

subMat()

Katalogi >

subMat(*Matriisi1***[, *alkurivi*] [, *alkusarake*] [, *loppurivi*] [, *loppusarake*]) ⇒***matriisi***

Laskee *Matriisi1*:n määritetyn alimatriisin.

Oletusarvot: *alkurivi*=1, *alkusarake*=1, *loppurivi*=viimeinen rivi, *endCol*=viimeinen sarake.

$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\rightarrow m1$	$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}$
subMat($m1$,2,1,3,2)	$\begin{bmatrix}4&5\\7&8\end{bmatrix}$
subMat($m1$,2,2)	$\begin{bmatrix}5&6\\8&9\end{bmatrix}$

Sum (Sigma)

Katso Σ(), sivu 197.

sum() Katalogi >

sum(*Lista***[, *Alku*[, *Loppu*]])⇒*lauseke***

Laskee *Lista*n elementtien summan.

Alku ja *Loppu* ovat valinnaisia. Ne määrittävät elementtien alueen.

Mikä tahansa tyhjä argumentti tuottaa tyhjän vastauksen. *Lista*ssa olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

sum({1,2,3,4,5})	15
sum({a,2·a,3·a})	"Error: Variable is not defined"
sum(seq(n,n,1,10))	55
sum({1,3,5,7,9},3)	21

sum(*Matriisi1***[, *Alku*[, *Loppu*]])⇒*matriisi***

Laskee rivivektorin, joka sisältää *Matriisi1*:n sarakkeiden elementtien summat.

Alku ja *Loppu* ovat valinnaisia. Ne määrittävät rivialueen.

Mikä tahansa tyhjä argumentti tuottaa tyhjän vastauksen. *Matriisi1*:ssä olevia tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

$\text{sum}\left(\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}\right)$	$\begin{bmatrix}5&7&9\end{bmatrix}$
$\text{sum}\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\right)$	$\begin{bmatrix}12&15&18\end{bmatrix}$
$\text{sum}\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},2,3\right)$	$\begin{bmatrix}11&13&15\end{bmatrix}$

sumIf() Katalogi >

sumIf(*Lista*,*Kriteerit*[, *SummaLista*])⇒*arvo***

Laskee kaikkien niiden *Lista*n sisältämien elementtien kumuloituneen summan, jotka vastaavat määritettyjä kriteereitä *Kriteerit*. Voit halutessasi antaa kumuloivat elementit määrittämällä vaihtoehtoisen listan, *summaLista*.

Lista voi olla lauseke, lista tai matriisi. *SummaLista*lla, mikäli se määritetään, on oltava samat mitat kuin *Lista*lla.

Kriteeri voi olla:

- Arvo, lauseke tai merkkijono. Esimerkiksi **34** kumuloi vain niitä *Lista*n elementtejä, jotka sieventyvät arvoon 34.
- Boolen lauseke, joka sisältää symbolin **?** kunkin elementin paikanpitäjänä. Esimerkiksi lauseke **?<10** kumuloi vain

sumIf({1,2,e,3,π,4,5,6},2.5<?<4.5)	12.859874482
sumIf({1,2,3,4},2<?<5,{10,20,30,40})	70

sumIf() Katalogi >

niitä *Lista*n elementtejä, jotka ovat alle 10.

Kun jokin *Lista*n elementti vastaa kriteereitä *Kriteerit*, elementti lisätään kumuloituvaan summaan. Jos sisällytät funktioon *summaLista*n, summaan lisätäänkin sen sijaan vastaava *summaLista*n elementti.

Listat & Taulukot -sovelluksessa voit käyttää solualueita *Lista*n ja *summaLista*n tilalla.

Tyhjiä elementtejä ei huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

Huomaa: Katso myös **countIf()**, sivu 32.

sumSeq() Katso Σ(), sivu 197.

system() Katalogi >

system(*Arvo1* [**,** *Arvo2* [**,** *Arvo3* [**,** ...]]]**)**

Laskee yhtälöryhmän listaksi muotoiltuna. Voit luoda yhtälöryhmän myös mallin avulla.

T

T (transponoi) Katalogi >

Matriisi1$^{\mathbf{T}}$⇒*matriisi*

Laskee *Matriisi1*:n transponoidun liittokompleksimatriisin.

$$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}^{T} \qquad \begin{bmatrix}1&4&7\\2&5&8\\3&6&9\end{bmatrix}$$

Huomaa: Voit syöttää tämän operaattorin tietokoneen näppäimistöltä kirjoittamalla `@t`.

tan() trig -painike

tan(*Arvo1***)⇒***arvo*
tan(*Lista1***)⇒***lista*

tan(*Arvo1***)** laskee argumentin tangentin.

tan(*Lista1***)** määrittää *Lista1*:n kaikkien elementtien tangenttien listan.

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

Astekulmatilassa:

$\tan\left(\left(\frac{\pi}{4}\right)^{r}\right)$ 1.

$\tan(45)$ 1.

$\tan(\{0,60,90\})$ $\{0.,1.73205,\text{undef}\}$

Graadikulmatilassa:

$\tan\left(\left(\frac{\pi}{4}\right)^{r}\right)$ 1.

$\tan(50)$ 1.

$\tan(\{0,50,100\})$ $\{0.,1.,\text{undef}\}$

Radiaanikulmatilassa:

$\tan\left(\frac{\pi}{4}\right)$ 1.

$\tan(45°)$ 1.

$\tan\left(\left\{\pi,\frac{\pi}{3},-\pi,\frac{\pi}{4}\right\}\right)$ $\{0.,1.73205,0.,1.\}$

tan(*neliömatriisi1***)⇒***neliömatriisi*

Laskee *neliömatriisi1*:n matriisin tangentin. Tämä ei ole sama kuin kunkin elementin tangentin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Radiaanikulmatilassa:

$$\tan\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}-28.2912 & 26.0887 & 11.1142\\ 12.1171 & -7.83536 & -5.48138\\ 36.8181 & -32.8063 & -10.4594\end{bmatrix}$$

$\tan^{-1}$() trig -painike

tan⁻¹(*Arvo1***)⇒***arvo*

tan⁻¹(*Lista1***)⇒***lista*

tan⁻¹(*Arvo1***)** laskee kulman, jonka tangentti on *Arvo1*.

Astekulmatilassa:

$\tan^{-1}(1)$ 45

Graadikulmatilassa:

$\tan^{-1}(1)$ 50

$\tan^{-1}()$ trig -painike

$\tan^{-1}$(*Lista1*) luo listan *Lista1*:n jokaisen elementin käänteistangenteista.

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

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

Radiaanikulmatilassa:

$\tan^{-1}(\{0,0.2,0.5\})$ $\{0,0.197396,0.463648\}$

$\tan^{-1}$(*neliömatriisi1*)⇒*neliömatriisi*

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.

Radiaanikulmatilassa:

$$\tan^{-1}\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}-0.083658 & 1.26629 & 0.62263\\ 0.748539 & 0.630015 & -0.070012\\ 1.68608 & -1.18244 & 0.455126\end{bmatrix}$$

tanh() Katalogi >

tanh(*Arvo1*)⇒*arvo*
tanh(*Lista1*)⇒*lista*

$\tanh(1.2)$ 0.833655

$\tanh(\{0,1\})$ $\{0.,0.761594\}$

tanh(*Arvo1*) laskee argumentin hyperbolisen tangentin.

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()**.

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

Radiaanikulmatilassa:

$$\tanh\left(\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}\right)$$

$$\begin{bmatrix}-0.097966 & 0.933436 & 0.425972\\ 0.488147 & 0.538881 & -0.129382\\ 1.28295 & -1.03425 & 0.428817\end{bmatrix}$$

tanh⁻¹() Katalogi >

tanh⁻¹(*Arvo1***)⇒***arvo*

tanh⁻¹(*Lista1***)⇒***lista*

tanh⁻¹(*Arvo1***)** laskee argumentin käänteisen hyperbolisen tangentin.

tanh⁻¹(*Lista1***)** luo listan *Lista1*:n jokaisen elementin käänteisistä hyperbolisista tangenteista.

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

Suorakulmakompleksimuodossa:

```
tanh⁻¹(0)                                   0.
tanh⁻¹({1,2.1,3})
  {undef,0.518046−1.5708·i,0.346574−1.570
```

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

tanh⁻¹(*neliömatriisi1***)⇒***neliömatriisi*

Laskee *neliömatriisi1*:n matriisin käänteisen hyperbolisen tangentin. Tämä ei ole sama kuin kunkin elementin käänteisen hyperbolisen tangentin laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

```
tanh⁻¹([1 5 3; 4 2 1; 6 −2 1])
[−0.099353+0.164058·i   0.267834−1.4908
 −0.087596−0.725533·i   0.479679−0.9473
  0.511463−2.08316·i   −0.878563+1.7901
```

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

tCdf() Katalogi >

tCdf(*alaraja***,***yläraja***,***df***)⇒***luku*, jos *alaraja* ja *yläraja* ovat lukuja, *lista*, jos *alaraja* ja *yläraja* ovat listoja

Laskee Studentin *t*-todennäköisyysjakauman *alaraja*n ja *yläraja*n välillä määritetylle vapausasteelle *df*.

Kun P(X ≤ *yläraja*), aseta *alaraja* = **−9E**999.

Text Katalogi >

Text*kehotemerkkijono*[**,** *Näytälippu*]

Määritä ohjelma, joka keskeytyy ja näyttää kunkin viidestä satunnaisluvusta valintaruudussa.

Text

Katalogi >

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 131, tai RequestStr, sivu 133.**

Huomaa: Tätä komentoa voi käyttää käyttäjän määrittämän ohjelman sisällä mutta ei funktion sisällä.

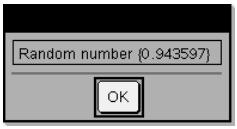
Paina mallin Prgm...EndPrgm jokaisen rivin lopussa näppäintä ↵ näppäimen enter sijaan. Tietokoneen näppäimistöllä **Alt**-näppäintä pidetään alhaalla ja painetaan **Enter**.

```
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:



Then

Katso If,sivu 72.

TInterval

Katalogi >

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 154.)

TInterval

Katalogi >

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

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.σx	Otoksen keskihajonta
stat.n	Otoksen keskiarvon sisältävän datasekvenssin pituus

TInterval_2Samp

Katalogi >

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 154.)

Poolaus=**1** poolaa varianssit; *Poolaus*=**0** ei poolaa variansseja.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Luottamusväli, joka sisältää jakauman luottamusvälin todennäköisyyden
stat.$\bar{x}$1-$\bar{x}$2	Satunnaisesta normaalijakaumasta olevien datasekvenssien otosten keskiarvot

Tulosmuuttuja	Kuvaus
stat.ME	Virhemarginaali
stat.df	Vapausasteet
stat.$\bar{x}$1, stat.$\bar{x}$2	Satunnaisesta normaalijakaumasta olevien datasekvenssien otosten keskiarvot
stat.σx1, stat.σx2	*Lista 1*:n ja *Lista 2*:n otosten keskihajonnat
stat.n1, stat.n2	Otosten lukumäärä datasekvensseissä
stat.sp	Poolattu keskihajonta. Laskettu, kun *Poolaus* = KYLLÄ

tPdf() Katalogi >

tPdf(*XArvo***,***df***)⇒***luku***,** jos *XArvo* on luku, *lista*, jos *XArvo* on lista

Laskee todennäköisyystiheysfunktion (pdf) Studentin *t*-jakaumalle määritetyllä *x*:n arvolla ja määritetyillä vapausasteilla *df*.

trace() Katalogi >

trace(*neliömatriisi***)⇒***arvo***

Laskee *neliömatriisi*n jäljityksen (päälävistäjän kaikkien elementtien summan).

$$\text{trace}\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\right) \qquad 15$$

$$a:=12 \qquad 12$$

$$\text{trace}\left(\begin{bmatrix}a&0\\1&a\end{bmatrix}\right) \qquad 24$$

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 232.

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.

$$eigenvals\left(\begin{bmatrix}-3\\-41\\5\end{bmatrix},\begin{bmatrix}-1 & 2 & -3.1\end{bmatrix}\right)$$

Huomaa: Katso myös **ClrErr**, sivu 24, ja **PassErr**, sivu 115.

```
Define prog1()=Prgm
    Try
    z:=z+1
    Disp "z incremented."
    Else
    Disp "Sorry, z undefined."
    EndTry
    EndPrgm
                                Done
z:=1:prog1()
                                z incremented.
                                Done
DelVar z:prog1()
                                Sorry, z undefined.
                                Done
```

```
Define eigenvals(a,b)=Prgm
© Program eigenvals(A,B) displays
eigenvalues of A·B
Try
  Disp "A= ",a
  Disp "B= ",b
  Disp " "
  Disp "Eigenvalues of A·B are:",eigVl(a*b)
Else
  If errCode=230 Then
    Disp "Error: Product of A·B must be a
square matrix"
    ClrErr
  Else
    PassErr
  EndIf
EndTry
```

Try (Kokeile) Katalogi >

EndPrgm

tTest Katalogi >

tTest $\mu 0$,*Lista*[,*Frekv*[,*Hypot*]]

(Datalistan syöte)

tTest $\mu 0$,$\bar{x}$,*sx*,*n*,[*Hypot*]

(Yhteenvetotilaston syöte)

Testaa hypoteesia yhden tuntemattoman perusjoukon keskiarvoon μ, kun perusjoukon keskihajontaa σ ei tunneta. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 223.

Tulosmuuttuja	Kuvaus
stat.t	($\bar{x}$ − $\mu 0$) / (stdev / sqrt(n))
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat.df	Vapausasteet
stat.$\bar{x}$	*Lista*n sisältämän datasekvenssin otoksen keskiarvo
stat.sx	Datasekvenssin otoksen keskihajonta
stat.n	Otoksen koko

tTest_2Samp Katalogi >

tTest_2Samp *Lista1*,*Lista2*[,*Frekv1*[,*Frekv2*[,*Hypot*[,*Poolaus*]]]]

tTest_2Samp Katalogi >

(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 154.)

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: $\mu1 > \mu2$, aseta *Hypot*>0

Poolaus=**1** poolaa varianssit
Poolaus=**0** ei poolaa variansseja

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

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	*Lista 1*:n ja *Lista 2*:n sisältämien datasekvenssien otosten keskiarvot
stat.sx1, stat.sx2	*Lista 1*:n ja *Lista 2*:n sisältämien datasekvenssien otosten keskihajonnat
stat.n1, stat.n2	Otosten koko
stat.sp	Poolattu keskihajonta. Laskettu, kun *Poolaus*=1

tvmFV() Katalogi >

tvmFV(*N,I,PV,Pmt*,[*PpY*],[*CpY*],[*PmtAt*]**)** ⇒*arvo*

```
tvmFV(120,5,0,-500,12,12)        77641.1
```

Talouslaskentafunktio, joka laskee rahan tulevan arvon.

tvmFV() Katalogi >

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 170. Katso myös **amortTbl ()**, sivu 7.

tvmI() Katalogi >

tvmI(*N***,***PV***,***Pmt***,***FV***,[***PpY***],[***CpY***],[***PmtAt***])** ⇒*arvo*

Talouslaskentafunktio, joka laskee vuosikoron.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 170. Katso myös **amortTbl ()**, sivu 7.

```
tvmI(240,100000,-1000,0,12,12)        10.5241
```

tvmN() Katalogi >

tvmN(*I***,***PV***,***Pmt***,***FV***,[***PpY***],[***CpY***],[***PmtAt***])** ⇒*arvo*

Talouslaskentafunktio, joka laskee maksuerien lukumäärän.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 170. Katso myös **amortTbl ()**, sivu 7.

```
tvmN(5,0,-500,77641,12,12)            120.
```

tvmPmt() Katalogi >

tvmPmt(*N***,***I***,***PV***,***FV***,[***PpY***],[***CpY***],[***PmtAt***])** ⇒*arvo*

Talouslaskentafunktio, joka laskee jokaisen maksuerän määrän.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 170. Katso myös **amortTbl ()**, sivu 7.

