

TI-Nspire™ CX CAS Referanseguide

Viktig Informasjon

Unntatt som uttrykkelig oppgitt i lisensen som medfølger et program, gir ikke Texas Instruments noen garantier, verken uttrykte eller implisitte, inkludert men ikke begrenset til implisitte garantier for salgbarhet eller egnethet til et bestemt formål, med hensyn til noen programmer eller bokmateriale, og gjør slike materialer tilgjengelige utelukkende på en "som det er"-basis ("as-is") Texas Instruments skal under ingen omstendigheter holdes ansvarlig overfor noen for spesielle, kollaterale, tilfeldige eller følgeskader i forbindelse med eller med bakgrunn i kjøp eller bruk av disse materialene, og det eneste og eksklusive økonomiske ansvaret til Texas Instruments, uavhengig av søksmålsform, skal ikke overskride prisen som er angitt i lisensen til programmet. Videre skal ikke Texas Instruments holdes økonomisk ansvarlig for noen form for krav mot bruk av dette materialet av noen annen part

© 2024 Texas Instruments Incorporated

Faktiske produkter kan være litt annerledes enn på bilder.

Innhold

Uttrykkssjabloner	1
Alfabetisk oversikt	7
A	7
B	15
C	19
D	35
E	44
F	52
G	59
I	69
L	76
M	91
N	100
O	107
P	110
Q	116
R	119
S	134
T	152
U	164
V	165
W	166
X	168
Z	169
Symboler	175
TI-Nspire™ CX II – Tegnekommandoer	194
Grafikkprogrammering	194
Grafikkskjerm	194
Standardvisning og innstillinger	195
Feilmeldinger på grafikkskjerm	196
Ugyldige kommandoer i grafikkmodus	196
C	197
D	198
F	201
G	203
P	204
S	206
U	208

Tomme (åpne) elementer	209
Snarveier/hurtigtaster for å legge inn matematiske uttrykk	211
EOS™ (Ligningsoperativsystem)-hierarkiet	213
TI-Nspire CX II – TI-Basic programmeringsfunksjoner	215
Auto-innrykk i Programmeringseditor	215
Forbedrede feilmeldinger for TI-Basic	215
Konstanter og verdier	218
Feilkoder og feilmeldinger	219
Advarselskoder og -meldinger	227
Generell informasjon	229
Stikkordregister	230

Uttrykkssjabloner

Med uttrykkssjablonene er det enkelt å skrive inn uttrykk i standardisert, matematisk fremstilling. Når du setter inn en sjablon, kommer den til syne på kommandolinjen med små blokker i posisjoner der du kan legge inn elementer. En markør viser hvilke elementer du kan sette inn.

Bruk pilknappene eller trykk på **tab** for å bevege markøren til hvert elements posisjon, og skriv inn en verdi eller et uttrykk for elementet. Trykk på **enter** eller **ctrl enter** for å behandle uttrykket.

Brøk-sjablon

ctrl **÷** taster



Merk: Se også / (divider), side 176.

Eksempel:

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

Eksponent-sjablon

^ tast



Merk: Skriv inn den første verdien, trykk på **^** og skriv så inn eksponenten. For å flytte markøren tilbake til grunnlinjen, trykk på høyre pil (**▶**).

Merk: Se også ^ (potens), side 176.

Eksempel:

$$2^3 = 8$$

Kvadratrot-sjablon

ctrl **x²** taster



Merk: Se også $\sqrt{}$ (kvadratrot), side 184.

Eksempel:

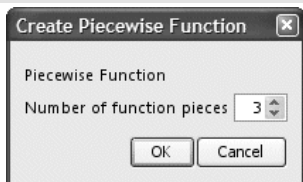
N-te rot-sjablon

ctrl **^** taster



Merk: Se også **rot()**, side 130.

Eksempel:



Merk: Se også `stykkevis()`, side 111.

Sjablon for ligningssystemer med 2 ukjente



Oppretter et system av to ligninger. For å legge en rad til et eksisterende system, klikk inn sjablonen og gjenta sjablonen.

Merk: Se også `system()`, side 152.

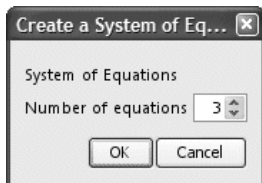
Eksempel:

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

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

Sjablon for ligningssystemer med N ukjente

Lar deg opprette et system av N ligninger. Ber om N .



Merk: Se også `system()`, side 152.

Eksempel:

Se eksemplet for Sjabloner for ligningssystemer (2 ligninger).

Sjablon for absoluttverdi



Merk: Se også `abs()`, side 7.

Eksempel:

Sjablon for absoluttverdi

Katalog > 

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

gg°mm'ss.ss'' sjablon

Katalog > 

°'°'°'

Eksempel:

Lar deg sette inn vinkler i **gg° mm' ss.ss''** - format, der **gg** er antallet desimale grader, **mm** er antallet minutter og **ss.ss** er antallet sekunder.

Matrise-sjablon (2 x 2)

Katalog > 

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

Eksempel:

Oppretter en 2 x 2-matrise.

Matrise-sjablon (1 x 2)

Katalog > 

$\left[\square \quad \square \right]$

Eksempel:

$$\text{crossP}(\left[\begin{array}{cc} 1 & 2 \end{array} \right], \left[\begin{array}{cc} 3 & 4 \end{array} \right]) \quad \left[\begin{array}{ccc} 0 & 0 & -2 \end{array} \right]$$

Matrise-sjablon (2 x 1)

Katalog > 

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

Eksempel:

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

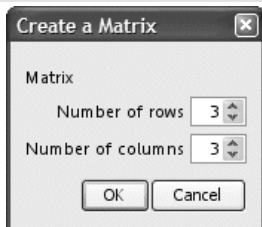
Matrise-sjablon (m x n)

Katalog > 

Sjablonen kommer til syne etter at du er blitt bedt om å spesifisere antallet rader og kolonner.

Eksempel:

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



Merk: Hvis du oppretter en matrise med et stort antall rader og kolonner, må du muligens vente en liten stund før den vises på skjermen.