```
tvmPmt(60,4,30000,0,12,12)            -552.496
```

tvmPV() Katalogi >

tvmPV(*N***,***I***,***Pmt***,***FV***,[***PpY***],[***CpY***],[***PmtAt***])** ⇒*arvo*

tvmPV(48,4,⁻500,30000,12,12) ⁻3426.7

Talouslaskentafunktio, joka laskee nykyarvon.

Huomaa: TVM-funktioissa käytetyt argumentit on kuvattu TVM-argumenttien taulukossa, sivu 170. Katso myös **amortTbl ()**, sivu 7.

TVM-argumentti*	Kuvaus	Datatyyppi
N	Maksuerien lukumäärä	reaaliluku
I	Vuosikorko	reaaliluku
PV	Nykyarvo	reaaliluku
Pmt	Maksun määrä	reaaliluku
FV	Tuleva arvo	reaaliluku
PpY	Maksuerien määrä vuodessa, oletusarvo=1	kokonaisluku > 0
CpY	Korkojaksoja vuodessa, oletusarvo=1	kokonaisluku > 0
PmtAt	Erääntyvän maksun määrä kunkin jakson lopussa tai alussa, oletusarvo=loppu	kokonaisluku (0=loppu, 1=alku)

***** Nämä rahan aika-arvon argumenttien nimet ovat samat kuin TVM-muuttujien nimet (kuten **tvm.pv** ja **tvm.pmt**), joita käytetään *Laskin*-sovelluksen talouslaskentatoiminnossa.Talouslaskentafunktioiden argumenttien arvot tai vastaukset eivät kuitenkaan tallennu TVM-muuttujiin.

TwoVar Katalogi >

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

Laskee kahden muuttujan tilastot. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

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 numeeristen tai merkkijonojen 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 tyhjistä elementeistä, katso sivu 223.

Tulosmuuttuja	**Kuvaus**
stat.$\bar{x}$	x:n arvojen keskiarvo
stat. x	x:n arvojen summa
stat. x2	x2:n arvojen summa
stat.sx	x:n otoksen keskihajonta
stat. x	x:n perusjoukon keskihajonta
stat.n	Datapisteiden lukumäärä
stat.$\bar{y}$	y:n arvojen keskiarvo
stat.Σy	y:n arvojen summa
stat.Σy^2	y2:n arvojen summa
stat.sy	y:n otoksen keskihajonta
stat.σy	y:n perusjoukon keskihajonta
stat.Σxy	x ·y -arvojen summa
stat.r	Korrelaatiokerroin
stat.MinX	x:n arvojen minimi

Tulosmuuttuja	Kuvaus
stat.Q_1X	x:n ensimmäinen neljännes
stat.MedianX	x:n mediaani
stat.Q_3X	x:n 3. neljännes
stat.MaxX	x:n arvojen maksimi
stat.MinY	y:n arvojen minimi
stat.Q_1Y	y:n ensimmäinen neljännes
stat.MedY	y:n mediaani
stat.Q_3Y	x:n kolmas neljännes
stat.MaxY	y:n arvojen maksimi
stat.$\Sigma(x-\bar{x})^2$	x:n keskiarvon poikkeamien neliöiden summa
stat.$\Sigma(y-\bar{y})^2$	y:n keskiarvon poikkeamien neliöiden summa

U

unitV() Katalogi >

yksikköV(*Vektori1***)⇒***vektori*

Laskee joko rivi- tai sarakeyksikkövektorin riippuen *Vektori1*:n muodosta.

Vektori1 :n on oltava joko yksirivinen matriisi tai yksisarakkeinen matriisi.

```
unitV([1  2  1])
                [0.408248  0.816497  0.408248]
unitV([1;2;3])                  [0.267261;0.534522;0.801784]
```

unLock Katalogi >

unLock*Muutt1*[, *Muutt2*] [, *Muutt3*] ...
unLock*Muutt***.**

Vapauttaa määritetyt muuttujat tai muuttujaryhmän. Lukittuja muuttujia ei voi muokata eikä poistaa.

Katso **Lock,** sivu 90, ja **getLockInfo(), sivu 67.**

```
a:=65                                   65
Lock a                                  Done
getLockInfo(a)                           1
a:=75                  "Error: Variable is locked."
DelVar a               "Error: Variable is locked."
Unlock a                                Done
a:=75                                   75
DelVar a                                Done
```

varPop() Katalogi >

varPop(*Lista***[,** *frekvLista***])⇒***lauseke*

Laskee *Lista*n perusjoukon varianssin.

Jokainen *frekvLista*n elementti näyttää *Lista*n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Lista*ssa 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 tyhjistä elementeistä, katso sivu 223.

$\text{varPop}(\{5,10,15,20,25,30\})$ 72.9167

varSamp() Katalogi >

varSamp(*Lista***[,** *frekvLista***])⇒***lauseke*

Laskee *Lista*n otosten varianssin.

Jokainen *frekvLista*n elementti näyttää *Lista*n vastaavien elementtien peräkkäisten esiintymien lukumäärän.

Huomaa: *Lista*ssa 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 tyhjistä elementeistä, katso sivu 223.

$\text{varSamp}(\{1,2,5,-6,3,-2\})$ $\frac{31}{2}$

$\text{varSamp}(\{1,3,5\},\{4,6,2\})$ $\frac{68}{33}$

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

Laskee rivivektorin, joka sisältää *Matriisi1*:n kaikkien sarakkeiden otoksen varianssin.

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

$\text{varSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}\right)$ $\begin{bmatrix} 4.75 & 1.03 & 4 \end{bmatrix}$

$\text{varSamp}\left(\begin{bmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{bmatrix},\begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}\right)$ $\begin{bmatrix} 3.91731 & 2.08411 \end{bmatrix}$

varSamp() Katalogi >

Jos jokin elementti jommassakummassa matriisissa on tyhjä, kyseistä elementtiä ei huomioida, eikä toisessa matriisissa olevaa vastaavaa elementtiä myöskään huomioida. Lisätietoja tyhjistä elementeistä, katso sivu 223.

Huomaa: *Matriisi1*:ssä on oltava vähintään kaksi elementtiä.

W

Wait Luettelo >

Wait *aikaSekunteina*

Keskeyttää toiminnon *aikaSekunteina* sekunnin ajaksi.

Wait on erityisen käyttökelpoinen ohjelmassa, jossa tarvitaan lyhyttä viivettä pyydettyjen tietojen saamiseksi käyttöön.

Argumentin *aikaSekunteina* on oltava lausekkeen muodossa, joka yksinkertaistaa desimaaliarvoon välillä 0–100. Komento pyöristää tämän arvon lähimpään 0,1 sekuntiin.

Peruuttaaksesi **Wait** käynnissä olevan,

- **Kämmenlaite:** Pidä [on]-painiketta painettuna ja paina toistuvasti [enter]-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 4 sekuntia:

Wait 4

Odottaaksesi 1/2 sekuntia:

Wait 0.5

Odottaaksesi 1,3 sekuntia käyttäen muuttujaa *seklukema*:

seklukema:=1.3
Wait seklukema

Tämä esimerkki sytyttää vihreän LED-valon päälle 0,5 sekunnin ajaksi ja sammuttaa sen sitten.

Send "SET GREEN 1 ON"
Wait 0.5
Send "SET GREEN 1 OFF"

warnCodes ()

Katalogi >

warnCodes(*Expr1***,** *StatusVar***)⇒***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 239.

warnCodes(det([1.23456E-999]),*warn*)	1.23456E-999
warn	{10029}

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(*x*<0,*x*+3)|*x*=5	undef

when() on hyödyllinen komento rekursiivisten funktioiden määrittämisessä.

when(*n*>0,*n*·*factoral*(*n*−1),1)→*factoral*(*n*)	*Done*
factoral(3)	6
3!	6

While

Katalogi >

While *Ehto*
Lohko
EndWhile

Suorittaa *Lohko*n 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 määritysten syöttämisestä löytyvät tuotteen ohjekirjan Laskin-osiosta.

```
Define sum_of_recip(n)=Func
                       Local i,tempsum
                       1→i
                       0→tempsum
                       While i≤n
                       tempsum+1/i→tempsum
                       i+1→i
                       EndWhile
                       Return tempsum
                       EndFunc
                                          Done
sum_of_recip(3)                           11/6
```

X

xor

Katalogi >

*BoolenLaus1***xor***BoolenLaus2* antaa vastauksena *Boolen lausekkeen*
BoolenList1 **xor***BoolenList2* antaa vastauksena *Boolen listan*
BoolenMatriisi1 **xor***BoolenMatriisi2* antaa vastauksena *Boolen matriisin*

Määrittää totuusarvoksi tosi, jos *BoolenLaus1* on tosi ja *BoolenLaus2* on epätosi, tai päin vastoin.

Määrittää totuusarvoksi epätosi, jos kumpikin argumentti on tosi tai kumpikin on epätosi. Antaa vastauksena sievennetyn Boolen lausekkeen, jos kummankaan argumentin totuusarvoa ei voi määrittää todeksi tai epätodeksi.

Huomaa: Katso **or, sivu 113.**

```
true xor true                             false
5>3 xor 3>5                               true
```

Kokonaisluku1 **xor** *Kokonaisluku2* ⇒ *kokonaisluku*

Heksadesimaalisessa kantalukutilassa:

Tärkeää: Nolla, ei O-kirjain.

```
0h7AC36 xor 0h3D5F                        0h79169
```

Binaarisessa kantalukutilassa:

xor

Katalogi >

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 kantalukutilan mukaisesti.

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

Jos syötät desimaalikokonaisluvun, joka on etumerkillisen, 64 bitin binaarimuodon lukualueen ulkopuolella, laskin käyttää symmetristä modulo-operaatiota, jotta arvo saadaan oikealle alueelle. Lisätietoja, katso ▶**Base2**, sivu 16.

Huomaa: Katso **or, sivu 113.**

0b100101 xor 0b100	0b100001

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

Z

zInterval

Katalogi >

zInterval σ,*Lista*[,*Frekv*[,*CTaso*]]

(Datalistan syöte)

zInterval σ,$\bar{x}$,*n* [,*CTaso*]

(Yhteenvetotilaston syöte)

Laskee *z*-luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

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 *Lista* tunnettu perusjoukon keskihajonta

zInterval_1Prop

Katalogi >

zInterval_1Prop *x*,*n* [,*CTaso*]

Laskee yhden osuuden *z*-luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

x on ei-negatiivinen kokonaisluku.

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.CLower, stat.CUpper	Luottamusväli, joka sisältää jakauman luottamusvälin todennäköisyyden
stat.$\hat{p}$	Laskettu onnistumisten osuus
stat.ME	Virhemarginaali
stat.n	Otosten lukumäärä datasekvenssissä

zInterval_2Prop

Katalogi >

zInterval_2Prop *x1*,*n1*,*x2*,*n2*[,*CTaso*]

Laskee kahden osuuden *z*-luottamusvälin. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

x1 ja *x2* ovat ei-negatiivisia kokonaislukuja.

zInterval_2Prop

Katalogi >

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

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

Katalogi >

zInterval_2Samp σ_1,σ_2,*Lista1*,*Lista2* [,*Frekv1*[,*Frekv2*,[*CTaso*]]]

(Datalistan syöte)

zInterval_2Samp σ_1,σ_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 154.)

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

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

Tulosmuuttuja	Kuvaus
stat.σx1, stat.σx2	*Lista 1*:n ja *Lista 2*:n otosten keskihajonnat
stat.n1, stat.n2	Otosten lukumäärä datasekvensseissä
stat.r1, stat.r2	Datasekvenssien *Lista 1* ja *Lista 2* tunnetut perusjoukon keskihajonnat

zTest

Katalogi >

zTest μ*0*,σ,*Lista*,[*Frekv*[,*Hypot*]]

(Datalistan syöte)

zTest μ*0*,σ,x̄,*n*[,*Hypot*]

(Yhteenvetotilaston syöte)

Suorittaa *z*-testin frekvenssillä *frekvlista*. Tulosten yhteenveto tallentuu *stat.results*-muuttujaan. (Katso sivu 154.)

Testaa H_0: μ = μ0, jonkin seuraavan vaihtoehdon suhteen:

Kun H_a: μ < μ0, aseta *Hypot*<0
Kun H_a: μ ≠ μ0 (oletus), aseta *Hypot*=0
Kun H_a: μ > μ0, aseta *Hypot*>0

Lisätietoja listassa olevien tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.z	(x̄ − μ0) / (σ / sqrt(n))
stat.P Value	Pienin todennäköisyys, jolla nollahypoteesi voidaan hylätä
stat.x̄	*Lista*n sisältämän datasekvenssin otoksen keskiarvo
stat.sx	Datasekvenssin otoksen keskihajonta. Lasketaan vain *Data*-syötteelle.
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 154.)

zTest_1Prop

Katalogi >

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 223.

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 154.)

$x1$ 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 tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.z	Osuuksien erotukselle laskettu vakio-ohjearvo
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä

Tulosmuuttuja	Kuvaus
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

Katalogi >

zTest_2Samp σ_1,σ_2 ,*Lista1*,*Lista2*[,*Frekv1*[,*Frekv2*[,*Hypot*]]]

(Datalistan syöte)

zTest_2Samp σ_1,σ_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 154.)

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 tyhjien elementtien vaikutuksesta, katso Tyhjät elementitsivulla sivu 223.

Tulosmuuttuja	Kuvaus
stat.z	Keskiarvojen erotukselle laskettu vakio-ohjearvo
stat.PVal	Alin merkitsevyystaso, jolla nollahypoteesi voidaan hylätä
stat.$\bar{x}1$, stat.$\bar{x}2$	*Lista1*:n ja *Lista2*:n sisältämien datasekvenssien otosten keskiarvot
stat.sx1, stat.sx2	*Lista1*:n ja *Lista2*:n sisältämien datasekvenssien otosten keskihajonnat
stat.n1, stat.n2	Otosten koko

Symbolit

+ (yhteenlasku) — ⊞ painike

Arvo1 + *Arvo2*⇒*arvo*

Laskee kahden argumentin summan.

56	56
56+4	60
60+4	64
64+4	68
68+4	72

Lista1 + *Lista2*⇒*listaMatriisi1*
+ *Matriisi2*⇒*matriisi*

Määrittää listan (tai matriisin), joka sisältää *Lista1*:n ja *Lista2*:n (tai *Matriisi1*:n ja *Matriisi2*:n) vastaavien elementtien summat.

Argumenttien tulee olla mitoiltaan samanlaisia.

$\left\{22,\pi,\frac{\pi}{2}\right\}\rightarrow l1$	$\{22,3.14159,1.5708\}$
$\left\{10,5,\frac{\pi}{2}\right\}\rightarrow l2$	$\{10,5,1.5708\}$
$l1+l2$	$\{32,8.14159,3.14159\}$

Arvo + *Lista1*⇒*lista*

Lista1 + *Arvo*⇒*lista*

Määrittää listan, joka sisältää *Arvo*n ja *Lista1*:n kunkin elementin summat.

$15+\{10,15,20\}$	$\{25,30,35\}$
$\{10,15,20\}+15$	$\{25,30,35\}$

Arvo + *Matriisi1*⇒*matriisi*

Matriisi1 + *Arvo*⇒*matriisi*

Laskee matriisin, jossa *Arvo* on lisätty jokaiseen elementtiin *Matriisi1*:n diagonaalimatriisissa. *Matriisi1* :n on oltava neliö.

Huomaa: Käytä merkintää **.+** (piste plus), kun haluat lisätä lausekkeen jokaiseen elementtiin.

$20+\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$	$\begin{bmatrix}21 & 2\\3 & 24\end{bmatrix}$

−(vähennyslasku) — ⊟ painike

Arvo1 − *Arvo2*⇒*value*

Laskee *Arvo1* miinus *Arvo2*.

6−2	4
$\pi-\frac{\pi}{6}$	2.61799

−(vähennyslasku) — [−] painike

Lista1 − Lista2⇒lista

Matriisi1 − Matriisi2⇒matriisi

Vähentää *Lista2*:n (tai *Matriisi2*:n) jokaisen elementin *Lista1*:n (tai *Matriisi1*:n) vastaavasta elementistä ja antaa tuloksena vastaukset.

Argumenttien tulee olla mitoiltaan samanlaisia.

$\left\{22,\pi,\frac{\pi}{2}\right\}-\left\{10,5,\frac{\pi}{2}\right\}$ → $\{12,{}^{-}1.85841,0.\}$

$[3\ \ 4]-[1\ \ 2]$ → $[2\ \ 2]$

Arvo − Lista1⇒lista
Lista1 − Arvo⇒lista

Vähentää jokaisen *Lista1*:n elementin *Arvo*sta tai vähentää *Arvo*n jokaisesta *Lista1*:n elementistä ja antaa vastauksena tuloslistan.

$15-\{10,15,20\}$ → $\{5,0,{}^{-}5\}$

$\{10,15,20\}-15$ → $\{{}^{-}5,0,5\}$

Arvo − Matriisi1⇒matriisi
Matriisi1 − Arvo⇒matriisi

Arvo − Matriisi1 laskee *Arvo*n matriisin kerrottuna identtisellä matriisilla miinus *Matriisi1*. *Matriisi1*:n on oltava neliö.

Matriisi1 − Arvo laskee *Arvo* n matriisin kerrottuna identtisellä matriisilla, joka on vähennetty *Matriisi1*:stä. *Matriisi1*:n on oltava neliö.

$20-\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$ → $\begin{bmatrix}19 & {}^{-}2\\ {}^{-}3 & 16\end{bmatrix}$

Huomaa: Käytä merkintää **.−** (piste miinus), kun haluat vähentää lausekkeen jokaisesta elementistä.

·(kertolasku) — [×] painike

Arvo1 ·Arvo2⇒arvo

Laskee kahden argumentin tulon.

$2\cdot 3.45$ → 6.9

Lista1 ·Lista2⇒lista

Luo listan, joka sisältää *Lista1*:n ja *Lista2*:n vastaavien elementtien tulot.

Listojen tulee olla mitoiltaan samanlaisia.

$\{1.,2,3\}\cdot\{4,5,6\}$ → $\{4,10,18\}$

·(kertolasku) — × painike

Matriisi1 ·*Matriisi2*⇒*matriisi*

Laskee *Matriisi1*:n ja *Matriisi2*:n matriisitulon.

Matriisi1:n sarakkeiden lukumäärän on oltava sama kuin *Matriisi2*:n rivien lukumäärä.

$$\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}\cdot\begin{bmatrix}7&8\\7&8\\7&8\end{bmatrix} \qquad \begin{bmatrix}42&48\\105&120\end{bmatrix}$$

Arvo ·*Lista1*⇒*lista*

Lista1 ·*Arvo*⇒*lista*

Määrittää listan, joka sisältää *Arvo*n ja kunkin *Lista1*:n elementin tulon.

$$\pi\cdot\{4,5,6\} \qquad \{12.5664,15.708,18.8496\}$$

Arvo ·*Matriisi1*⇒*matriisi*

Matriisi1 ·*Arvo*⇒*matriisi*

Määrittää matriisin, joka sisältää *Arvo*n ja kunkin *Matriisi1*:n elementin tulon.

$$\begin{bmatrix}1&2\\3&4\end{bmatrix}\cdot 0.01 \qquad \begin{bmatrix}0.01&0.02\\0.03&0.04\end{bmatrix}$$

$$6\cdot\text{identity}(3) \qquad \begin{bmatrix}6&0&0\\0&6&0\\0&0&6\end{bmatrix}$$

Huomaa: Käytä merkintää **.** ·(piste kerro), kun haluat kertoa lausekkeen jokaisella elementillä.

/ (jakolasku) — ÷ painike

Arvo1 / *Arvo2*⇒*arvo*

Laskee osamäärän *Arvo1* jaettuna *Arvo2*:lla.

Huomaa: Katso myös **Murtolukumalli**, sivu 1.

$$\frac{2}{3.45} \qquad .57971$$

Lista1 / *Lista2*⇒*lista*

Määrittää listan, joka sisältää osamäärät laskutoimituksista *Lista1* jaettuna *Lista2*:lla.

Listojen tulee olla mitoiltaan samanlaisia.

$$\frac{\{1.,2,3\}}{\{4,5,6\}} \qquad \left\{0.25,\frac{2}{5},\frac{1}{2}\right\}$$

Arvo / *Lista1* ⇒ *lista*

Lista1 / *Arvo* ⇒ *lista*

Määrittää listan, joka sisältää osamäärät laskutoimituksista *Arvo* jaettuna *Lista1*:llä tai *Lista1* jaettuna *Arvo*lla.

$$\frac{6}{\{3,6,\sqrt{6}\}} \qquad \{2,1,2.44949\}$$

$$\frac{\{7,9,2\}}{7\cdot 9\cdot 2} \qquad \left\{\frac{1}{18},\frac{1}{14},\frac{1}{63}\right\}$$

/ (jakolasku) ÷ painike

Arvo / *Matriisi1* ⇒ *matriisi*

Matriisi1 / *Arvo* ⇒ *matriisi*

Laskee matriisin, joka sisältää osamäärät laskutoimituksesta *Matriisi1*/*Arvo*.

Huomaa: Käytä merkintää **./** (piste jaa), kun haluat jakaa lausekkeen jokaisella elementillä.

$$\frac{[7\ \ 9\ \ 2]}{7\cdot 9\cdot 2} \qquad \left[\frac{1}{18}\ \ \frac{1}{14}\ \ \frac{1}{63}\right]$$

^ (potenssi) ^ painike

Arvo1 **^** *Arvo2* ⇒ *arvo*

Lista1 **^** *Lista2* ⇒ *lista*

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.

$$4^2 \qquad 16$$

$$\{2,4,6\}^{\{1,2,3\}} \qquad \{2,16,216\}$$

Arvo **^** *Lista1* ⇒ *lista*

Laskee *Arvo*n korotettuna *Lista1*:n elementtien potenssiin.

$$\pi^{\{1,2,-3\}} \qquad \{3.14159,9.8696,0.032252\}$$

Lista1 **^** *Arvo* ⇒ *lista*

Laskee *Lista1*:n elementit korotettuna *Arvo*n potenssiin.

$$\{1,2,3,4\}^{-2} \qquad \left\{1,\frac{1}{4},\frac{1}{9},\frac{1}{16}\right\}$$

^ (potenssi) — [^] painike

neliömatriisi1 **^** *kokonaisluku* ⇒ *matriisi*

Laskee *neliömatriisi1*:n korotettuna kokonaisluvun *kokonaisluku* potenssiin.

neliömatriisi1:n on oltava neliömatriisi.

Jos *kokonaisluku* = ⁻1, laskee käänteismatriisin.
Jos *kokonaisluku* < ⁻1, laskee käänteismatriisin korotettuna sopivaan positiiviseen potenssiin.

$$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{2} \qquad \begin{bmatrix}7&10\\15&22\end{bmatrix}$$

$$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{-1} \qquad \begin{bmatrix}-2&1\\\frac{3}{2}&\frac{-1}{2}\end{bmatrix}$$

$$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{-2} \qquad \begin{bmatrix}\frac{11}{2}&\frac{-5}{2}\\\frac{-15}{4}&\frac{7}{4}\end{bmatrix}$$

x^2 (neliö) — [x²] painike

Arvo1^2 ⇒ *arvo*

Laskee argumentin neliön.

Lista1^2 ⇒ *lista*

Laskee listan, joka sisältää *Lista1*:n elementtien neliöt.

neliömatriisi1^2 ⇒ *matriisi*

Laskee *neliömatriisi1*:n matriisineliön. Tämä ei ole sama kuin kunkin elementin neliön laskeminen. Käytä merkintää .^2, kun haluat laskea jokaisen elementin neliön.

$$4^2 \qquad 16$$

$$\{2,4,6\}^2 \qquad \{4,16,36\}$$

$$\begin{bmatrix}2&4&6\\3&5&7\\4&6&8\end{bmatrix}^2 \qquad \begin{bmatrix}40&64&88\\49&79&109\\58&94&130\end{bmatrix}$$

$$\begin{bmatrix}2&4&6\\3&5&7\\4&6&8\end{bmatrix}.\hat{}\ 2 \qquad \begin{bmatrix}4&16&36\\9&25&49\\16&36&64\end{bmatrix}$$

.+ (piste lisää) — [.][+] painikkeet

Matriisi1 **.+** *Matriisi2* ⇒ *matriisi*

Arvo **.+** *Matriisi1* ⇒ *matriisi*

Matriisi1 **.+** *Matriisi2* laskee matriisin, joka on *Matriisi1*:n ja *Matriisi2*:n vastaavien elementtiparien summa.

Arvo **.+** *Matriisi1* laskee matriisin, joka on *Arvo*n ja kunkin *Matriisi1*:n elementin summa.

$$\begin{bmatrix}1&2\\3&4\end{bmatrix}.+\begin{bmatrix}10&30\\20&40\end{bmatrix} \qquad \begin{bmatrix}11&32\\23&44\end{bmatrix}$$

$$5.+\begin{bmatrix}10&30\\20&40\end{bmatrix} \qquad \begin{bmatrix}15&35\\25&45\end{bmatrix}$$

.− (piste-erotus)

. − painikkeet

Matriisi1 **.−** *Matriisi2* ⇒ *matriisi*

Arvo **.−***Matriisi1* ⇒ *matriisi*

Matriisi1 **.−***Matriisi2* laskee matriisin, joka on *Matriisi1*:n ja *Matriisi2*:n vastaavien elementtiparien välinen erotus.

Arvo **.−***Matriisi1* laskee matriisin, joka on *Arvo*n ja kunkin *Matriisi1*:n elementin erotus.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} .- \begin{bmatrix}10 & 20\\30 & 40\end{bmatrix} \qquad \begin{bmatrix}-9 & -18\\-27 & -36\end{bmatrix}$$

$$5 .- \begin{bmatrix}10 & 20\\30 & 40\end{bmatrix} \qquad \begin{bmatrix}-5 & -15\\-25 & -35\end{bmatrix}$$

. ·(pistetulo)

. × painikkeet

Matriisi1 **.·** *Matriisi2* ⇒ *matriisi*

Arvo **.·***Matriisi1* ⇒ *matriisi*

Matriisi1 **.·** *Matriisi2* laskee matriisin, joka on *Matriisi1*:n ja *Matriisi2*:n vastaavien elementtiparien tulo.

Arvo **.·** *Matriisi1* laskee matriisin, joka sisältää *Arvo*n ja kunkin *Matriisi1*:n elementin tulot.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} .\cdot \begin{bmatrix}10 & 20\\30 & 40\end{bmatrix} \qquad \begin{bmatrix}10 & 40\\90 & 160\end{bmatrix}$$

$$5 .\cdot \begin{bmatrix}10 & 20\\30 & 40\end{bmatrix} \qquad \begin{bmatrix}50 & 100\\150 & 200\end{bmatrix}$$

. / (pisteosamäärä)

. ÷ painikkeet

Matriisi1 **./** *Matriisi2* ⇒ *matriisi*

Arvo **./** *Matriisi1* ⇒ *matriisi*

Matriisi1 **./** *Matriisi2* laskee matriisin, joka on *Matriisi1*:n ja *Matriisi2*:n vastaavien elementtiparien osamäärä.

Arvo **./** *Matriisi1* laskee matriisin, joka on *Arvo*n ja kunkin *Matriisi1*:n elementin osamäärä.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} ./ \left(\begin{bmatrix}10 & 20\\30 & 40\end{bmatrix}\right) \qquad \begin{bmatrix}\frac{1}{10} & \frac{1}{10}\\\frac{1}{10} & \frac{1}{10}\end{bmatrix}$$

$$5 ./ \left(\begin{bmatrix}10 & 20\\30 & 40\end{bmatrix}\right) \qquad \begin{bmatrix}\frac{1}{2} & \frac{1}{4}\\\frac{1}{6} & \frac{1}{8}\end{bmatrix}$$

.^ (pistepotenssi) — [.][^] painikkeet

Matriisi1 **.^** *Matriisi2* ⇒ *matriisi*

Arvo **. ^** *Matriisi1* ⇒ *matriisi*

Matriisi1 **.^** *Matriisi2* laskee matriisin, jossa *Matriisi2*:n jokainen elementti on eksponentti *Matriisi1*:n vastaavalle elementille.

Arvo **.^** *Matriisi1* laskee matriisin, jossa *Matriisi1*:n jokainen elementti on eksponentti *Arvo*lle.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} .^{\wedge} \begin{bmatrix}0 & 2\\3 & -1\end{bmatrix} \qquad \begin{bmatrix}1 & 4\\27 & \frac{1}{4}\end{bmatrix}$$

$$5 .^{\wedge} \begin{bmatrix}0 & 2\\3 & -1\end{bmatrix} \qquad \begin{bmatrix}1 & 25\\125 & \frac{1}{5}\end{bmatrix}$$

-(negaatio) — [(−)] painike

-*Arvo1* ⇒ *arvo*

-*Lista1* ⇒ *lista*

-*Matriisi1* ⇒ *matriisi*

Laskee argumentin negaation.

Kun kyseessä on lista tai matriisi, määrittää kaikkien elementtien negaatiot.

Jos argumentti on binaarinen tai heksadesimaalinen kokonaisluku, negaatio antaa kakkosen komplementin.

```
-2.43                          -2.43
-{-1,0.4,1.2E19}     {1.,-0.4,-1.2E19}
```

Binaarisessa kantalukutilassa:

```
-0b100101
0b11111111111111111111111111111111▸
```

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

% (prosentti) — [ctrl][📖] painikkeet

Arvo1 **%** ⇒ *arvo*

Lista1 **%** ⇒ *lista*

Matriisi1 **%** ⇒ *matriisi*

Määrittää $\frac{argument}{100}$

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*⇒*Boolen lauseke*

Lista1 = *Lista2*⇒ *Boolen lista*

Matriisi1 = *Matriisi2*⇒ *Boolen 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.

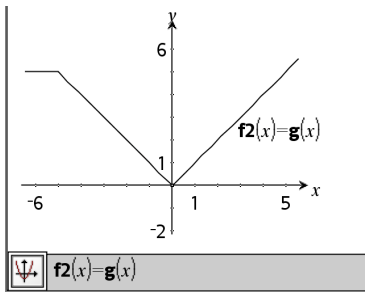
Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

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 ≠ *Laus2* ⇒ *Boolen lauseke*

Lista1≠ *Lista2* ⇒ *Boolen lista*

Matriisi1≠ *Matriisi2* ⇒ *Boolen 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).

≠ (ei yhtä kuin)

ctrl = **painikkeet**

Listojen ja matriisien kohdalla vastauksena on vertailuja elementti elementiltä.

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla **/=**

< (pienempi kuin)

ctrl = **painikkeet**

Laus1 **<** *Laus2* ⇒ *Boolen lauseke*

Lista1 **<** *Lista2* ⇒ *Boolen lista*

Matriisi1 **<** *Matriisi2* ⇒ *Boolen 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.

Katso esimerkki kohdasta “=” (on yhtä kuin).

≤ (pienempi tai yhtä suuri kuin)

ctrl = **painikkeet**

Laus1 ≤ *Laus2* ⇒ *Boolen lauseke*

Lista1 ≤ *Lista2* ⇒ *Boolen lista*

Matriisi1 ≤ *Matriisi2* ⇒ *Boolen 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.

Katso esimerkki kohdasta “=” (on yhtä kuin).

≤ (pienempi tai yhtä suuri kuin)

ctrl = painikkeet

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla **<=**

> (suurempi kuin)

ctrl = painikkeet

Laus1 **>** *Laus2* ⇒ *Boolen lauseke*

Lista1 **>** *Lista2* ⇒ *Boolen lista*

Matriisi1 **>** *Matriisi2* ⇒ *Boolen 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.

Katso esimerkki kohdasta “=” (on yhtä kuin).

≥ (suurempi tai yhtä suuri kuin)

ctrl = painikkeet

Laus1 ≥ *Laus2* ⇒ *Boolen lausek*

Lista1 ≥ *Lista2* ⇒ *Boolen lista*

Matriisi1 ≥ *Matriisi2* ⇒ *Boolen 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 **>=**

Katso esimerkki kohdasta “=” (on yhtä kuin).

⇒ (looginen seuraus) ctrl = näppäimet

BoolenLaus1 ⇒*BoolenLaus2*antaa vastauksena *Boolen lausekkeen*

BoolenList1⇒ *BoolenList2* antaa vastauksena *Boolen listan*

BoolenMatriisi1⇒ *BoolenMatriisi2* antaa vastauksena *Boolen matriisin*

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 **=>**

5>3 or 3>5	true
5>3 ⇒ 3>5	false
3 or 4	7
3 ⇒ 4	-4
{1,2,3} or {3,2,1}	{3,2,3}
{1,2,3} ⇒ {3,2,1}	{-1,-1,-3}

⇔ (looginen kaksoisseuraus, XNOR) ctrl = näppäimet

BoolenLaus1 ⇔*BoolenLaus2*antaa vastauksena *Boolen lausekkeen*

BoolenList1⇔ *BoolenList2* antaa vastauksena *Boolen listan*

BoolenMatriisi1⇔ *BoolenMatriisi2* antaa vastauksena *Boolen matriisin*

Kokonaisluku1⇔ *Kokonaisluku2* antaa vastauksena *kokonaisluvun*

Antaa vastauksena XOR Boolen 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 **<=>**

5>3 xor 3>5	true
5>3 ⇔ 3>5	false
3 xor 4	7
3 ⇔ 4	-8
{1,2,3} xor {3,2,1}	{2,0,2}
{1,2,3} ⇔ {3,2,1}	{-3,-1,-3}

! (kertoma)

?!▸ painike

Arvo1! ⇒ *arvo*

Lista1! ⇒ *lista*

Matriisi1! ⇒ *matriisi*

Määrittää argumentin kertoman.

Jos kyseessä on lista tai matriisi, määrittää listan tai matriisin elementtien kertomista.

5!	120
{5,4,3}!	{120,24,6}
$\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)!$	$\begin{bmatrix}1 & 2\\6 & 24\end{bmatrix}$

& (liitä)

ctrl painikkeet

Merkkijono1 **&** *Merkkijono2* ⇒ *merkkijono*

Antaa vastauksena tekstimerkkijonon, joka on *Merkkijono2* liitettynä *Merkkijono1*:een.

"Hello "&"Nick"	"Hello Nick"

d() (derivaatta)

Katalogi >

d(*Laus1*, *Muutt*[, *Aste*]**) |** *Muutt=Arvo⇒arvo*

d(*Laus1*, *Muutt*[, *Aste*]**)⇒***arvo*

d(*Lista1*, *Muutt*[, *Aste*]**)⇒***lista*

d(*Matriisi1*, *Muutt*[, *Aste*]**)⇒***matriisi*

Lukuun ottamatta tilannetta, jolloin käytät ensimmäistä syntaksia, muuttujaan *Muutt* on tallennettava numeerinen arvo ennen funktion **d()** sieventämistä. Katso esimerkkejä.

d()-funktiota voi käyttää laskettaessa numeerisesti ensimmäisen ja toisen asteen derivaatta pisteessä käyttäen automaattisia derivointimenetelmiä.

*Aste*en, mikäli se otetaan mukaan, on oltava **1** tai **2**. Oletusarvo on **1**.

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla `derivative(...)`.

$\frac{d}{dx}(\lvert x\rvert)\lvert x=0$	undef
$x:=0:\frac{d}{dx}(\lvert x\rvert)$	undef
$x:=3:\frac{d}{dx}(\{x^2,x^3,x^4\})$	{6,27,108}

d() (derivaatta) Katalogi >

Huomaa: Katso myös
Ensimmäinen derivaatta, sivu 5 tai
Toinen derivaatta, sivu 6.

Huomaa: d()-algoritmiin liittyy rajoitus: se laskee rekursiivisesti sieventämättömästä lausekkeesta ensimmäisen (ja toisen, mikäli mahdollista) derivaatan numeerisen arvon ja sieventää jokaisen alalausekkeen, mistä voi olla tuloksena odottamaton vastaus.

Tarkastele oikealla olevaa esimerkkiä. Yhtälön x·(x^2+x)^(1/3), kun x=0, ensimmäinen derivaatta on yhtä kuin 0. Koska alalausekkeen (x^2+x)^(1/3) ensimmäinen derivaatta kuitenkin on määrittämätön pisteessä x=0, ja tällä arvolla lasketaan koko lausekkeen derivaatta, **d()** ilmaisee vastauksen määrittämättömänä ja näyttää varoitusviestin.

Jos tämä rajoitus esiintyy, varmista ratkaisu graafisesti. Voit myös kokeilla funktiota **centralDiff()**.

$$\frac{d}{dx}\left(x\cdot\left(x^2+x\right)^{\frac{1}{3}}\right)\Big|_{x=0} \qquad \text{undef}$$

$$\text{centralDiff}\left(x\cdot\left(x^2+x\right)^{\frac{1}{3}},x\right)\Big|_{x=0} \qquad 0.000033$$

∫() (integraali) Katalogi >

∫(*Laus1*, *Muutt*, *Ala*, *Ylä*) ⇒ *arvo*

Laskee *Laus1*:n integraalin muuttujan *Muutt* suhteen välillä *Ala* - *Ylä* olevista arvoista. Funktiota voi käyttää myös laskettaessa numeerisesti määrätty integraali käyttäen samaa menetelmää kuin nInt().

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

Huomaa: Katso myös **nInt()**, sivu 107, ja **Määrätyn integraalin malli**, sivu 6.

$$\int_0^1 x^2\,dx \qquad 0.333333$$

√() (neliöjuuri) — ctrl x² painikkeet

√ (*Arvo1*)⇒*arvo*

√ (*Lista1*)⇒*lista*

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.

$\sqrt{4}$ → 2

$\sqrt{\{9,2,4\}}$ → $\{3,1.41421,2\}$

Π() (tulo) — Katalogi >

Π(*Laus1*, *Muutt*, *Matala*, *Korkea*) ⇒*lauseke*

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

Sieventää *Laus1*:n jokaisen *Muutt*:n arvon suhteen väliltä *Matala* - *Korkea* ja laskee vastausten tulon.

Huomaa: Katso myös **Kertolaskumalli (Π)**, sivu 5.

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

$\prod_{n=1}^{5}\left(\left\{\frac{1}{n},n,2\right\}\right)$ → $\left\{\frac{1}{120},120,32\right\}$

Π(*Laus1*, *Muutt*, *Matala*, *Matala*−1)⇒1

Π(*Laus1*, *Muutt*, *Matala*, *Korkea*) ⇒**1/**Π(*Laus1*, *Muutt*, *Korkea*+1, *Matala*−1), jos *Korkea* < *Matala*−1

$\prod_{k=4}^{3}(k)$ → 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

$\prod_{k=4}^{1}\left(\frac{1}{k}\right)\cdot\prod_{k=2}^{4}\left(\frac{1}{k}\right)$ → $\frac{1}{4}$

Σ() (summa) Katalogi >

Σ(*Laus1***, *Muutt*, *Matala*, *Korkea***)** ⇒*lauseke*

$$\sum_{n=1}^{5}\left(\frac{1}{n}\right) \qquad \frac{137}{60}$$

Huomaa: Voit syöttää tämän funktion näppäimistöltä kirjoittamalla `sumSeq(...)`.

Sieventää *Laus1*:n jokaisen muuttujan *Muutt* arvon suhteen väliltä *Matala* - *Korkea* ja laskee vastausten summan.

Huomaa: Katso myös **Summamalli**, sivu 5.

Σ(*Laus1***, *Muutt*, *Matala*, *Matala*−1)**⇒0

$$\sum_{k=4}^{3}(k) \qquad 0$$

Σ(*Laus1***, *Muutt*, *Matala*, *Korkea***) ⇒⁻Σ(***Laus1***, *Muutt*, *Korkea*+1, *Matala*−1)**, jos *Korkea* < *Matala*−1

$$\sum_{k=4}^{1}(k) \qquad -5$$

Esimerkkien yhteenlaskukaavat on otettu seuraavasta viitteestä:

$$\sum_{k=4}^{1}(k)+\sum_{k=2}^{4}(k) \qquad 4$$

Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

ΣInt() Katalogi >

ΣInt(*NPmt1***, *NPmt2*, *N*, *I*, *PV* ,[*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*pyörArvo*])**⇒*arvo*

ΣInt(1,3,12,4.75,20000,,12,12) −213.48

ΣInt(*NPmt1*,*NPmt2*,*amortTable***)**⇒*arvo*

Lyhennystoiminto, joka laskee koron summan määritetyn maksueräjakson ajalta.

NPmt1 ja *NPmt2* määrittävät maksujakson alku- ja loppurajat.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 170.

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

ΣInt() Katalogi >

- 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 7.

Huomaa: Katso myös ΣPrn(), jäljempänä, sekä **Bal()**, sivu 16.

tbl:=amortTbl(12,12,4.75,20000,,12,12)

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

ΣInt(1,3,*tbl*) -213.48

ΣPrn() Katalogi >

ΣPrn(*NPmt1***,** *NPmt2***,** *N***,** *I***,** *PV***,** [*Pmt*]**,** [*FV*]**,** [*PpY*]**,** [*CpY*]**,** [*PmtAt*]**,** [*pyörArvo*]**)** ⇒*arvo*

ΣPrn(*NPmt1***,***NPmt2***,***amortTable***)**⇒*arvo*

Lyhennystoiminto, joka laskee pääoman summan määritetyn maksujakson ajalta.

NPmt1 ja *NPmt2* määrittävät maksujakson alku- ja loppurajat.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* ja *PmtAt* on kuvattu TVM-argumenttien taulukossa, sivu 170.

- 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.

ΣPrn(1,3,12,4.75,20000,,12,12) -4916.28

tbl:=amortTbl(12,12,4.75,20000,,12,12)

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

ΣPrn(1,3,*tbl*) -4916.28

ΣPrn() Katalogi >

pyörArvo määrittää pyöristyksessä 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 7.

Huomaa: Katso myös ΣInt(), edellä, sekä **Bal ()**, sivu 16.

(epäsuora operaattori) ctrl painikkeet

*muuttNimiMerkkijono*

Viittaa muuttujaan, jonka nimi on *muuttNimiMerkkijono*. Tällä operaattorilla voit luoda muuttujanimiä funktion sisältä merkkijonojen avulla.

xyz:=12	12
#("x"&"y"&"z")	12

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*

Syöttää luvun kymmenpotenssimuodossa. Luku tulkitaan seuraavasti: *mantissa* × $10^{\text{eksponentti}}$.

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 2.3**E**4.

23000.	23000.
2300000000.+4.1E15	4.1E15
$3\cdot10^4$	30000

^{g} (graadi) π▸ painike

Laus1^{g}⇒*lauseke*

Lista1^{g}⇒*lista*

Matriisi1^{g}⇒*matriisi*

Tämän funktion avulla voit määrittää graadikulman ollessasi aste- tai radiaanikulmatilassa.

Kun laskin on radiaanikulmatilassa, kertoo *Laus1*:n arvolla π/200.

Kun laskin on astekulmatilassa, kertoo *Laus1*:n arvolla g/100.

Graadikulmatilassa antaa vastauksena lausekkeen *Laus1* muuttumattomana.

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla `@g`.

Aste-, graadi- tai radiaanikulmatilassa.

$\cos(50^{g})$	0.707107
$\cos(\{0,100^{g},200^{g}\})$	$\{1,0.,-1.\}$

^{r} (radiaani) π▸ painike

Arvo1^{r}⇒*arvo*

Lista1^{r}⇒*lista*

Matriisi1^{r}⇒*matriisi*

Tämän funktion avulla voit määrittää radiaanikulman ollessasi aste- tai graadikulmatilassa.

Kun laskin on astekulmatilassa, kertoo argumentin arvolla 180/π.

Radiaanikulmatilassa antaa vastauksena argumentin muuttumattomana.

Kun laskin on graadikulmatilassa, kertoo argumentin arvolla 200/π.

Vinkki: Käytä komentoa ^{r}, jos haluat pakottaa funktion määritelmän yksiköksi radiaanit riippumatta tilasta, joka on käytössä funktion käytön aikana.

Aste-, graadi- tai radiaanikulmatilassa:

$\cos\left(\frac{\pi}{4^{r}}\right)$	0.707107
$\cos\left(\left\{0^{r},\left(\frac{\pi}{12}\right)^{r},-(\pi)^{r}\right\}\right)$	$\{1,0.965926,-1.\}$

ʳ (radiaani) π▸ painike

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla **@r**.

° (aste) π▸ painike

Arvo1°⇒arvo

Lista1°⇒lista

Matriisi1°⇒matriisi

Tämän funktion avulla voit määrittää astekulman ollessasi graadi- tai radiaanikulmatilassa.

Kun laskin on radiaanikulmatilassa, kertoo argumentin arvolla π/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:

$\cos(45°)$	0.707107

Radiaanikulmatilassa:

Huom: Vastauksen pakottaminen likimääräiseksi:

Kämmenlaite: Paina ctrl enter.
Windows®: Paina **Ctrl+Enter**.
Macintosh®: Paina ⌘+**Enter**.
iPad®: Pidä **enter** ja valitse ≈.

°, ', " (astetta/minuuttia/sekuntia) ctrl 📖 painikkeet

dd°mm'ss.ss"⇒lauseke

*dd*Positiivinen tai negatiivinen luku
*mm*Ei-negatiivinen luku
*ss.ss*Ei-negatiivinen luku

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.

Astekulmatilassa:

25°13'17.5"	25.2215
25°30'	$\frac{51}{2}$

°, ', " (astetta/minuuttia/sekuntia)

ctrl 📖 **painikkeet**

Huomaa: Merkitse sekuntien ss.ss perään kaksi heittomerkkiä (''), ei lainausmerkkiä (").

∠ (kulma)

ctrl 📖 **painikkeet**

[*Säde*,∠θ_*Kulma*]⇒*vektori*
(polaarinen syöte)

[*Säde*,∠θ_*Kulma*,*Z_Koordinaatti*]
⇒*vektori*
(lieriömäinen syöte)

[*Säde*,∠θ_*Kulma*,∠θ_*Kulma*]⇒*vektori*
(pallonmuotoinen syöte)

Laskee koordinaatit vektorina riippuen vektorimuototilan asetuksesta: suorakulma, sylinteri tai pallo.

Huomaa: Voit syöttää tämän symbolin tietokoneen näppäimistöltä kirjoittamalla @<.

Radiaanikulmatilassa ja kun vektorimuoto on asetettu valintaan:
suorakulma

$[5 \ \angle 60° \ \angle 45°]$
$[1.76777 \quad 3.06186 \quad 3.53553]$

sylinteri

$[5 \ \angle 60° \ \angle 45°]$
$[3.53553 \quad \angle 1.0472 \quad 3.53553]$

pallo

$[5 \ \angle 60° \ \angle 45°]$
$[5. \quad \angle 1.0472 \quad \angle 0.785398]$

(*Magnitudi* ∠ *Kulma*)⇒*kompleksiArvo*
(polaarinen syöte)

Syöttää kompleksilukuarvon (r∠θ) polaarisessa muodossa. *Kulma* tulkitaan nykyisen kulmatila-asetuksen mukaisesti.

Radiaanikulmatilassa ja suorakulmakompleksimuodossa:

$5+3\cdot i-\left(10 \angle \frac{\pi}{4}\right)$ $\quad -2.07107-4.07107\cdot i$

_ (alaviiva edustaa tyhjää elementtiä)

KatsoTyhjät elementit, sivu 223.

10^()

Katalogi > 📖

10^ (*Arvo1***)⇒***arvo*

10^ (*Lista1***)⇒***lista*

$10^{1.5}$ $\quad 31.6228$

10^() Katalogi >

Laskee luvun 10 korotettuna argumentin potenssiin.

Jos kyseessä on lista, laskee luvun 10 korotettuna *Lista1*:n elementtien potenssiin.

10^(*neliömatriisi1***)⇒***neliömatriisi*

Laskee luvun 10 korotettuna *neliömatriisi1*:n potenssiin. Tämä ei ole sama kuin kunkin elementin 10-potenssiin korottamisen laskeminen. Laskentamenetelmä on kuvattu kohdassa **cos()**.

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

$$10^{\begin{bmatrix}1 & 5 & 3\\ 4 & 2 & 1\\ 6 & -2 & 1\end{bmatrix}}$$

$$\begin{bmatrix}1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6\\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6\\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6\end{bmatrix}$$

^⁻¹(käänteisluku) Katalogi >

Arvo1 **^⁻¹**⇒*arvo*

Lista1 **^⁻¹**⇒*lista*

Laskee argumentin käänteisluvun.

Jos kyseessä on lista, laskee *Lista1*:n elementtien käänteisluvut.

neliömatriisi1 **^⁻¹**⇒*neliömatriisi*

Laskee *neliömatriisi1*:n käänteisluvun.

neliömatriisi1:n oltava ei-singulaarinen neliömatriisi.

$(3.1)^{-1}$ 0.322581

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}^{-1} \qquad \begin{bmatrix}-2 & 1\\ \frac{3}{2} & \frac{-1}{2}\end{bmatrix}$$

| (rajoittava operaattori) ctrl -näppäimet

Laus **|** *Boolenlaus1* [**and***Boolenlaus2*]...

Laus **|** *Boolenlaus1* [**or***Boolenlaus2*]...

$x+1|x=3$ 4

$x+55|x=\sin(55)$ 54.0002

| (rajoittava operaattori)

ctrl -näppäimet

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:

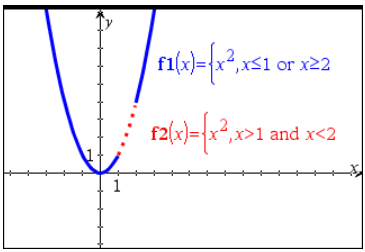
- Sijoitukset
- Välin rajoitusehdot
- Pois rajaaminen

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*.

$x^3-2\cdot x+7\rightarrow f(x)$	*Done*
$f(x)\|x=\sqrt{3}$	8.73205

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.

$\text{nSolve}(x^3+2\cdot x^2-15\cdot x=0,x)$	0.
$\text{nSolve}(x^3+2\cdot x^2-15\cdot x=0,x)\|x>0 \text{ and } x<5$	3.



Pois sulkemisessa käytetään suhdeoperaattoria "ei ole yhtä kuin" (/= tai ≠), jonka avulla jokin tietty arvo suljetaan pois.

→ (tallenna) ctrl var painike

Arvo → *Muutt*

Lista → *Muutt*

Matriisi → *Muutt*

Laus → *Funktio*(*Param1*,...)

Lista → *Funktio*(*Param1*,...) *Matriisi* → *Funktio*(*Param1*,...)

Jos muuttujaa *Muutt* ei ole, laskin luo sen ja alustaa sen muotoon *Arvo*, *Lista* tai *Matriisi*.

Jos muuttuja *Muutt* on jo olemassa eikä se ole lukittu tai suojattu, laskin korvaa sen sisällön arvolla *Arvo*, listalla *Lista* tai matriisilla *Matriisi*.

Huomaa: Voit syöttää tämän operaattorin näppäimistöltä kirjoittamalla **=:** pikavalintana. Kirjoita esimerkiksi **`pi/4 =: myvar`**.

$\frac{\pi}{4} \to myvar$	0.785398
$2\cdot\cos(x) \to y1(x)$	*Done*
$\{1,2,3,4\} \to lst5$	$\{1,2,3,4\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \to matg$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
"Hello" → *str1*	"Hello"

:= (määritä) ctrl painikkeet

Muutt := *Arvo*

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 *Arvo*, *Lista* tai *Matriisi*.

Jos muuttuja *Muutt* on jo olemassa eikä se ole lukittu tai suojattu, laskin korvaa sen sisällön *Arvo*lla, *Lista*lla tai *Matriisi*lla.

$myvar := \frac{\pi}{4}$	.785398
$y1(x) := 2\cdot\cos(x)$	*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:="Hello"	"Hello"

© (kommentti)

ctrl 📖 **painikkeet**

© [*teksti*]

© käsittelee *teksti*n 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)=Func
    © Declare variables
    Local i,result
    result:=0
    For i,1,n,1 ©Loop n times
    result:=result+i^2
    EndFor
    Return result
    EndFunc
                                    Done
g(3)                                  14
```

0b, 0h

0 B **painikkeet,** 0 H **painikkeet**

0b *binaariluku*
0h *heksadesimaaliluku*

Määrittää binaari- (0b) tai heksadesimaaliluvun (0h). Syöttääksesi binaari- tai heksadesimaaliluvun sinun on syötettävä etuliite 0b tai 0h riippumatta kantalukutilasta. Ilman etuliitettä lukua käsitellään desimaalilukuna (kantaluku 10).