Sum-sjablon (Σ)

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

Eksempel:

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

Merk: Se også $\Sigma()$ (sumSeq), side 185.

Produkt-sjablon (Π)

$$\prod_{i=1}^n ()$$

Eksempel:

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

Merk: Se også $\Pi()$ (prodSeq), side 184.

Første derivert-sjablon

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

Eksempel:

Merk: Se også $d()$ (derivert), side 184.

Andre derivert-sjablon

Katalog > 

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

Eksempel:

Merk: Se også **d()** (derivert), side 184.

Bestemt integral-sjablon

Katalog > 

$$\int_a^b \square dx$$

Eksempel:

Alfabetisk oversikt

Elementer med navn som ikke er alfabetiske (som f.eks. +, !, og >) er opplistet på slutten av dette avsnittet (side 175). Hvis ikke annet er spesifisert, er alle eksemplene i dette avsnittet utført i grunninnstilling-modus, og det antas at ingen av variablene er definert.

A

abs()

Katalog > 

abs(Liste1) ⇒ *liste*

abs(Matrise1) ⇒ *matrise*

Returnerer argumentets absoluttverdi.

Merk: Se også **Absoluttverdi-sjablon**, side 3.

Hvis argumentet er et komplekst tall, returneres absoluttverdien (modulus).

Merk: Alle ubestemte variabler behandles som reelle variabler.

amortTbl()

Katalog > 

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

Amortiseringsfunksjon som returnerer en matrise som en amortiseringstabell for et sett med TVM-argumenter.

NPmt er antallet betalinger som skal inkluderes i tabellen. Tabellen starter med den første betalingen.

N, I, PV, Pmt, FV, PpY, CpY og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 162.

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

- Hvis du utelater *Pmt*, grunninnstilles den til **$Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$** .
- Hvis du utelater *FV*, grunninnstilles den til **$FV = 0$** .
- Grunninnstillingene for *PpY*, *CpY* og *PmtAt* er de samme som for TVM-funksjonene.

avrundVerdi spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

Kolonnene i resultatmatrisen er i denne rekkefølgen: Betalingsnummer, betalt rentebeløp, betalt hovedbeløp og balanse.

Balansen som vises i rad n er balansen etter betaling n .

Du kan bruke resultatmatrisen som inndata for de andre amortiseringsfunksjonene $\Sigma\text{Int}()$ og $\Sigma\text{Prn}()$, side 185, og **bal()**, side 15.

and

BoolskUttr1 and *BoolskUttr2* \Rightarrow *Boolsk uttrykk*

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

Boolsk liste1 and *Boolsk liste2* \Rightarrow *Boolsk liste*

Boolsk matrise1 and *Boolsk matrise2* \Rightarrow *Boolsk matrise*

Returnerer sann eller usann eller en forenklet form av opprinnelig uttrykk.

Heltall1 and *Heltall2* \Rightarrow *heltall*

Sammenlikner to reelle heltall bit-for-bit med en and-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis en av bitene er 1; ellers er resultatet 0. Den returnerte verdien representerer bit-resultatene og vises i grunntallmodus.

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 10).

I heksades grunntall-modus:

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

Viktig: Null, ikke bokstaven O.

I binær grunntall-modus:

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

I desimalt grunntall-modus:

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

and

Katalog > 

Hvis du skriver inn et desimalt heltall som er for stort for en 64-biters binær form med fortegn, brukes en symmetrisk modul-handling for å sette verdien inn i gyldig område.

Merk: Et binært innlegg kan bestå av opptil 64 siffer (i tillegg til prefikset Ob). Et heksadesimalt innlegg kan bestå av opptil 16 siffer.

angle() vinkel

Katalog > 

Returnerer vinkelen til argumentet, tolker argumentet som et komplekst tall.

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

$\text{angle}(\text{Liste1}) \Rightarrow \text{liste}$

$\text{angle}(\text{Matrise1}) \Rightarrow \text{matrise}$

Returnerer en liste eller vinkelmatrise av elementene i *Liste1* eller *Matrise1*, tolker hvert element som et komplekst tall som representerer et to-dimensjonalt, rektangulært koordinatpunkt.

ANOVA

Katalog > 

ANOVA *Liste1, Liste2[,Liste3,..., Liste20]*
[,Merke]

Utfører en enveis analyse av varians for å sammenlikne gjennomsnitt for mellom 2 og 20 populasjoner. En oversikt over resultatene lagres i *stat.results*-variabelen (side 147).

Merke=0 for Data, *Merke*=1 for Stats

Utdata-variabel	Beskrivelse
stat.F	Verdi av F-statistikken

Utdata-variabel	Beskrivelse
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader frihet for gruppene
stat.SS	Sum av kvadrater for gruppene
stat.MS	Gjennomsnitt av kvadrater for gruppene
stat.dfError	Grader av frihet for feilene
stat.SSError	Sum av kvadrater av feilene
stat.MSError	Gjennomsnitt av kvadrater av feilene (gjennomsnittlig kvadratavvik)
stat.sp	Felles standardavvik
stat.xbarliste	Gjennomsnitt av listenes inndata
stat.CLowerList	95% konfidensintervaller for gjennomsnittet av hver inndata-liste
stat.UpperList	95% konfidensintervaller for gjennomsnittet av hver inndata-liste

ANOVA2way

Katalog > 

ANOVA2way *Liste1, Liste2[,...[,Liste10]]*
[,LevRad]

Beregner en toveis analyse av varians for å sammenlikne gjennomsnitt for mellom 2 og 10 populasjoner. En oversikt over resultatene lagres i *stat.results*-variabelen (side 147).