Tulokset näytetään kantalukutilan mukaisesti.

Desimaalisessa kantalukutilassa:

```
0b10+0hF+10                           27
```

Binaarisessa kantalukutilassa:

```
0b10+0hF+10                      0b11011
```

Heksadesimaalisessa kantalukutilassa:

```
0b10+0hF+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 tai PublishView-asiakirjassa (.tnsp) 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.

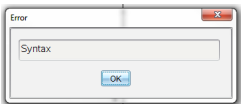
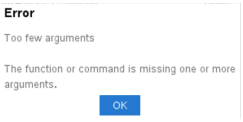
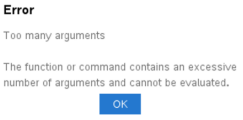
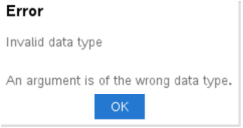
Kuvaajanäyttö	**Oletus**
Korkeus	212
Leveys	318
Väri	valkoinen: 255,255,255

Oletusnäkymä ja asetukset

- Yläpalkin tilakuvakkeet (akun tila, press-to-start-tila, verkon ilmaisin 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.

DrawArc

Katalogi > CXII

DrawArc *x, y, width, height, startAngle, arcAngle*

Piirrä kaari rajaavan suorakulmion sisällä annetulla aloitus- ja kaarikulmalla.

x, *y*: rajaavan suorakulmion vasen yläkoordinaatti

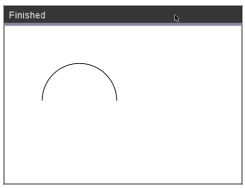
width, *height*: rajaavan suorakulmion mitat

”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

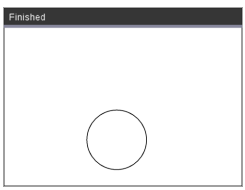
Katalogi > CXII

DrawCircle *x, y, radius*

x, *y*: keskikohdan koordinaatti

radius: ympyrän säde

```
DrawCircle 150,150,40
```



Katso myös: FillCircle

DrawLine

Katalogi > CXII

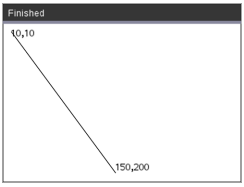
DrawLine *x1, y1, x2, y2*

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

Katalogi > CXII

Komennoilla on kaksi varianttia:

DrawPoly *xlist, ylist*

tai

DrawPoly *x1, y1, x2, y2, x3, y3...xn, yn*

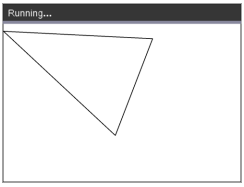
Huomaa: DrawPoly *xlist, ylist*
Muoto yhdistää kohteet *x1, y1* to *x2, y2, x2, y2* to *x3, y3* ja niin edelleen.

Huomaa: DrawPoly *x1, y1, x2, y2, x3, y3...xn, yn*
Kohteita *xn, yn* **EI** yhdistetä automaattisesti kohteisiin *x1, y1*.

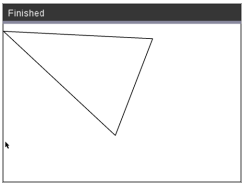
Lausekkeet, jotka arvioidaan todellisten liukulukujen luetteloon
xlist, ylist

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
```



DrawPoly

Katalogi > CXII

Huomaa: DrawPoly: Lisää kokomitat (leveys/korkeus), jotka vastaavat piirrettyjä viivoja.
Viivat piirretään rajaavan ruudun määritellyn koordinaatin ja mittojen ympärille siten, että piirretyn monikulmion todellinen koko on suurempi kuin leveys ja korkeus.

Katso myös: FillPoly

DrawRect

Katalogi > CXII

DrawRect *x, y, width, height*

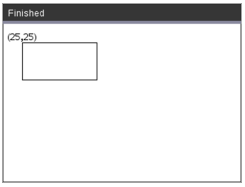
x, y: suorakulmion vasen yläkoordinaatti

width, *height*: suorakulmion leveys ja korkeus (suorakulmio, joka on piirretty alaspäin ja oikealle lähtökoordinaatista)

Huomaa: Viivat piirretään rajaavan ruudun määritellyn 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

Katalogi > CXII

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*

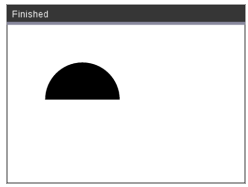
x, y: rajaavan suorakulmion vasen yläkoordinaatti

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

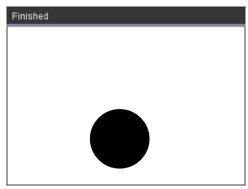
FillCircle *x, y, radius*

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

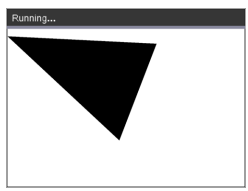
FillPoly *xlist, ylist*

tai

FillPoly *x1, y1, x2, y2, x3, y3...xn, yn*

Huomaa: Viivan ja värin määrittää komennot SetColor ja SetPen

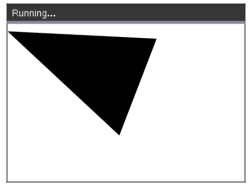
```
xlist:={0,200,150,0}
ylist:={10,20,150,10}
FillPoly xlist,ylist
```



FillPoly

Katalogi > CXII

```
FillPoly 0,10,200,20,150,150,0,10
```



FillRect

Katalogi > CXII

FillRect *x, y, width, height*

x, *y*: suorakulmion vasen yläkoordinaatti

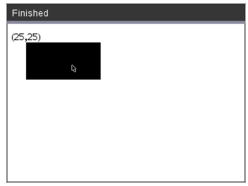
width, *height*: suorakulmion leveys ja korkeus

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

```
FillRect 25,25,100,50
```



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

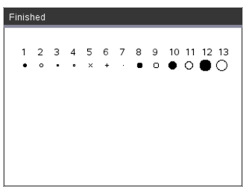
11 – suurempi piste, tyhjä

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

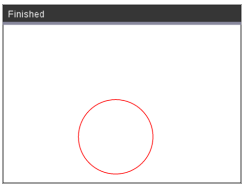
SetColor

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

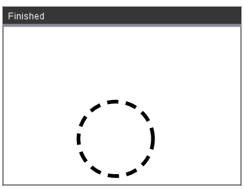
paksuus, tyyli

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.

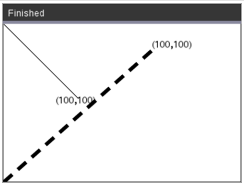
Jos piirretyn objektin osa on ikkunan ulkopuolella, syöttö leikataan (ei näytetä) eikä virheilmoitusta näytetä.

```
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 xmin on suurempi tai yhtä suuri kuin xmax tai ymin on suurempi tai yhtä suuri kuin ymax, 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™ 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 41, ja **isVoid()**, sivu 79.

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 [ctrl] [␣].

Tyhjiä elementtejä sisältävät laskutoimitukset

Suurin osa tyhjän syötteen sisältävistä laskutoimituksista tuottaa tyhjän tuloksen. Katso erikoistapaukset jäljempänä.

$\lvert _ \rvert$	_
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

sum({2,_,3,5,6.6})	16.6
median({1,2,_,_,_,3})	2
cumulativeSum({1,2,_,4,5})	{1,3,_,7,12}
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} → *list1*	{5,4,3,_,1}
{5,4,3,2,1} → *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} → *list1*	{1,2,3,_,5}
{1,2,3,4,5} → *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.}

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.}
stat.XReg	{_,_,4.,5.}
stat.YReg	{_,_,5.,6.6}
stat.FreqReg	{_,_,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.}

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	**pi**
θ	**theta**
∞	**infinity**
$\leq$	**<=**
$\geq$	**>=**
$\neq$	**/=**
$\Rightarrow$ (looginen seuraus)	=>
$\Leftrightarrow$ (looginen kaksoisseuraus, XNOR)	<=>
→ (tallennusoperaattori)	**=:**
| | (itseisarvo)	**abs(...)**
$\sqrt{}$**()**	**sqrt(...)**
Σ() (yhteenlaskumalli)	**sumSeq(...)**
Π() (tulon malli)	**prodSeq(...)**
sin^{-1}(), cos^{-1}(), ...	**arcsin(...), arccos(...)**, ...
Δ**List()**	**deltaList(...)**

Tietokoneen näppäimistöltä

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™ -oppimisteknologiassa. Luvut, muuttujat ja funktiot syötetään yksinkertaisena ja suoraviivaisena sekvenssinä. EOS™-ohjelmisto sieventää lausekkeet ja yhtälöt käyttäen sulkuryhmityksiä sekä alla kuvattuja prioriteetteja.

Laskemisjärjestys

Taso	Operaattori
1	Kaarisulkeet ( ), hakasulkeet [ ], aaltosulkeet { }
2	Epäsuora operaattori (#)
3	Funktioiden komennot
4	Jälkioperaattorit: astetta-minuuttia-sekuntia (°,',"), kertoma (!), prosentti (%), radiaani ($\times^{r}$), alaindeksi ([ ]), transponointi (^{T})
5	Potenssiin korotus, potenssioperaattori (^)
6	Negaatio ($^{-}$)
7	Merkkijonojen ketjutus (&)
8	Kertolasku (•), jakolasku (/)
9	Yhteenlasku (+), vähennyslasku (-)
10	Yhtäläisyyssuhteet: 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™ -ohjelmistossa, ohjelmisto tulkitsee muuttujan nimen, jonka perässä on sulkeissa oleva lauseke, funktion komennoksi eikä implisiittiseksi kertolaskuksi. Esimerkiksi a(b+c) on funktio a, jota sievennetään lausekkeella b+c. Kun haluat kertoa lausekkeen b+c muuttujalla a, käytä eksplisiittistä kertolaskua: $a*$(b+c).

Epäsuora operaattori

Epäsuora operaattori (#) muuttaa merkkijonon muuttujaksi tai funktion nimeksi. Esimerkiksi lausekkeesta #("x"&"y"&"z") luodaan muuttujanimi xyz. Epäsuora operaattori sallii myös muuttujien luomisen ja muokkaamisen ohjelman sisällä. Esimerkiksi, jos 10→r ja "r"→s1, niin #s1=10.

Jälkioperaattorit

Jälkioperaattorit ovat operaattoreita, jotka tulevat suoraan argumentin perään, kuten 5!, 25% tai 60°15' 45". Argumentit, joiden perässä on jälkioperaattori, sievennetään neljännellä prioriteettitasolla. Esimerkiksi lausekkeesta 4^3! sievennetään ensimmäiseksi 3! Vastauksesta, 6, tulee luvun 4 eksponentti, josta saadaan 4096.

Eksponenttilausekkeet

Eksponenttilausekkeet (^) ja potenssiin korotus elementti elementiltä (.^) sievennetään oikealta vasemmalle. Esimerkiksi lauseke 2^3^2 sievennetään samalla tavalla kuin 2^(3^2), josta saadaan 512. Tämä on eri asia kuin (2^3)^2, josta saadaan 64.

Negaatio

Negatiivinen luku syötetään painamalla ensin näppäintä [(−)] ja syöttämällä luku sen perään. Jälkioperaattorit ja eksponenttilausekkeet suoritetaan ennen negaatiota. 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

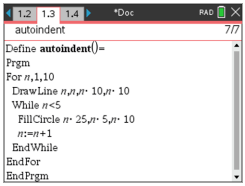
Automaattinen sisennys Ohjelmointieditorissa

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 lohkokomennoissa.



Kopioidut ja liitetyt koodifragmentit säilyttävät alkuperäisen sisennyksen.

Ohjelmiston aikaisemmassa versiossa luodun ohjelman avaaminen säilyttää alkuperäisen sisennyksen.

TI-Basicin virheviestien parannukset

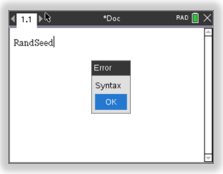
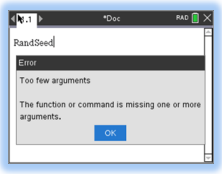
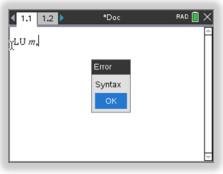
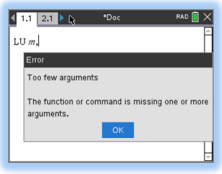
Virheet

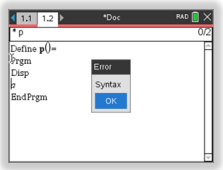
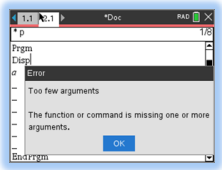
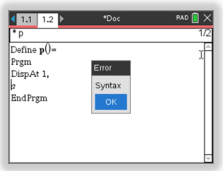
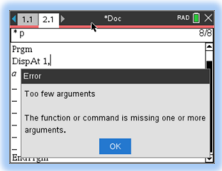
Virhetila	Uusi viesti
Ehtokohteen virhe (**If/While**)	Ehdollista lausuntoa ei voitu ratkaista arvolla **TRUE** tai **FALSE** **HUOMAA:** Koska muutos on sijoitettu riville virheen kanssa, ei ole enää tarpeen tarkentaa, onko virhe ”**If**”- vai ”**While**”-kohde.
Puuttuva **EndIf**	Odotettiin kohdetta **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

Virhetila	Uusi viesti
Puuttuva **EndTry**	Odotettiin kohdetta **EndTry**, mutta löydettiin 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**

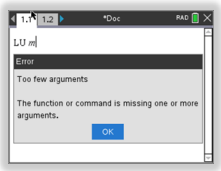
Syntaksivirheet

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. Ne voidaan kirjoittaa manuaalisesti tai valita **Vakiot**-luettelosta **Utilities> Yksikkömuunnokset** (Kannettava laite: Paina 📖 **3**).

Vakio	Nimi	Arvo
_c	valon nopeus	299792458 _m/_s
_Cc	Coulombin vakio	8987551787.3682 _m/_F
_Fc	Faradayn vakio	96485.33289 _coul/_mol
_g	Painovoiman kiihtyvyys	9.80665 _m/_s^2
_Gc	Gravitaatiovakio	6.67408E-11 _m^3/_kg/_s^2
_h	Planckin vakio	6.626070040E-34 _J _s
_k	Boltzmannin vakio	1.38064852E-23 _J/_°K
_μ0	Tyhjiön permeabiliteetti	1.2566370614359E-6 _N/_A^2
_μb	Bohr magnetoni	9.274009994E-24 _J _m^2/_Wb
_Me	Elektronin lepomassa	9.10938356E-31 _kg
_Mμ	Myonin massa	1.883531594E-28 _kg
_Mn	Neutronin lepomassa	1.674927471E-27 _kg
_Mp	Protonin lepomassa	1.672621898E-27 _kg
_Na	Avogadron vakio	6.022140857E23 /_mol
_q	Elektronin varaus	1.6021766208E-19 _coul
_Rb	Bohrin säde	5.2917721067E-11 _m
_Rc	Moolinen kaasuvakio	8.3144598 _J/_mol/_°K
_Rdb	Rydbergin vakio	10973731.568508/_m
_Re	Elektronin säde	2.8179403227E-15 _m
_u	Atomimassa	1.660539040E-27 _kg
_Vm	moolinen tilavuus	2.2413962E-2 _m^3/_mol
_ε0	Tyhjiön permittiivisyys (vakio)	8.8541878176204E-12 _F/_m
_σ	Stefan-Boltzmannin vakio	5.670367E-8 _W/_m^2/_°K^4
_ϕ0	Magneettivuon kvantti	2.067833831E-15 _Wb

Virhekoodit ja viestit

Kun tapahtuu virhe, virhekoodi määritetään muuttujaan *errCode*. Käyttäjän määrittämillä ohjelmilla ja funktioilla voidaan tutkia *errCode*-muuttujaa virheen syyn määrittämiseksi. Esimerkki muuttujan *errCode* käytöstä on esitetty Esimerkissä 2 **Try**-komennon kohdalla, sivu 166.

Huomaa: Jotkin virhetilanteet koskevat vain *TI-Nspire™ CAS -tuotteita ja toiset taas vain TI-Nspire™-tuotteita.*

Virhekoodi	Kuvaus
10	Funktio ei laskenut arvoa.
20	Kokeen tulos ei ollut TOSI eikä EPÄTOSI. Määrittämättömiä muuttujia ei yleensä voi verrata. Esimerkiksi testi If a<b aiheuttaa virheen, jos joko a tai b on määrittämätön suoritettaessa If-lauseketta.
30	Argumentti ei voi olla kansion nimi.
40	Argumenttivirhe
50	Argumentin yhteensopivuusvirhe Kahden tai useamman argumentin on oltava samaa tyyppiä.
60	Argumentin on oltava Boolen lauseke tai kokonaisluku.
70	Argumentin on oltava desimaaliluku.
90	Argumentin on oltava lista.
100	Argumentin on oltava matriisi.
130	Argumentin on oltava merkkijono.
140	Argumentin on oltava muuttujan nimi. Tarkista, että nimi: • ei ala numerolla • ei sisällä välilyöntejä tai erikoismerkkejä • ei käytä alaviivaa tai pistettä väärällä tavalla • ei ylitä pituusrajoituksia Lisätietoja löytyy ohjekirjan Laskin-sovellusta käsittelevästä osasta.
160	Argumentin on oltava lauseke.
165	Paristovirta ei riitä lähettämiseen tai vastaanottamiseen. Asenna uudet paristot ennen lähettämistä tai vastaanottoa.
170	Raja Alarajan on oltava ylärajaa pienempi hakuväliä määritettäessä.
180	Keskeytys Näppäintä [esc] tai [on] on painettu pitkän laskutoimituksen tai ohjelman suorituksen aikana.

Virhekoodi	Kuvaus
190	Kehämääritys
	Tämä viesti tulee näkyviin muistin loppumisen välttämiseksi muuttujien arvojen äärettömän korvauksen aikana sievennettäessä lausekkeita. Virheen aiheuttaa esimerkiksi lauseke a+1->a, jossa a on määrittämätön muuttuja.
200	Määrittelyalueen lauseke ei kelpaa.
	Tämän virheilmoituksen voi aiheuttaa esimerkiksi lauseke solve(3x^2-4=0,x) | x<0 tai x>5, koska ehto on erotettu operaattorilla “or” eikä operaattorilla “and”.
210	Datatyyppi ei kelpaa.
	Argumentin datatyyppi on väärä.
220	Riippuvainen raja-arvo
230	Dimensio
	Lista tai matriisi-indeksi ei kelpaa. Jos esimerkiksi lista {1,2,3,4} on tallennettu L1:een, tällöin L1[5] aiheuttaa dimensiovirheen, koska L1 sisältää vain neljä elementtiä.
235	Dimensiovirhe Listoissa ei ole riittävästi elementtejä.
240	Dimensioiden yhteensopivuusvirhe
	Kahdella tai useammalla argumentilla on oltava sama dimensio. Esimerkiksi lauseke [1,2]+[1,2,3] aiheuttaa dimensioiden yhteensopivuusvirheen, koska matriisit sisältävät eri määrän elementtejä.
250	Jako nollalla
260	Määrittelyjoukkovirhe
	Argumentin on kuuluttava tiettyyn määrittelyjoukkoon. Esimerkiksi **rand(0)** ei kelpaa.
270	Muuttujan nimi jo käytössä
280	Else ja ElseIf eivät kelpaa lohkon If..EndIf ulkopuolella
290	EndTry-komennosta puuttuu sopiva Else-lauseke
295	Liian monta iteraatiota
300	Odotettavissa 2- tai 3-elementtinen lista tai matriisi
310	Funktion **nSolve** ensimmäisen argumentin on oltava yhden muuttujan yhtälö. Se ei voi sisältää muuta arvotonta muuttujaa kuin käsittelyn kohteena olevan muuttujan.
320	solve- tai cSolve-funktion ensimmäisen argumentin on oltava yhtälö tai epäyhtälö.
	Esimerkiksi 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.

Virhekoodi	Kuvaus
390	Epäkelpo tehtävä
400	Epäkelpo tehtävän arvo
410	Virheellinen komento
430	Virheellinen nykyisille tila-asetuksille
435	Virheellinen arvaus
440	Virheellinen kertomerkitön kertolasku
	Esimerkiksi lauseke x(x+1) ei kelpaa, mutta lausekkeen x*(x+1) syntaksi on oikein. Tarkoituksena on välttää sekaannusta kertomerkittömien kertolaskujen ja funktioiden komentojen kanssa.
450	Virhe funktiossa tai nykyisessä lausekkeessa
	Vain tietyt komennot kelpaavat käyttäjän määrittämässä funktiossa.
490	Virhe Try..EndTry-lohkossa
510	Virheellinen lista tai matriisi
550	Virheellinen ulkopuolinen funktio tai ohjelma
	Jotkin komennot eivät kelpaa funktion tai ohjelman ulkopuolella. Esimerkiksi **Local**-komentoa ei voi käyttää, ellei se ole funktion tai ohjelman sisällä.
560	Virhe lohkojen Loop..EndLoop, For..EndFor tai While..EndWhile ulkopuolella
	Esimerkiksi Exit-komento kelpaa vain näiden silmukkalohkojen sisällä.
565	Virheellinen ulkopuolinen ohjelma
570	Polkunimi ei kelpaa
	Esimerkiksi \muutt ei kelpaa.
575	Virheellinen polaarinen kompleksiluku
580	Virheellinen ohjelmaviittaus
	Ohjelmiin ei voi viitata funktioiden tai lausekkeiden sisällä, kuten 1+p(x), jossa p on ohjelma.
600	Kelpaamaton taulukko
605	Yksiköitä käytetty väärin
610	Virheellinen muuttujan nimi Local-lausekkeessa
620	Virheellinen muuttujan tai funktion nimi
630	Virheellinen muuttujan viittaus
640	Virheellinen vektorin syntaksi
650	Linkkisiirto
	Kahden laitteen välistä siirtoa ei ole suoritettu loppuun. Tarkista, että kaapeli on kytketty kunnolla kumpaankin laitteeseen.
665	Matriisia ei voi diagonalisoida
670	Muisti vähissä

Virhekoodi	Kuvaus
	1. Poista tietoja tästä asiakirjasta
	2. Tallenna ja sulje tämä asiakirja
	Jos toimenpiteet 1 ja 2 eivät auta, poista ja asenna paristot uudelleen
672	Resource exhaustion (Resurssit lopussa)
673	Resource exhaustion (Resurssit lopussa)
680	Puuttuva (
690	Puuttuva)
700	Puuttuva "
710	Puuttuva]
720	Puuttuva }
730	Syntaksista puuttuu lohkon alku tai loppu
740	Then puuttuu lohkosta If..EndIf
750	Nimi ei ole funktio eikä ohjelma
765	Yhtään funktiota ei ole valittu
780	Ratkaisua ei löydy
800	Ei-reaalinen tulos
	Esimerkiksi, jos ohjelmisto on Real-asetuksessa, $\sqrt{(-1)}$ ei kelpaa.
	Jotta voit saada kompleksilukuvastauksia, muuta reaali- tai kompleksitilan asetus valintaan RECTANGULAR (SUORAKULMA) tai POLAR (POLAARINEN).
830	Laskualueen ylitys
850	Ohjelmaa ei löydy
	Toisen ohjelman sisällä olevaa ohjelman viittausta ei löytynyt määritetystä polusta ohjelman suorittamisen aikana.
855	Rand-tyyppiset funktiot eivät ole sallittuja kuvaajan piirtämisen aikana
860	Liian syvä rekursio
870	Varattu nimi tai järjestelmän muuttuja
900	Argumenttivirhe
	Mediaani-mediaani-mallia ei voitu käyttää datasarjaan.
910	Syntaksivirhe
920	Tekstiä ei löydy
930	Liian vähän argumentteja
	Funktiosta tai komennosta puuttuu yksi tai useampia argumentteja.
940	Liian monta argumenttia

Virhekoodi	Kuvaus
	Lauseke tai yhtälö sisältää liian monta argumenttia, eikä sitä voi ratkaista.
950	Liian monta alaindeksiä
955	Liian monta määrittämätöntä muuttujaa
960	Muuttujaa ei ole määritetty Muuttujalle ei ole määritetty arvoa. Käytä jotakin seuraavista komennoista: • sto → • := • **Define** määrittääksesi muuttujille arvot.
965	Käyttöjärjestelmälle ei ole lisenssiä
970	Muuttuja on käytössä, joten viittaukset tai muutokset eivät ole sallittuja
980	Muuttuja on suojattu
990	Virheellinen muuttujan nimi Varmista, että nimi ei ylitä pituusrajoituksia
1000	Ikkunamuuttujien määrittelyalue
1010	Zoomaus
1020	Sisäinen virhe
1030	Suojatun muistin rikkomus
1040	Toimintoa ei tueta Tämä toiminto vaatii tietokonealgebrajärjestelmän Kokeile TI-Nspire™ CAS -järjestelmää.
1045	Operaattoria ei tueta. Tämä operaattori vaatii tietokonealgebrajärjestelmän. Kokeile TI-Nspire™ CAS -järjestelmää.
1050	Toimintoa ei tueta. Tämä operaattori vaatii tietokonealgebrajärjestelmän. Kokeile TI-Nspire™ CAS -järjestelmää.
1060	Syötetyn argumentin on oltava numeerinen. Vain numeerisia arvoja sisältävät syötteet sallitaan.