LevRad=0 for Blokk

LevRad=2,3,...,Len-1, for To Faktor, hvor
 $Len = \text{lengde}(\text{Liste1}) = \text{lengde}(\text{Liste2}) = \dots = \text{lengde}(\text{Liste10})$ og $Len / LevRad \in \mathbb{N} \setminus \{2,3,\dots\}$

Utdata: Blokk-oppsett

Utdata-variabel	Beskrivelse
stat.F	F-statistikk over kolonnefaktoren
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader frihet for kolonnefaktoren
stat.SS	Sum av kvadrat for kolonnefaktoren
stat.MS	Gjennomsnitt av kvadrater for kolonnefaktor

Utdata-variabel	Beskrivelse
stat.FBlock	F-statistikk for faktor
stat.PValBlock	Minste sannsynlighet som null-hypotesen kan forkastes ved
stat.dfBlockstat.dfBlock	Grader frihet for faktor
stat.SSBlock	Sum av kvadrater for faktor
stat.MSBlock	Gjennomsnitt av kvadrater for faktor
stat.dfError	Grader av frihet for feilene
stat.SSError	Sum av kvadrater av feilene
stat.MSError	Gjennomsnitt av kvadrater av feilene (gjennomsnittlig kvadratavvik)
stat.s	Standardavvik for feilen

KOLONNEFAKTOR Utdata

Utdata-variabel	Beskrivelse
stat.Fcol	F-statistikk over kolonnefaktoren
stat.PValCol	Kolonnefaktorens sannsynlighetsverdi
stat.dfCol	Grader frihet for kolonnefaktoren
stat.SSCol	Sum av kvadrater av kolonnefaktoren
stat.MSCol	Gjennomsnitt av kvadrater for kolonnefaktor

RADFAKTOR Utdata

Utdata-variabel	Beskrivelse
stat.Frow	F-statistikk over radfaktoren
stat.PValRow	Kolonnefaktorens sannsynlighetsverdi
stat.dfRow	Grader frihet for radfaktoren
stat.SSRow	Sum av kvadrater for radfaktoren
stat.MSRow	Gjennomsnitt av kvadrater for radfaktor

INTERAKSJON Utdata

Utdata-variabel	Beskrivelse
stat.FInteract	F-statistikk over interaksjonen
stat.PValInteract	Interaksjonens sannsynlighetsverdi

Utdata-variabel	Beskrivelse
stat.dfInteract	Grader av frihet for interaksjonen
stat.SSInteract	Sum av kvadrater for interaksjonen
stat.MSInteract	Gjennomsnitt av kvadrater for interaksjon

FEIL Utdata

Utdata-variabel	Beskrivelse
stat.dfError	Grader av frihet for feilene
stat.SSError	Sum av kvadrater av feilene
stat.MSError	Gjennomsnitt av kvadrater av feilene (gjennomsnittlig kvadratavvik)
s	Standardavvik for feilen

Ans (svar)

ctrl (-) taster

Ans⇒verdi

56 56

Returnerer resultatet av det sist behandlede uttrykket.

56+4 60

60+4 64

approx() (tilnærm)

Katalog > 

Returnerer behandlingen av argumentet som et uttrykk med desimalverdier, hvis mulig, uavhengig av om modus er **Auto** eller **Tilnærmet**.

Dette er det samme som å skrive inn argumentet og trykke på **ctrl enter**.

$\text{approx}\left(\frac{1}{3}\right)$ 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}

$\text{approx}([\sqrt{2}, \sqrt{3}])$ [1.41421 1.73205]

$\text{approx}\left(\left[\frac{1}{3}, \frac{1}{9}\right]\right)$ [0.333333 0.111111]

approx(Liste1)⇒liste

$\text{approx}\{\{\sin(\pi), \cos(\pi)\}\}$ {0,-1}

approx(Matrise1)⇒matrise

$\text{approx}([\sqrt{2}, \sqrt{3}])$ [1.41421 1.73205]

Returnerer en liste eller *matrise* hvor hvert element er blitt behandlet til en desimalverdi, hvis mulig.

►approxFraction()Katalog > *Liste* ►**approxFraction**([*Tol*])⇒*liste*

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

0.833333

Matrise ►**approxFraction**([*Tol*])⇒*matrise*

$$0.8333333333333333 \text{►approxFraction}(5 \cdot 10^{-14})$$

 $\frac{5}{6}$

Returnerer argumentet som en brøk med en toleranse på *Tol*. Hvis *tol* utelates, brukes en toleranse på 5.E-14.

$$\{\pi, 1.5\} \text{►approxFraction}(5 \cdot 10^{-14})$$

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

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive @>**approxFraction** (...).

approxRational()Katalog > *Liste*[, *Tol*]⇒*liste*

$$\text{approxRational}(0.333, 5 \cdot 10^{-5})$$

 $\frac{333}{1000}$ *approxRational*(*Matrise*[, *Tol*])⇒*matrise*

$$\text{approxRational}(\{0.2, 0.33, 4.125\}, 5 \cdot 10^{-14})$$

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

Returnerer argumentet som en brøk med en toleranse på *Tol*. Hvis *Tol* utelates, brukes en toleranse på 5.E-14.

arccos()Se $\cos^{-1}()$, side 27.**arccosh()**Se $\cosh^{-1}()$, side 29.**arccot()**Se $\cot^{-1}()$, side 29.**arcoth()**Se $\coth^{-1}()$, side 30.**arccsc()**Se $\csc^{-1}()$, side 32.

arccsch()Se $\text{csch}^{-1}()$, side 33.**arcsec()**Se $\text{sec}^{-1}()$, side 134.**arcsech()**Se $\text{sech}^{-1}()$, side 134.**arcsin()**Se $\text{sin}^{-1}()$, side 142.**arcsinh()**Se $\text{sinh}^{-1}()$, side 143.**arctan()**Se $\text{tan}^{-1}()$, side 153.**arctanh()**Se $\text{tanh}^{-1}()$, side 154.**() (utvid/sett sammen)**Katalog > **augment(Liste1, Liste2)⇒liste**Returnerer en ny liste som er *Liste2* lagt til på slutten av *Liste1*. $\text{augment}(\{1,-3,2\},\{5,4\}) \quad \{1,-3,2,5,4\}$ **augment(Matrise1, Matrise2)⇒matrise**Returnerer en ny matrise som er *Matrise2* lagt til på *Matrise1*. Når tegnet “,” brukes, må matrisen ha like raddimensjoner, og *Matrise2* er lagt til på *Matrise1* som nye kolonner. Endrer ikke *Matrise1* eller *Matrise2*.

1 2	→ m1	1 2
3 4		3 4

5	→ m2	5
6		6

$\text{augment}(m1,m2)$	1 2 5
	3 4 6

avgRC(*Uttr1*, *Var* [=Verdi] [, *Trinn*]) \Rightarrow uttrykk

avgRC(*Uttr1*, *Var* [=Verdi] [, *Liste1*]) \Rightarrow liste

avgRC(*Liste1*, *Var* [=Verdi] [, *Trinn*]) \Rightarrow liste

avgRC(*Matrise1*, *Var* [=Verdi] [, *Trinn*]) \Rightarrow matrise

Returnerer differenskvotienten tatt i positiv retning (gjennomsnittlig endringshastighet).

Uttr1 kan være et brukerdefinert funksjonsnavn (se **Func**).

Hvis *verdi* er spesifisert, opphever den eventuell forhåndsstilt verdi eller aktuell "|" erstatning for variabelen.

Trinn er trinnverdien. Hvis *Trinn* utelates, brukes grunninnstilling 0,001.

Merk at den liknende funksjonen **centralDiff** () bruker derivasjonskvotienten.

B

bal()

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

bal(*NPmt*, *amortTabell*) \Rightarrow verdi

Amortiseringsfunksjon som beregner planlagt balanse etter en spesifisert betaling.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 162.

NPmt spesifiserer det betalingsnummeret som du vil at dataene skal beregnes etter.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 162.

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

- Hvis du utelater Pmt , grunninnstilles den til $Pmt=tvmpmt$ ($N, I, PV, FV, PpY, CpY, PmtAt$).
- Hvis du utelater FV , grunninnstilles den til $FV=0$.
- Grunninnstillingene for PpY , CpY og $PmtAt$ er de samme som for TVM-funksjonene.

avrundVerdi spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

bal($NPmt, amortTabell$) beregner balansen etter betalingsnummer $NPmt$, basert på amortiseringstabell $amortTabell$. Argumentet $amortTabell$ må være en matrise i den form som er beskrevet under **amortTbl()**, side 7.

Merk: Se også $\Sigma Int()$ og $\Sigma Prn()$, side 185.

►Base2 (Grunntall2)

Heltall1 ►Base2⇒*heltall*

256►Base2	0b100000000
0h1F►Base2	0b11111

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @►Base2.

Regner om *Heltall1* til et binært tall. Binære eller heksadesimale tall har alltid et prefiks, hhv. 0b eller 0h. Null, ikke bokstaven O, fulgt av b eller h.

0b *binærTall*

0h *heksadesimalTall*

Et binært tall kan bestå av opptil 64 siffer. Et heksadesimaltall kan bestå av opptil 16.

Uten prefiks blir *Heltall1* behandlet som et desimalt tall (grunntall 10). Resultatet vises binært, uavhengig av grunntallets modus.

Negative tall vises på "toerkomplement"-form. Eksempel:

-1 vises som

0hFFFFFFFFFFFFFFFF i heksadesimal
modus

0b111...111 (64 1-ere) i binær modus

-2⁶³ vises som

0h8000000000000000 i heksadesimal
modus

0b100...000 (63 nuller) i binær modus

Hvis du oppgir et desimalt heltall som
ligger utenfor verdiområdet for et 64-bit
binært tall med fortegn, vil en
symmetrisk modulusoperasjon bli brukt
til å konvertere tallet inn i gyldig
verdiområde. Se følgende eksempler på
verdier utenfor verdiområdet.

2⁶³ blir -2⁶³ og vises som

0h8000000000000000 i heksadesimal
modus

0b100...000 (63 nuller) i binær modus

2⁶⁴ blir 0 og vises som

0h0 i heksadesimal modus

0b0 i binær modus

-2⁶³ - 1 blir 2⁶³ - 1 og vises som

0h7FFFFFFFFFFFFFFF i heksadesimal
modus

0b111...111 (64 1-ere) i binær modus

►Base10 (Grunntall10)

Katalog > 

Heltall1 ►Base10⇒*heltall*

0b10011►Base10	19
----------------	----

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @►Base10.

0h1F►Base10	31
-------------	----

Omregner *Heltall1* til et desimaltall (grunntall 10). Binært eller heksadesimalt inndata må alltid ha et prefiks, hhv. 0b eller 0h.

0b binærTall

0h *heksadesimalTall*

Null, ikke bokstaven O, fulgt av b eller h.

Et binært tall kan bestå av opptil 64 siffer. Et heksadesimaltall kan bestå av opptil 16.

Uten prefiks behandles *Heltall1* som desimaltall. Resultatet vises i desimaltall, uavhengig av grunntall-modus.

►Base16 (Grunntall16)

Katalog > 

Heltall1 ►Base16⇒*heltall*

256►Base16	0h100
------------	-------

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @►Base16.

0b111100001111►Base16	0hFOF
-----------------------	-------

Omregner *Heltall1* til et heksadesimaltall. Binære eller heksadesimale tall har alltid et prefiks, hhv. 0b eller 0h.

0b binærTall

0h *heksadesimalTall*

Null, ikke bokstaven O, fulgt av b eller h.

Et binært tall kan bestå av opptil 64 siffer. Et heksadesimaltall kan bestå av opptil 16.

Uten prefiks blir *Heltall1* behandlet som et desimaltall (grunntall 10). Resultatet vises i heksadesimal, uavhengig av grunntallets modus.

Hvis du oppgir et desimalt heltall som er for stort for et 64-bit binært tall med fortegn, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se ►Base2, side 16.

binomCdf()

binomCdf(n,p)⇒*liste*

binomCdf

binomCdf($n,p, \text{nedreGrense}, \text{øvreGrense}$)⇒*tall* hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er lister

binomCdf($n,p, \text{øvreGrense}$)for $P(0 \leq X \leq \text{øvreGrense})$ ⇒*tall* hvis *øvreGrense* er et tall, *liste* hvis *øvreGrense* er en liste

Beregner en kumulativ sannsynlighet for diskret binomisk fordeling med n antall forsøk og sannsynlighet p for å finne treff ved hvert forsøk.