1070	Trig-funktion argumentti on liian suuri täsmälliseen sievennykseen
1080	Ans-muuttujan käyttöä ei tueta. Tämä sovellus ei tue Ans-muuttujan käyttöä.
1090	Funktiota ei ole määritetty. Käytä jotakin seuraavista komennoista: • **Define** • := • sto → määrittääksesi funktion.
1100	Ei-reaalinen laskutoimitus Esimerkiksi, jos ohjelmisto on Real-asetuksessa, $\sqrt{}$(-1) ei kelpaa.

Virhekoodi	Kuvaus
	Jotta voit saada kompleksilukuvastauksia, muuta reaali- tai kompleksitilan asetus valintaan RECTANGULAR (SUORAKULMA) tai POLAR (POLAARINEN).
1110	Virheelliset rajat
1120	Ei etumerkin muutosta
1130	Argumentti ei voi olla lista tai matriisi
1140	Argumenttivirhe 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 *xxx**yyy*, jossa: • *xxx*-osassa voi olla 1-16 merkkiä. • *yyy*-osassa voi olla 1–15 merkkiä. Lisätietoja löydät ohjekirjan osasta Kirjasto.
1170	Kirjaston polkunimeä on käytetty väärin • Polkunimelle ei voi määrittää arvoa komennon **Define**, **:=** tai sto → avulla. • Polkunimeä ei voi määrittää paikalliseksi muuttujaksi tai käyttää funktion tai ohjelman parametrina.
1180	Virheellinen kirjastomuuttujan nimi. Tarkista, että nimi: • ei sisällä pistettä • ei ala alaviivalla • ei ylitä 15 merkkiä Lisätietoja löydät ohjekirjan osasta Kirjasto.
1190	Kirjastodokumenttia ei löydy: • Tarkista, että kirjasto on MyLib-kansiossa. • Päivitä kirjastot. Lisätietoja löydät ohjekirjan osasta Kirjasto.
1200	Kirjastomuuttujaa ei löydy: • Tarkista, että kirjaston ensimmäisessä tehtävässä on kirjastomuuttuja. • Tarkista, että kirjastomuuttujaksi on määritetty LibPub tai LibPriv. • Päivitä kirjastot. Lisätietoja löydät ohjekirjan osasta Kirjasto.

Virhekoodi	Kuvaus
1210	Virheellinen kirjaston pikavalintanimi. Tarkista, että nimi: • ei sisällä pistettä • ei ala alaviivalla • ei ylitä 16 merkkiä • ei ole varattu nimi Lisätietoja löydät ohjekirjan osasta Kirjasto.
1220	Määrittelyjoukkovirhe: Funktiot tangentLine ja normalLine tukevat vain reaaliarvoisia funktioita.
1230	Määrittelyjoukkovirhe. Trigonometristen muunnosten operaattoreita ei tueta aste- eikä graadikulmatilassa.
1250	Argumenttivirhe Käytä lineeariyhtälöryhmää. Esimerkki kahden lineeariyhtälön ryhmästä, jossa on muuttujat x ja y: 3x+7y=5 2y-5x=-1
1260	Argumenttivirhe: **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	Argumenttivirhe Derivaatan asteen on oltava 1 tai 2.
1280	Argumenttivirhe Käytä yhden muuttujan sisältämää laajennetussa muodossa olevaa polynomia.
1290	Argumenttivirhe Käytä yhden muuttujan sisältämää polynomia.
1300	Argumenttivirhe Polynomin kertoimien tuloksena on oltava numeerinen arvo.
1310	Argumenttivirhe: Yhtä tai useampaa funktion argumenttia ei voi sieventää.
1380	Argumenttivirhe: 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(), sivu 175.

Varoituskoodi	Viesti
10000	Laskutoimitus voi antaa vääriä ratkaisuja.
10001	Yhtälön derivointi voi antaa väärän yhtälön.
10002	Kyseenalainen ratkaisu
10003	Kyseenalainen tarkkuus
10004	Laskutoimituksessa voi hävitä ratkaisuja.
10005	cSolve voi määrittää lisää nollakohtia.
10006	Solve voi määrittää lisää nollia.
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)
10008	Tuloksen määrittelyjoukko voi olla pienempi kuin syötteen määrittelyjoukko.
10009	Tuloksen määrittelyjoukko voi olla suurempi kuin syötteen määrittelyjoukko.
10012	Ei-reaalinen laskutoimitus
10013	∞^0 tai undef^0 korvattu arvolla 1
10014	undef^0 korvattu arvolla 1
10015	1^∞ tai 1^undef korvattu arvolla1
10016	1^undef korvattu arvolla 1
10017	Laskualueen ylitys korvattu arvolla ∞ tai ⁻∞
10018	Laskutoimitus vaatii ja laskee 64-bittisen arvon.
10019	Muisti täynnä, sievennystä ei mahdollisesti suoriteta loppuun.
10020	Trig-funktion argumentti on liian suuri täsmälliseen sievennykseen.
10021	Syöte sisältää määrittämättömän parametrin.

Varoituskoodi	Viesti
	Tulos ei ehkä ole voimassa kaikille mahdollisille parametriarvoille.
10022	Ratkaisu on mahdollinen, kun määritetään oikeat ala- ja ylärajat.
10023	Skalaari on kerrottu identtisellä matriisilla.
10024	Tulos saatu käyttämällä likimääräistä aritmetiikkaa.
10025	Ekvivalenssia ei voida varmistaa EXACT-tilassa.
10026	Ehto saatetaan ohittaa. Määrittele ehto muodossa "\" 'Muuttuja MatemaattinenTestiSymboli Vakio' tai näiden muotojen yhdistelmänä, esimerkiksi 'x<3 ja x>-12'

Yleistä

Online-tuki

education.ti.com/eguide

Valitse maasi, niin näet lisää tuotetietoja.

Ota yhteyttä TI-tukeen

education.ti.com/ti-cares

Valitse maasi, niin näet teknisiä tietoja ja muita tukiresursseja.

Huolto- ja takuutiedot

education.ti.com/warranty

Valitse maasi saadaksesi lisätietoja takuun kestosta ja ehdoista tai tuotetuesta.

Rajoitettu takuu. Tämä takuu ei vaikuta lainmukaisiin oikeuksiisi

Indeksi

–

–, vähennyslasku[*] 183

!

!, kertoma 194

"

", sekuntimuoto 201

#

#, epäsuora operaattori 199, 227

%

%, prosentti 189

&

&, liitä 194

*****

*, kertolasku 184

,

, minuuttimuoto 201

.

.-, piste-erotus 188
.*, pistetulo 188
./, pisteosamäärä 188
.^, pistepotenssi 189
.+, pistesumma 187

:

:=, määritä 205

^

$^{-1}$, käänteisluku 203
^, potenssi 186

|

|, (rajoittava operaattori) 203

+

+, yhteenlasku 183

/

/, jakolasku[*] 185

≠

≠, ei yhtä kuin[*] 190

=

=, on yhtä kuin 190

>

>, suurempi kuin 192

∏

∏, tulo[*] 196

∑

∑( ), summa[*] 197
∑Int( ) 197
∑Prn( ) 198

√

√, neliöjuuri[*] 196

∫

∫, integraali[*] 195

≤

≤, pienempi tai yhtä suuri kuin 191

≥

≥, suurempi tai yhtä suuri kuin 192

▸

▸, muunna graadikulmaksi[Grad] ... 71
▸. asteen regressio, QuartReg 123
▸approxFraction( ) 13
▸Cylind, näytä vektorina sylinterikoordinaatistossa [Cylind] 36
▸DD, näytä desimaalikulmana[DD] .. 37
▸Desimaali, näytä tulos desimaalilukuna[Desimaali] 38
▸DMS, näytä asteina/minuutteina/sekunteina[DMS] 45
▸Kantaluku10, näytä desimaalikokonaislukuna [Kantaluku10] 18
▸Kantaluku16, näytä heksadesimaalilukuna [Kantaluku16] 19
▸Kantaluku2, näytä binaarisena [Kantaluku2] 16
▸Polaarinen, näytä polaarisena vektorina[Polaarinen] 117
▸Rad, muunna radiaanikulmaan ... 125
▸Rect, näytä suorakulmavektorina . 128
▸Sphere, näytä pallonmuotoisena vektorina[Sphere] 153

⇒

⇒, looginen seuraus[*] 193, 225

→

→, tallenna 205

⇔

⇔, looginen kaksoisseuraus[*] 193

©

©, kommentti 206

°

°, astekulmamuoto[*] 201
°, astetta/minuuttia/sekuntia[*] ... 201

0

0b, binaarinen indikaattori 206
0h, heksadesimaalinen indikaattori . 206

1

10^( ), kymmenen potenssi 202

2

2 otoksen F-testi 59
2. asteen regressio, QuadReg 122

3

3. asteen regressio, CubicReg 34

A

abs( ), itseisarvo 7
alaraja, floor( ) 56
alimatriisi, subMat( ) 157, 159
amortTbl( ), lyhennystaulukko 7, 16
and, Boolean operator 8
angle( ), kulma 9
ANOVA, yksisuuntainen varianssianalyysi 9
ANOVA2way, kaksisuuntainen varianssianalyysi 10
ans, viimeinen vastaus 12
approx( ), likimääräinen 12
approxRational( ) 13
arccos() 13
arccosh() 13
arccot() 13
arccoth() 14
arccsc() 14
arccsch() 14
arcsec() 14
arcsech() 14
arcsin() 14
arcsinh() 14
arctan() 14
arctanh() 14
arkuskosini, $\cos^{-1}$( ) 28
arkussini, $\sin^{-1}$( ) 149

arkustangentti, $\tan^{-1}$( ) 160
aseta
tila, setMode( ) 143
asetukset, hae nykyiset 68
aste/minuutti/sekunti-muoto 201
astekulmamuoto, - 201
augment( ), lisää/ketjuta 15
avgRC( ), keskimääräinen
muutosnopeus 15

B

binaarinen
indikaattori, 0b 206
näytä, 4Kantaluku2 16
binomCdf( ) 19, 77
binomPdf( ) 20
Boolean operators
and 8
Boolen operaattorit
⇒ 193, 225
⇔ 193
ei 109
ei-ja 103
eikä 107
tai 113
xor 176

C

Cdf( ) 54
ceiling( ), yläraja 20
centralDiff( ) 21
char( ), merkkijono 21
$\chi^2$2way 21
clearAZ 23
ClrErr, poista virhe 24
colAugment 24
colDim( ), matriisin sarakemitat 25
colNorm( ), matriisin sarakenormi 25
conj( ), liittokompleksiluku 25
constructMat( ), luo matriisi 25
corrMat( ), korrelaatiomatriisi 26
$\cos^{-1}$, arkuskosini 28
cos( ), kosini 27
$\cosh^{-1}$( ), hyperbolinen arkuskosini 29
cosh( ), hyperbolinen kosini 29
$\cot^{-1}$( ), arkuskotangentti 30
cot( ), kotangentti 30
$\coth^{-1}$( ), hyperbolinen
arkuskotangentti 31
coth( ), hyperbolinen kotangentti 30
count( ), laske listan kohtien
lukumäärä 31
countif( ), laske listan yksiköiden
lukumäärä ehdollisesti 32
cPolyRoots() 32
crossP( ), ristitulo 33
$\csc^{-1}$( ), käänteiskosekantti 33
csc( ), kosekantti 33
csch^{-1}( ), käänteinen hyperbolinen
kosekantti 34
csch( ), hyperbolinen kosekantti 34
CubicReg, 3. asteen regressio 34
cumulativeSum( ), kumulatiivinen
summa 35
cycle, Cycle 36
Cycle, cycle 36

D

d( ), ensimmäinen derivaatta 194
dbd( ), päivämäärien väliset päivät 37
Define (Määritä) 38
Define LibPriv (Määritä LibPriv) 39
Define LibPub (Määritä LibPub) 39
Define, määritä 38
deltaList() 40
DelVar, poista muuttuja 40
delVoid( ), poista tyhjät elementit 41
derivaatat
ensimmäinen derivaatta, d ( ) 194
numeerinen derivaatta, nDeriv( ) 106
numeerinen derivaatta,
nDerivative( ) 105
desimaali
kokonaislukunäyttö,
4Kantaluku10 18
kulmanäyttö, ▸DD 37
det( ), matriisin determinantti 41

diag( ), diagonaalimatriisi 42
dim( ), mitta 42
Disp, näytä tiedot 43, 141
DispAt 43
dotP( ), pistetulo 45

E

e eksponentti
    malli 2
e korotettuna potenssiin, e^( ) 46, 51
E, eksponentti 199
e^( ), e korotettuna potenssiin 46
efektiivinen korko, eff( ) 46
eff), muunna nimelliskorko
        efektiiviseksi koroksi 46
ei-ja, Boolen operaattori 103
ei yhtä kuin, ≠ 190
ei, Boolen operaattori 109
eigVc( ), ominaisvektori 47
eigVl( ), ominaisarvo 47
eikä, Boolen operaattori 107
ekskluusio "|"-operaattorilla 203
eksponentiaalinen regressio, ExpReg 52
eksponentit
    malli 1
eksponentti, E 199
else if, ElseIf 48
ElseIf, else if 48
end
    for, EndFor 56
    funktio, EndFunc 60
    ohjelma, EndPrgm 119
    silmukka, EndLoop 93
    try, EndTry 166
    while, EndWhile 176
end while, EndWhile 176
EndTry, kokeilun loppu 166
EndWhile, end while 176
ensimmäinen derivaatta
    malli 5
EOS (yhtälökäyttöjärjestelmä) 226
epäsuora operaattori (#) 227
epäsuora operaattori, # 199
etumerkki, sign( ) 146
euler( ), Euler function 49
Exit, lopeta 51
exp( ), e korotettuna potenssiin 51
expr( ), merkkijono lausekkeeksi 52
ExpReg, eksponentiaalinen regressio 52

F

factor( ), tekijä 54
Fill, matriisin täyttö 55
FiveNumSummary 55
floor( ), alaraja 56
For 56
for, For 56
For, for 56
format( ), merkkijonon muoto 57
fpart( ), funktion osa 58
freqTable( ) 58
frequency( ) 59
Frobeniusin normi, norm( ) 108
Func, funktio 60
Func, ohjelmafunktio 60
funktion loppu, EndFunc 60
funktiot
    käyttäjän määrittämät 38
    ohjelmafunktio, Func 60
    osa, fpart( ) 58
funktiot ja muuttujat
    kopioiminen 26

G

g, graadit 200
gcd( ), suurin yhteinen jakaja 61
geomCdf( ) 61
geomPdf( ) 62
Get 62, 217
getDenom( ), hae/laske nimittäjä 63
getKey() 63
getLangInfo( ), hae/anna kielitiedot 67
getLockInfo( ), testaa muuttujan tai
        muuttujaryhmän
        lukitustilan 67
getMode( ), hae tila-asetukset 68
getNum( ), hae/laske luku 69

GetStr 69
getType( ), get type of variable 69
getVarInfo( ), hae/laske muuttujien tiedot 70
Goto, siirry 71
graadimuoto, g 200

H

hae/laske
luku, getNum( ) 69
muuttujien tiedot, getVarInfo( ) 67, 70
nimittäjä, getDenom( ) 63
heksadesimaali
näytä, 4Kantaluku16 19
heksadesimaalinen
indikaattori, 0h 206
hyperbolinen
areatangentti, $\tanh^{-1}$( ) 162
arkuskosini, $\cosh^{-1}$( ) 29
arkussini, $\sinh^{-1}$( ) 150
kosini, cosh( ) 29
sini, sinh( ) 149
tangentti, tanh( ) 161

I

identity(), identtinen matriisi 71
identtinen matriisi, identity() 71
If, jos 72
ifFn( ) 73
imag(), imaginaarinen osa 74
imaginaarinen osa, imag() 74
inString(), merkkijonon sisällä 74
int(), kokonaisluku 75
intDiv(), kokonaisluvun jakaminen .. 75
integraali, ∫ 195
interpolate( ), interpoloi 75
invF( ) 76
invNorm(), käänteinen kumulatiivinen normaalijakauma) 77
invt( ) 78
Invχ^2( ) 76
iPart(), kokonaisosa 78
irr( ), sisäinen palautuksen määrä
sisäinen palautuksen määrä , irr( ) 78
isPrime(), jaottoman testi 78
isVoid( ), tyhjän testi 79
itseisarvo
malli 4

J

jakakaumafunktiot
binomCdf( ) 77
invNorm( ) 77
invt( ) 78
jakaumafunktiot
binomCdf( ) 19
binomPdf( ) 20
Invχ^2( ) 76
normCdf( ) 108
normPdf( ) 109
poissCdf( ) 116
poissPdf( ) 116
tCdf( ) 162
tPdf( ) 165
$\chi^2$2way( ) 21
χ^2Cdf( ) 22
χ^2GOF( ) 22
χ^2Pdf( ) 23
jakolasku, P 185
jaottoman luvun testi, isPrime() 78
jos käytettävissäsi on | 203
jos, EndIf 72
jos, If 72
jäännös, remain() 131

K

kahden muuttujan tulokset, TwoVar 170
kertolasku, * 184
kertoma, ! 194
keskiarvo, mean( ) 95
keskihajonta, stdDev( ) 155-156, 173
keskimmäinen merkkijono, mid( ) .. 98
keskimääräinen muutosnopeus, avgRC( ) 15

kieli
hae kielitiedot 67
kierrä, rotate() 136
kirjasto
luo pikavalinnat objekteihin ... 80
kokonaisluku, int() 75
kokonaisluvun jakaminen, intDiv() .. 75
kokonaisosa, iPart() 78
kombinaatiot, nCr( ) 104
kommentti, © 206
kompleksi
liitto, conj( ) 25
kopioi muuttuja tai funktio, CopyVar 26
korkoerien summa 197
korrelaatiomatriisi, corrMat( ) 26
korvaaminen "|"-operaattorilla 203
kosini, cos( ) 27
kotangentti, cot( ) 30
kulma, angle( ) 9
kumulatiivinen summa,
cumulativeSum( ) 35
kun, when( ) 175
kymmenen potenssi, 10^( ) 202
käyttäjän määrittämät funktiot 38
käyttäjän määrittämät funktiot ja
ohjelmat 39
käänteinen kumulatiivinen
normaalijakauma (invNorm(
) 77
käänteinen, ^⁻¹ 203
käänteisluku, ^⁻¹ 203

L

lajitteleminen
laskeva järjestys, SortD 152
nouseva järjestys, SortA 152
laske listan kohtien lukumäärä,
count( ) 31
laske listan yksiköiden lukumäärä
ehdollisesti, countif( ) 32
laske polynomi, polyEval( ) 117
laske päivämäärien väliset päivät,
dbd( ) 37
lausekkeet
merkkijono lausekkeeksi, expr( ) 52
Lbl, tunnus 79
lcm, pienin yhteinen jaettava 80
left( ), vasen 80
LibPriv 39
LibPub 39
libShortcut( ), luo pikavalinnat
kirjasto-objekteihin 80
liitä, & 194
likimääräinen, approx( ) 12
lineaarinen regressio, LinRegAx 82
lineaarinen regressio, LinRegBx 81, 83
LinRegBx, lineaarinen regressio 81
LinRegMx, lineaarinen regressio 82
LinRegtIntervals, lineaarinen
regressio 83
LinRegtTest 85
linSolve() 86
Δlist( ), listaerotus 87
list▸mat( ), listasta matriisiksi 87
lista, laske kohtien lukumäärä 31
lista, laske yksiköiden lukumäärä