For $P(X \leq \text{øvreGrense})$, sett *nedreGrense*=0

binomPdf()

binomPdf(n,p)⇒*liste*

binomPdf($n,p, XVerd$)⇒*tall* hvis *XVerd* er et tall, *liste* hvis *XVerd* er en liste

Beregner en sannsynlighet ved *XVerd* for diskret binomisk fordeling med n antall forsøk og sannsynlighet p for å finne treff ved hvert forsøk.

C**ceiling() (øvre)**

Returnerer det nærmeste heltallet som er \geq argumentet.

ceiling(.456)

1.

ceiling() (øvre)

Katalog > 

Argumentet kan være et reelt eller et komplekst tall.

Merk: Se også **floor()** (nedre).

ceiling(*Liste1*) \Rightarrow *liste*

ceiling(*Matrise1*) \Rightarrow *matrise*

Returnerer en liste eller matrise med den øvre i hvert element.

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

centralDiff()

Katalog > 

centralDiff(*Uttr1*, *Var* [=Verdi]
[, *Trinn*]) \Rightarrow *uttrykk*

centralDiff(*Uttr1*, *Var*
[, *Trinn*]) | *Var*=*Verdi* \Rightarrow *uttrykk*

centralDiff(*Uttr1*, *Var* [=Verdi]
[, *Liste*]) \Rightarrow *liste*

centralDiff(*Liste1*, *Var* [=Verdi]
[, *Trinn*]) \Rightarrow *liste*

centralDiff(*Matrise1*, *Var* [=Verdi]
[, *Trinn*]) \Rightarrow *matrise*

Returnerer den numeriske deriverte ved hjelp av derivasjonskvotient-formelen.

Hvis *verdi* er spesifisert, opphever den eventuell forhåndsvalgt verdi eller aktuell "|" erstatning for variabelen.

Trinn er trinnverdien. Hvis *Trinn* utelates, brukes grunninnstilling 0,001.

Hvis du bruker *Liste1* eller *Matrise1*, blir handlingen avbildet gjennom verdiene i listen eller gjennom matriseelementene.

Merk: Se også.

char()

Katalog > 

char(*Heltall*) \Rightarrow *tegn*

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

char()

Katalog > 

Returnerer en tegnstring som inneholder det tegnet som er nummerert med *Heltall* fra tegnsettet på grafregneren. Gyldig område for *Heltall* er 0–65535.

χ^2 2way

Katalog > 

χ^2 2way *ObsMatrise*

chi22way *ObsMatrise*

Beregner en χ^2 test for samling av “tellingene” på toveis-tabellen i den observerte matrisen *ObsMatrise*. En oversikt over resultatene lagres i *stat.results*-variabelen (side 147).

For informasjon om effekten av tomme elementer i en matrise, se “Tomme (åpne) elementer” (side 209).

Utdata-variabel	Beskrivelse
stat. χ^2	Chi-kvadratstat: $\text{sum}(\text{observert} - \text{forventet})^2 / \text{forventet}$
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader av frihet for chi-kvadratstatistikk
stat.UttrMat	Matrise av forventet element-telletebll ved antatt nullhypotese
stat.KompMat	Matrise av elementbidrag til chi kvadratstatistikk

χ^2 Cdf()

Katalog > 

χ^2 Cdf(*nedreGrense*,*øvreGrense*,*df*) \Rightarrow tall
hvis *nedreGrense* og *øvreGrense* er tall,
liste hvis *nedreGrense* og *øvreGrense* er lister

chi2Cdf(*nedreGrense*,*øvreGrense*,*df*) \Rightarrow tall
hvis *nedreGrense* og *øvreGrense* er tall,
liste hvis *nedreGrense* og *øvreGrense* er lister

Beregner χ^2 -fordelingens sannsynlighet mellom *nedreGrense* og *øvreGrense* for det angitte antall frihetsgrader *df*.

For $P(X \leq \text{øvreGrense})$, sett *nedreGrense* = 0.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

$\chi^2\text{GOF}$ *obsListe,uttrListe,df*

chi2GOF *obsListe,uttrListe,df*

Utfører en test for å bekrefte at utvalgsdata er fra en populasjon som er i overensstemmelse med en angitt fordeling. *obsListe* er en liste over antall, og må inneholde heltall. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat. χ^2	Chi-kvadratstat: $\text{sum}((\text{observert} - \text{forventet})^2 / \text{forventet})$
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader av frihet for chi-kvadratstatistikk
stat.CompList	Elementbidrag til chi kvadratstatistikk

$\chi^2\text{Pdf}(X\text{Verd},df) \Rightarrow$ tall hvis *XVerd* er et tall, liste hvis *XVerd* er en liste

chi2Pdf(*XVerd,df*) \Rightarrow tall hvis *XVerd* er et tall, liste hvis *XVerd* er en liste

Beregner sannsynlighetstettheten (pdf) for χ^2 -fordelingen ved en bestemt *XVerd*-verdi for det angitte antallet frihetsgrader *df*.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 209).

ClearAZ (slettAZ)

ClearAZ

Sletter alle enkelttegn-variabler i det aktuelle oppgaveområdet.

Hvis en eller flere av variablene er låst, viser denne kommandoen en feilmelding og sletter kun de ulåste variablene. Se **unLock**, side 164.

ClrErr (SlettFeil)

ClrErr

Tømmer feilstatus og stiller systemvariabelen *feilKode* til null.

Else -leddet i **Try...Else...EndTry**-blokken bør bruke **ClrErr** eller **PassErr**. Hvis feilen skal bearbeides eller ignoreres, bruk **ClrErr**. Hvis det ikke er kjent hva som skal gjøres med feilen, bruk **PassErr** for å sende den til den neste feilbehandleren. Hvis det ikke er flere ventende **Try...Else...EndTry** feilbehandlere, vises feil-dialogboksen som normalt.

Merk: Se også **PassErr**, side 111, og **Try**, side 158.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

For et eksempel på **ClrErr**, se eksempel 2 under **Try** -kommandoen, side 158.

colAugment() (kolUtvid)

Katalog > 

colAugment(*Matrise1*,
Matrise2) \Rightarrow *matrise*

Returnerer en ny matrise som er *Matrise2* lagt til på *Matrise1*. Matrisene må ha like kolonnedimensjoner, og *Matrise2* er lagt til *Matrise1* som nye rader. Endrer ikke *Matrise1* eller *Matrise2*.

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

colDim()

Katalog > 

colDim(*Matrise*) \Rightarrow *uttrykk*

Returnerer antallet kolonner som ligger i *Matrise*.

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

Merk: Se også **radDim()**.

colNorm()

Katalog > 

colNorm(*Matrise*) \Rightarrow *uttrykk*

Returnerer den største summen av absoluttverdiene for elementene i kolonnene i *Matrise*.

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

Merk: Udefinerte matriseelementer er ikke tillatt. Se også **radNorm()**.

conj()

Katalog > 

conj(*Liste1*) \Rightarrow *liste*

conj(*Matrise1*) \Rightarrow *matrise*

Returnerer den komplekse konjugerte av argumentet.

Merk: Alle ubestemte variabler behandles som reelle variabler.

constructMat

(
Uttr
,Var1,Var2,antRad,antKol) \Rightarrow matrise

Returnerer en matrise basert på argumentene.

Uttr er et uttrykk i variablene *Var1* og *Var2*. Elementene i resultatmatrisen dannes ved å beregne *Uttr* for hver økte verdi av *Var1* og *Var2*.

Var1 økes automatisk fra 1 og opp til *antRad*. I hver rad øker *Var2* fra 1 og opp til *antKol*.

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

CopyVar (kopiVar)

CopyVar *Var1, Var2*

CopyVar *Var1., Var2.*

CopyVar *Var1, Var2* kopierer verdien av variabelen *Var1* til variabelen *Var2*, og oppretter *Var2* om nødvendig. Variabel *Var1* må ha en verdi.

Hvis *Var1* er navnet på en eksisterende brukerdefinert funksjon, kopieres definisjonen av denne funksjonen til funksjon *Var2*. Funksjon *Var1* må være definert.













Var1 må følge reglene for variabelnavn eller være et indirekte uttrykk som kan forenkles til et variabelnavn som oppfyller reglene.

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

Var1. må være navnet på en eksisterende variabelgruppe, for eksempel statistikk *stat.nn*-resultater eller variabler som er opprettet med **LibShortcut()**-funksjonen). Hvis *Var2.* allerede eksisterer, vil denne kommandoen erstatte alle medlemmer som er felles for begge grupper og legge til de medlemmene som ikke allerede eksisterer. Hvis ett eller flere medlemmer av *Var2.* er låst, blir alle medlemmer av *Var2.* værende uendret.

CopyVar *Var1.*, *Var2.* kopierer alle medlemmene av *Var1.* variabelgruppe til *Var2.* gruppe, og oppretter *Var2.* om nødvendig.

Var1. må være navnet på en eksisterende variabelgruppe, for eksempel statistikk *stat.nn*-resultater, eller variabler som er opprettet med **LibShortcut()**-funksjonen. Hvis *Var2.* allerede finnes, vil denne kommandoen erstatte alle medlemmer som er felles for begge grupper, og legge til de medlemmene som ikke allerede finnes. Hvis en enkel (ikke i gruppe) variabel med navnet *Var2* finnes, oppstår det en feil.

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

corrMat()

corrMat(*Liste1*,*Liste2*[,...[,*Liste20*]])

Beregner korrelasjonsmatrisen for den utvidede matrisen [*Liste1*, *Liste2*, . . . , *Liste20*].

cos()

cos(*Liste1*)⇒*liste*

I Grader-vinkelmodus:

cos(*Liste1*) returnerer en liste av cosinus til alle elementer i *Liste1*.

I Gradian-vinkelmodus:

I Radian-vinkelmodus:

Merk: Argumentet tolkes som grader, gradian eller radian av en vinkel, avhengig av aktuell vinkelmodus-innstilling. Du kan bruke °, G eller r for å hoppe over vinkelmodusen midlertidig.

cos(kvadratMatriseI) ⇒ kvadratMatrise

Returnerer matrisens cosinus til kvadratMatriseI. Dette er ikke det samme som å beregne cosinus til hvert element.

Når en skalarfunksjon f(A) virker på kvadratMatriseI (A), beregnes resultatet av algoritmen:

Beregner egenverdiene (li) og egenvektorene (V i) av A.

kvadratMatriseI må kunne diagonaliseres. Den kan heller ikke ha symbolske variabler som ikke er tildelt noen verdi.

Utform matrisene:

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

Da er $A = X B X^{-1}$ og $f(A) = X f(B) X^{-1}$. For eksempel, $\cos(A) = X \cos(B) X^{-1}$ hvor:

cos (B) =

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

Alle beregningene utføres med flytende desimalpunkt-aritmetikk.

I Radian-vinkelmodus:

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

cos⁻¹(ListeI) ⇒ liste

I Grader-vinkelmodus:

$\cos^{-1}()$



$\cos^{-1}(\text{Liste1})$ returnerer en liste over invers cosinus for hvert element i *Liste1*.

Merk: Resultatet returneres som en vinkel i enten grader, gradian eller radian, avhengig av aktuell vinkelmodus-innstilling.

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive `arccos (...)`.

\cos^{-1}
(*kvadratMatrise1*) \Rightarrow *kvadratMatrise*

Returnerer matrisens inverse cosinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers cosinus til hvert element. For mer informasjon om beregningsmetode, se under `cos()`.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Gradian-vinkelmodus:

I Radian-vinkelmodus:

I radian-vinkelmodus og rektangulært, kompleks format:

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

For å se hele resultatet, trykk på \blacktriangle og bruk så \blacktriangleleft og \blacktriangleright for å bevege markøren.

`cosh()`

Katalog >

`cosh(Liste1)` \Rightarrow *liste*

`cosh(Liste1)` returnerer en liste over hyperbolsk cosinus til hvert element i *Liste1*.

`cosh`
(*kvadratMatrise1*) \Rightarrow *kvadratMatrise*

Returnerer matrisens hyperbolske cosinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne hyperbolsk cosinus til hvert element. For mer informasjon om beregningsmetode, se under `cos()`.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Grader-vinkelmodus:

I Radian-vinkelmodus:

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

cosh⁻¹(Liste1) ⇒ *liste*

cosh⁻¹(Liste1) returnerer en liste over invers hyperbolsk cosinus til hvert element i *Liste1*.