ehdollisesti 32
listasta matriisiksi, list▸mat( ) 87
listat
erotukset listassa, @list( ) 87
erotus, @list( ) 87
keskimmäinen merkkijono, mid(
) 98
kumulatiivinen summa,
cumulativeSum( ) 35
lajittele laskevaan järjestykseen,
SortD 152
lajittele nousevaan
järjestykseen, SortA .. 152
listasta matriisiksi, list▸mat( ) ... 87
lisää/ketjuta, augment( ) 15
maksimi, max( ) 95
matriisi listaksi, mat▸list( ) 94
minimi, min( ) 98
pistetulo, dotP( ) 45
ristitulo, crossP( ) 33
tulo, product( ) 120
tyhjät elementit 223
uusi, newList( ) 106
yhteenlasku, sum( ) 158

lisää/ketjuta, augment( ) 15
ln( ), luonnollinen logaritmi 87
LnReg, logaritminen regressio 88
Local, paikallinen muuttuja 89
Lock, lukitse muuttuja tai muuttujaryhmä 90
Logaritmi
malli 2
logaritminen regressio, LnReg 88
logaritmit 87
Logistic, logistinen regressio 91
LogisticD, logistinen regressio 92
logistinen regressio, Logistic 91
logistinen regressio, LogisticD 92
looginen kaksoisseuraus, ⇔ 193
looginen seuraus, ⇒ 193, 225
Loop, silmukka 93
lopeta, Exit 51
loppu 72
loppu jos, EndIf 72
LU, matriisin ala-ylä-dekomponointi 94
lukujono
kierrä, rotate( ) 136
oikea, right( ) 134
lukujonot
oikea, right( ) 75
luo matriisi, constructMat( ) 25
luonnollinen logaritmi, ln( ) 87
lyhennystaulukko, amortTbl( ) 7, 16

M

maksimi, max( ) 95
mallit
e eksponentti 2
eksponentti 1
ensimmäinen derivaatta 5
itseisarvo 4
Logaritmi 2
matriisi (1 × 2) 4
matriisi (2 × 1) 4
matriisi (2 × 2) 4
matriisi (m × n) 4
murtoluku 1
määrätty integraali 6
n:s juuri 2
neliöjuuri 1
paloittain määritelty funktio (2-osainen) 2
paloittain määritelty funktio (N-osainen) 3
sum (G) 5
toinen derivaatta 6
tulo (P) 5
yhtälöpari (2 yhtälöä) 3
yhtälöryhmä (N yhtälöä) 3
mat▸list( ), matriisi listaksi 94
matriisi (1 × 2)
malli 4
matriisi (2 × 1)
malli 4
matriisi (2 × 2)
malli 4
matriisi (m × n)
malli 4
matriisi listaksi, mat▸list( ) 94
matriisit
ala-ylä-dekomponointi, LU 94
alimatriisi, subMat( ) 157, 159
determinantti, det( ) 41
diagonaali, diag( ) 42
identtinen, identity() 71
kumulatiivinen summa, cumulativeSum( ) 35
listasta matriisiksi, list▸mat( ) 87
lisää/ketjuta, augment( ) 15
maksimi, max( ) 95
matriisi listaksi, mat▸list( ) 94
minimi, min( ) 98
mitta, dim( ) 42
ominaisarvo, eigVl( ) 47
ominaisvektori, eigVc( ) 47
piste-erotus, .- 188
pisteosamäärä, ./ 188
pistepotenssi, .^ 189
pistesumma, .+ 187
pistetulo, .* 188
QR-hajottaminen, QR 121

rivi-echelon-muoto, ref( ) 129
rivien kertominen ja yhteenlasku, mRowAdd ( ) 100
rivien yhteenlasku, rowAdd( ) 138
rivikoko, rowDim( ) 138
rivinormi, rowNorm( ) 138
rivinvaihto, rowSwap( ) 138
rivioperaatio, mRow( ) 100
sarakemitat, colDim( ) 25
sarakenormi, colNorm( ) 25
satunnainen, randMat( ) 126
transponoi, T 159
tulo, product( ) 120
täyttäminen, Fill 55
uusi, newMat( ) 106
yhteenlasku, sum( ) 158
max( ), maksimi 95
mean( ), keskiarvo 95
mediaani-mediaani-suoran regressio, MedMed 96
mediaani, median( ) 96
median( ), mediaani 96
MedMed, mediaani-mediaani-suoran regressio 96
merkit
merkkijono, char( ) 21
numeerinen koodi, ord( ) 114
merkkijono
mitta, dim( ) 42
pituus 42
merkkijono, char( ) 21
merkkijonon muoto, format( ) 57
merkkijonon pituus 42
merkkijonon sisällä, inString() 74
merkkijonot
epäsuora operaattori, # 199
keskimmäinen merkkijono, mid( ) 98
käyttö muuttujanimien luomisessa 227
lauseke merkkijonoksi, string( ) 157
liitä, & 194
merkkijono lausekkeeksi, expr( ) 52
merkkijono, char( ) 21
merkkikoodi, ord( ) 114
muoto, format( ) 57
muotoilu 57
siirrä, shift( ) 145
sisällä, inString() 74
vasen, left( ) 80
mid( ), keskimmäinen merkkijono 98
min( ), minimi 98
minimi, min( ) 98
minuuttimuoto, 201
mirr( ), modifioitu sisäinen korkokanta 99
mitta, dim( ) 42
mod( ), modulo 99
modifioitu sisäinen korkokanta, mirr( ) 99
modulo, mod( ) 99
Moninkertaisen lineaarisen regression t-testi 102
mRow( ), matriisin rivioperaatio 100
mRowAdd( ), matriisin rivien kertominen ja yhteenlasku 100
MultReg 100
MultRegIntervals( ) 101
MultRegTests( ) 102
murtoluvut
malli 1
propFrac 120
muunna
4Grad 71
muuntaminen
▸Rad 125
muutoin, Else 72
muuttuja
nimen luominen merkkijonosta 227
muuttujat
paikallinen, Local 89
poista kaikki yksikirjaimiset 23
poista, DelVar 40
muuttujat ja funktiot
kopioiminen 26
muuttujat, lukitseminen ja vapauttaminen 67, 90, 172
muuttujien ja muuttujaryhmien lukitseminen 90

muuttujien ja muuttujaryhmien
vapauttaminen 172
määrittäminen
julkinen funktio tai ohjelma 39
yksityinen funktio tai ohjelma .. 39
määritä, Define 38
määrätty integraali
malli 6

N

n:s juuri
malli 2
nCr( ), kombinaatiot 104
nDerivative( ), numeerinen
derivaatta 105
negaatio, negatiivisten lukujen
syöttäminen 227
neliöjuuri
malli 1
neliöjuuri, ‡( ) 153, 196
nettonykyarvo, npv( ) 110
newList( ), uusi lista 106
newMat( ), uusi matriisi 106
nfMax( ), numeerisen funktion
maksimi 106
nfMin( ), numeerisen funktion
minimi 106
nimelliskorko, nom( ) 107
nInt( ), numeerinen integraali 107
nom), muuta efektiivinen korko
nimelliskoroksi 107
norm( ), Frobeniusin normi 108
normaalijakauman todennäköisyys,
normCdf( ) 108
normCdf( ) 108
normPdf( ) 109
nPr( ), permutaatiot 110
npv( ), nettonykyarvo 110
nSolve( ), numeerinen ratkaisu 111
numeerinen
derivaatta, nDeriv( ) 106
derivaatta, nDerivative( ) 105
integraali, nInt( ) 107
ratkaisu, nSolve( ) 111
näppäimistön pikavalinnat 225
näyttö
asteina/minuutteina/sekun
teina, ▸DMS 45
näyttö pallonmuotoisena vektorina,
▸Sphere 153
näyttö vektorina
sylinterikoordinaatistossa,
▸Cylind 36
näytä
asteina/minuutteina/sekunteina,
▸DMS 45
binaarisena, 4Kantaluku2 16
desimaalikokonaislukuna,
4Kantaluku10 18
desimaalikulmana, ▸DD 37
heksadesimaalilukuna,
4Kantaluku16 19
pallonnmuotoinen vektori,
▸Sphere 153
polaarinen vektori, 4Polaarinen 117
suorakulmavektorina, ▸Rect ... 128
vektorina
sylinterikoordinaatistos
sa, 4Cylind 36
näytä tiedot, Disp 43, 141

O

objektit
luo pikavalinnat kirjastoon 80
Odota-komento 174
ohita virhe, PassErr 115
ohjelmat
julkisen kirjaston määrittäminen 39
yksityisen kirjaston
määrittäminen 39
ohjelmat ja ohjelmointi
kokeilun loppu, EndTry 166
lopeta ohjelma, EndPrgm 119
näytä I/O-näyttö, Disp 43, 141
poista virhe, ClrErr 24
try, Try 166
ohjelmointi
määritä ohjelma, Prgm 119
näytä tiedot, Disp 43, 141
ohita virhe, PassErr 115

oikea, right() 75, 134
ominaisarvo, eigVl( ) 47
ominaisvektori, eigVc( ) 47
on yhtä kuin, = 190
OneVar, yhden muuttujan tilastot 112
operaattorit
sievennysjärjestys 226
ord( ), numeerinen merkkikoodi 114

P

P▸Rx( ), x:n suorakulmakoordinaatti 114
P▸Ry( ), y:n suorakulmakoordinaatti 115
paikallinen muuttuja, Local 89
paikallinen, Local 89
Palaa, return 134
paloittain määritelty funktio (2-osainen)
malli 2
paloittain määritelty funktio (N-osainen)
malli 3
PassErr, ohita virhe 115
Pdf( ) 58
permutaatiot, nPr( ) 110
piecewise( ) 116
pienempi tai yhtä suuri kuin, { 191
pienin yhteinen jaettava, lcm 80
piirrä 212-214
pikavalinnat, näppäimistö 225
piste
erotus, .- 188
osamäärä, ./ 188
potenssi, .^ 189
summa, .+ 187
tulo, .* 188
tulo, dotP( ) 45
poissCdf( ) 116
poissPdf( ) 116
poista
poista, ClrErr 24
tyhjät elementit listasta 41
poistaminen
muuttuja, DelVar 40
polaarinen
koordinaatti, R▸Pr( ) 125
koordinaatti, R▸Pθ( ) 124
vektorinäyttö, 4Polaarinen 117
polyEval( ), laske polynomi 117
polynomit
laske, polyEval( ) 117
satunnainen, randPoly( ) 127
PolyRoots() 118
potenssi, ^ 186
potenssiregressio,
PowerReg 118, 131, 133, 162
PowerReg, potenssiregressio 118
Prgm, määritä ohjelma 119
prodSeq() 120
product( ), tulo 120
propFrac, varsinainen murtoluku 120
prosentti, % 189
PyydäMerkkij 133
Pyyntö 131
pyöristä, round() 137
päivämäärien väliset päivät, dbd( ) 37
pääoman maksuerien summa 198

Q

QR-hajottaminen, QR 121
QR, QR-hajottaminen (tekijöihin jako) 121
QuadReg, 2. asteen regressio 122
QuartReg, 4. asteen regressio 123

R

R, radiaani 200
R▸Pr( ), polaarinen koordinaatti 125
R▸Pθ( ), polaarinen koordinaatti 124
radiaani, R 200
rahan aika-arvo, Korko 169
rahan aika-arvo, maksuerien lukumäärä 169
rahan aika-arvo, maksuerä 169
rahan aika-arvo, nykyarvo 170
rahan aika-arvo, Tuleva arvo 168
rajoittava operaattori "|" 203
rajoittava operaattori, 226

laskemisjärjestys .....
rand(), satunnaisluku 125
randBin, satunnaisluku 125
randInt(), satunnainen kokonaisluku 126
randMat(), satunnaismatriisi 126
randNorm(), satunnainen normaalijakauma 127
randPoly(), satunnaispolynomi 127
randSamp( ) 127
RandSeed, satunnaisluvun siemenluku 127
reaali, real() 128
real(), reaali 128
ref(), rivi-echelon-muoto 129
RefreshProbeVars 130
regressiot
2. asteen, QuadReg 122
3. asteen, CubicReg 34
4. asteen, QuartReg 123
eksponentiaalinen, ExpReg 52
lineaarinen regressio, LinRegAx . 82
lineaarinen regressio, LinRegBx . 81, 83
logaritminen, LnReg 88
Logistinen 91
logistinen, Logistic 92
mediaani-mediaani-suora, MedMed 96
MultReg 100
potenssiregressio, PowerReg . 118, 131, 133, 162
sinimuotoinen, SinReg 150
remain(), jäännös 131
return, Palaa 134
right( ), oikea 134
right, right( ) 49, 175
ristitulo, crossP( ) 33
rivi-echelon-muoto, ref() 129
rk23( ), Runge Kutta -funktio 134
rotate(), kierrä 136
round(), pyöristä 137
rowAdd(), matriisin rivien yhteenlasku 138
rowDim(), matriisin rivikoko 138
rowNorm(), matriisin rivinormi 138
rowSwap(), matriisin rivinvaihto ... 138
rref( ), sievennetty rivi-echelon-muoto 139
ryhmät, lukitseminen ja vapauttaminen 90, 172
ryhmät, lukitustilan testaaminen ... 67

S

samanaikaiset yhtälöt, simult( ) 147
satunnainen
matriisi, randMat( ) 126
normaalijakauma, randNorm() . 127
polynomi, randPoly() 127
siemenluku, RandSeed 127
satunnaisotos 127
sec^{-1}( ), käänteissekantti 140
sec( ), sekantti 139
$sech^{-1}$( ), käänteinen hyperbolinen sekantti 140
sech( ), hyperbolinen sekantti 140
sekaluvut, käyttäen funktiota propFrac(› 120
sekuntimuoto, " 201
sekvenssi, seq( ) 141
seq( ), sekvenssi 141
seqGen( ) 142
seqn( ) 143
sequence, seq( ) 142-143
setMode( ), aseta tila 143
shift( ), siirrä 145
sievennetty rivi-echelon-muoto, rref( ) 139
sievennysjärjestys 226
sign( ), etumerkki 146
siirry, Goto 71
silmukan loppu, EndLoop 93
silmukka, Loop 93
simult( ), samanaikaiset yhtälöt 147
sin^{-1}( ), arkussini 149
sin( ), sini 148
$sinh^{-1}$( ), hyperbolinen arkussini 150
sinh( ), hyperbolinen sini 149
sini, sin( ) 148
siniregressio, SinReg 150

SinReg, siniregressio 150
SortA, lajittele nousevaan järjestykseen 152
SortD, lajittele laskevaan järjestykseen 152
sqrt( ), neliöjuuri 153
stat.results 154
stat.values 155
stdDevPop( ), perusjoukon keskihajonta 155
stdDevSamp( ), otoksen keskihajonta 156
Stop-komento 157
string( ), lauseke merkkijonoksi 157
strings
right, right( ) 49, 175
studentint-todennäköisyysjakauma, tCdf( ) 162
studentint-todennäköisyystiheys, tPdf( ) 165
subMat( ), alimatriisi 157, 159
sum( ), yhteenlasku 158
sumIf( ) 158
summa (G)
malli 5
summa, S( ) 197
sumSeq() 159
suorakulmavektorinäyttö, ▸Rect 128
suurempi kuin, > 192
suurempi tai yhtä suuri kuin, | 192
suurin yhteinen jakaja, gcd( ) 61

T

t-testi, tTest 167
T, transponoi 159
tai (Boolen), tai 113
tai, Boolen operaattori 113
tallentaminen
symboli, & 205
talouslaskentafunktiot, tvmFV( ) 168
talouslaskentafunktiot, tvmI( ) 169
talouslaskentafunktiot, tvmN( ) 169
talouslaskentafunktiot, tvmPmt( ) 169
talouslaskentafunktiot, tvmPV( ) 170
$\tan^{-1}$( ), arkustangentti 160
tan( ), tangentti 160
tangentti, tan( ) 160
$\tanh^{-1}$( ), hyperbolinen areatangentti 162
tanh( ), hyperbolinen tangentti 161
tCdf( ), studentint-todennäköisyysjakauma 162
tekijä, factor( ) 54
Teksti-komento 162
Test_2S, 2 otoksen F-testi 59
tila-asetukset, getMode( ) 68
tilastot
kahden muuttujan tulokset, TwoVar 170
kertoma, ! 194
keskiarvo, mean( ) 95
keskihajonta, stdDev( ) 155-156, 173
kombinaatiot, nCr( ) 104
mediaani, median( ) 96
permutaatiot, nPr( ) 110
satunnainen normaalijakauma, randNorm( ) 127
satunnainen siemenluku, RandSeed 127
varianssi, variance( ) 173
yhden muuttujan tilastot, OneVar 112
tilat
asetus, setMode( ) 143
TInterval, t-luottamusväli 163
TInterval_2Samp, kahdenotoksen t-luottamusvälin 164
todennäköisyystiheys, normPdf( ) 109
toinen derivaatta
malli 6
tPdf( ), studentint-todennäköisyystiheys 165
trace( ) 165
transponoi, T 159
Try, virheenkäsittelykomento 166
tTest, t-testi 167
tTest_2Samp, kahden otoksen t-testi 167
tulo (P)
malli 5
tulo, P( ) 196
tulo, product( ) 120

tulokset, tilastot 154
tulosarvot, tilastot 155
tunnus, Lbl 79
TVM-argumentit 170
TVM-funktioiden argumentit 170
tvmFV( ) 168
tvmI( ) 169
tvmN( ) 169
tvmPmt( ) 169
tvmPV( ) 170
TwoVar, kahden muuttujan tulokset 170
Tyhjennä 211
tyhjä, testi 79
tyhjän testi, isVoid( ) 79
tyhjät elementit 223
tyhjät elementit, poista 41
täyttö 215-216

U

unitV( ), yksikkövektori 172
unLock, vapauta muuttuja tai muuttujaryhmä 172
uusi
lista, newList( ) 106
matriisi, newMat( ) 106

V

vaihto, shift( ) 145
varianssi, variance( ) 173

W

warnCodes( ), Warning codes 175
varoituskoodit ja -viestit 239
varPop( ) 173
varSamp( ), otoksen varianssi 173
varsinainen murtoluku, propFrac 120
vasen, left( ) 80
vastaus (viimeinen), ans 12
vektorit
näyttö vektorina sylinterikoordinaatistossa, ▸Cylind 36
pistetulo, dotP( ) 45
ristitulo, crossP( ) 33
yksikkö, unitV( ) 172
when( ), kun 175
while, While 176
While, while 176
virheet ja vianmääritys
ohita virhe, PassErr 115
poista virhe, ClrErr 24
vähennyslasku, N 183

X

x:n suorakulmakoordinaatti, P▸Rx( ) 114
x², neliö 187
XNOR 193
xor, Boolen eksklusiivinen tai (or) 176

Y

y:n suorakulmakoordinaatti, P▸Ry( ) 115
yhden muuttujan tilastot, OneVar 112
yhteenlasku, + 183
yhteenlasku, sum( ) 158
Yhtälökäyttöjärjestelmä (EOS) 226
yhtälöpari (2 yhtälöä)
malli 3
yhtälöryhmä (N yhtälöä)
malli 3
yksikkövektori, unitV( ) 172
yläraja, ceiling( ) 20-21, 32

Z

zInterval, z-luottamusväli 177
zInterval_1Prop, yhden osuuden z-luottamusväli 178
zInterval_2Prop, kahden osuuden z-luottamusväli 178
zInterval_2Samp, kahden näytteen z-luottamusväli 179
zTest 180
zTest_1Prop, yhden osuuden z-testi 180
zTest_2Prop, kahden osuuden z-testi 181
zTest_2Samp, kahden otoksen z-testi 182

X

χ^2Cdf( ) 22
χ^2GOF 22
χ^2Pdf( ) 23