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccosh (...)**.

cosh⁻¹
(*kvadratMatrise1*) ⇒ *kvadratMatrise*

Returnerer matrisens inverse hyperbolsk cosinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers hyperbolsk cosinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I radian-vinkelmodus og rektangulært, kompleks format:

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

For å se hele resultatet, trykk på og bruk så og for å bevege markøren.

cot()



cot(Liste1) ⇒ *liste*

I Grader-vinkelmodus:

Merk: Argumentet tolkes som grader, gradianer eller radianer av en vinkel, avhengig av aktuell vinkelmodus-innstilling. Du kan bruke °, G eller r for å hoppe over vinkelmodusen midlertidig.

I Gradian-vinkelmodus:

I Radian-vinkelmodus:

cot⁻¹()



cot⁻¹(Liste1) ⇒ *liste*

I Grader-vinkelmodus:

Merk: Resultatet returneres som en vinkel i enten grader, gradian eller radian, avhengig av aktuell vinkelmodus-innstilling.

cot ⁻¹ (1)	45.
-----------------------	-----

I Gradian-vinkelmodus:

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccot (...)**.

cot ⁻¹ (1)	50.
-----------------------	-----

I Radian-vinkelmodus:

coth()

Katalog > 

coth(Liste1) ⇒ liste

coth⁻¹()Katalog > coth⁻¹(Liste1) ⇒ liste

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccoth** (...).

count() (antall)

Katalog > 

count(Verdi1 eller Liste1 [,Verdi2 eller Liste2 [,...]]) ⇒ verdi

Returnerer samlet antall av alle elementer i argumentene som behandles til numeriske verdier.

Hvert argument kan være et uttrykk, en verdi, liste eller matrise. Du kan blande datatyper og bruke argumenter med forskjellige dimensjoner.

For en liste, matrise eller et celleområde blir hver element behandlet for å bestemme om det bør inkluderes i antallet.

I applikasjonen Lister og regneark kan du bruke et celleområde istedenfor et argument.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 209.

countIf() (tellif)

Katalog > 

countIf(Liste, Kriterium) ⇒ verdi

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

countIf() (tellIf)

Returnerer samlet antall av alle argumenter i *Liste* som møter de spesifiserte *kriterier*.

Kriterium kan være:

- En verdi, et uttrykk eller en streng. For eksempel, **3** teller kun de elementene i *Liste* som forenkles til verdien 3.
- Et boolsk uttrykk som inneholder symbolet **?** som plassholder for hvert element. For eksempel, **?<5** teller kun de elementene i *Liste* som er mindre enn 5.

I applikasjonen Lister og regneark kan du bruke et celleområde istedenfor *Liste*.

Tomme (åpne) elementer i listen ignoreres. For mer informasjon om tomme elementer, se side 209.

Merk: Se også **sumIf()**, side 151, og **frequency()**, side 57.

Teller alle elementer som er lik 3.

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

Teller alle elementer som er lik "def."

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

Teller 1 og 3.

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

Teller 3, 5 og 7.

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

Teller 1, 3, 7 og 9.

cPolyRoots()

cPolyRoots(*Poly*, *Var*) ⇒ *liste*

cPolyRoots(*KoeffListe*) ⇒ *liste*

Den første syntaksen, **cPolyRoots**(*Poly*, *Var*), returnerer en liste over komplekse røtter av polynom *Poly* med hensyn på variabel *Var*.

Den andre syntaksen, **cPolyRoots**(*KoeffListe*), returnerer en liste over komplekse røtter for koeffisienter i *KoeffListe*.

Merk: Se også **polyRoots()**, side 113.

crossP() (kryssprodukt)

crossP(*Liste1*, *Liste2*) ⇒ *liste*

Returnerer kryssproduktet av *Liste1* og *Liste2* som en liste.

crossP() (kryssprodukt)

Katalog > 

Liste1 og *Liste2* må ha lik dimensjon, og dimensjonen må være enten 2 eller 3.

crossP(Vektor1, Vektor2) ⇒ vektor

Returnerer en rad- eller kolonnevektor (avhengig av argumentene) som er kryssproduktet av *Vektor1* og *Vektor2*.

Både *Vektor1* og *Vektor2* må være radvektorer, eller begge må være kolonnevektorer. Begge vektorene må ha lik dimensjon, og dimensjonen må være enten 2 eller 3.

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

csc()

 **tast**

csc(Liste1) ⇒ liste

I Grader-vinkelmodus:

I Gradian-vinkelmodus:

I Radian-vinkelmodus:

csc⁻¹()

 **tast**

csc⁻¹(Liste1) ⇒ liste

I Grader-vinkelmodus:

Merk: Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling.

csc⁻¹(1) 90.

I Gradian-vinkelmodus:

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccsc (...)**.

csc⁻¹(1) 100.

I Radian-vinkelmodus:

csch()

Katalog > 

csch(Liste1) ⇒ liste

csch⁻¹(Liste1) ⇒ liste

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **arccsch** (...).

CubicReg

CubicReg *X*, *Y*, [*Frekv*] [, *Kategori*, *Inkluder*]]

Finner den kubiske polynomiske regresjonen $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$ for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og *Y* er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall ≥ 0 .

Kategori er en liste over kategorikoder for de tilsvarende *X* og *Y*-dataene.

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
stat.a, stat.b, stat.c, stat.d	Regresjonskoeffisienter
stat.R ²	Koeffisientbestemmelse

Utdata-variabel	Beskrivelse
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> , og <i>Inkludert kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>inkludert kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

cumulativeSum()

Katalog > 

cumulativeSum(Liste1) ⇒ liste

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

Returnerer en liste over de kumulative summene av elementene i *Liste1*, og starter med element 1.

cumulativeSum(Matrise1) ⇒ matrise

Returnerer en matrise av de kumulative summene av elementene i *Matrise1*.

Hvert element er den kumulative summen av kolonnen fra topp til bunn.

1 2	→ m1	1 2
3 4		3 4
5 6		5 6
cumulativeSum(m1)		1 2
		4 6
		9 12

Et tomt (åpent) element i *Liste1* eller *Matrise1* produserer et åpent element i den resulterende listen eller matrisen.

For mer informasjon om tomme elementer, se side 209.

Cycle (Løkke)

Katalog > 

Cycle (Løkke)

Funksjonsliste som summerer heltallene fra 1 til 100 og hopper over 50.

Overfører øyeblikkelig kontroll til den neste iterasjonen i aktuell løkke (**For**, **While**, eller **Loop**).

Cycle er ikke tillatt utenfor de tre løkkestrukturene (**For**, **While**, eller **Loop**).

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

Define $g()$ =Func	Done
Local $temp,i$	
$0 \rightarrow temp$	
For $i,1,100,1$	
If $i=50$	
Cycle	
$temp+i \rightarrow temp$	
EndFor	
Return $temp$	
EndFunc	
$g()$	5000

►Cylind

Vektor ►Cylind

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>Cylind.

Viser rad- eller kolonnevektor i sylindrisk form $[r, \angle\theta, z]$.

Vektor må ha nøyaktig tre elementer. Det kan være enten en rad eller en kolonne.

D

dbd()

dbd(dato1,dato2)⇒verdi

Returnerer antallet dager mellom *dato1* og *dato2* ved hjelp av aktuelt-antall-dager-metoden.

dato1 og *dato2* kan være tall eller lister av tall innenfor datoområdet på en vanlig kalender. Hvis både *dato1* og *dato2* er lister, må de være like lange.

dato1 og *dato2* må ligge mellom årene 1950 og 2049.

Du kan legge inn datoene i ett av to formater. Hvor du setter desimalkommaet bestemmer hvilket datoformat du bruker.

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

dbd()

Katalog > 

MM.DDÅÅ (format som vanligvis brukes i USA)

DDMM.ÅÅ (format som vanligvis brukes i Europa)

►DD

Katalog > 

Verdi ►DD⇒verdi

Liste1 ►DD⇒liste

Matrise1 ►DD⇒matrise

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>DD.

Returnerer desimalekvivalenten til argumentet uttrykt i grader. Argumentet er et tall, en liste eller matrise som tolkes av vinkelmodus-innstillingen i gradianer, radianer eller grader.

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

►Decimal

Katalog > 

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>Decimal.

Viser argumentet i desimalform. Denne operatoren kan kun brukes på slutten av kommandolinjen.

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

Define (Definer)

Katalog > 

Define *Var* = *Uttrykk*

Define *Funksjon*(*Param1*, *Param2*, ...) = *Uttrykk*

Definerer variabelen *Var* eller den egendefinerte funksjonen *Funksjon*.

Parametere, som f.eks. *Param1*, er plassholdere for å sette argumenter til funksjonen. Når du kaller opp en egendefinert funksjon, må du legge til argumenter (for eksempel verdier eller variabler) som samsvarer med parametrene. Når funksjonen er kalt opp, behandler den *Uttrykk* ved hjelp av de argumentene som er lagt til.

Var og *Funksjon* kan ikke være navnet på systemvariabel eller innebygget funksjon eller kommando.

Merk: Denne type **Define** er ekvivalent til å utføre uttrykket: *uttrykk* → *Funksjon* (*Param1, Param2*).

Define Funksjon(*Param1, Param2, ...*) = **Funk**
Blokk
EndFunk

Define Program(*Param1, Param2, ...*) = **Prgm**
Blokk
EndPrgm

I denne formen kan egendefinert funksjon eller program utføre en blokk med flere utsagn.

Blokk kan enten være et enkelt utsagn eller en rekke med utsagn på separate linjer. *Blokk* kan også inkludere uttrykk og instruksjoner (som **If**, **Then**, **Else** og **For**).

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

Merk: Se også **Define BiblPriv**, side 37 og **Define BiblOff**, side 38.

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

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

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

Define LibPriv (Definer BiblPriv)

Define LibPriv *Var* = *Uttrykk*

Define LibPriv *Funksjon*(*Param1*, *Param2*, ...) = *Uttrykk*

Define LibPriv *Funksjon*(*Param1*, *Param2*, ...) = **Funk**
Blokk
EndFunk

Define LibPriv *Program*(*Param1*, *Param2*, ...) = **Prgm**
Blokk
EndPrgm

Opererer på samme måte som **Define**, men definerer en privat biblioteksvariabel, -funksjon eller et -program. Private funksjoner og programmer forekommer ikke i Katalogen.

Merk: Se også **Define**, side 36 og **Define LibPub**, side 38.

Define LibPub *Var* = *Uttrykk*

Define LibPub *Funksjon*(*Param1*, *Param2*, ...) = *Uttrykk*

Define LibPub *Funksjon*(*Param1*, *Param2*, ...) = **Funk**
Blokk
EndFunk

Define LibPub *Program*(*Param1*, *Param2*, ...) = **Prgm**
Blokk
EndPrgm

Opererer på samme måte som **Define**, men definerer en felles (offentlig) biblioteksvariabel, -funksjon eller et -program. Felles (offentlige) funksjoner og programmer forekommer i Katalogen etter at biblioteket er blitt lagret og oppdatert.

Merk: Se også **Define**, side 36 og **Define LibPriv**, side 37.

DelVarkatalog > **DelVar** *Var1*[, *Var2*] [, *Var3*] ...**DelVar** *Var*.

Sletter den angitte variabelen eller variabelgruppen fra minnet.

Hvis en eller flere av variablene er låst, viser denne kommandoen en feilmelding og sletter kun de ulåste variablene. Se **unLock**, side 164.

DelVar *Var*. sletter alle medlemmer av *Var*. variabelgruppen (for eksempel statistikk *stat.nn*-resultater eller variabler som er opprettet med **LibShortcut()**-funksjonen). Prikken (.) i denne formen av **DelVar**-kommandoen begrenser den til å slette en variabelgruppe. Enkeltvariabelen *Var* påvirkes ikke.

<i>aa.a:=45</i>	45									
<i>aa.b:=5.67</i>	5.67									
<i>aa.c:=78.9</i>	78.9									
<i>getVarInfo()</i>	<table border="1"> <tbody> <tr> <td><i>aa.a</i></td> <td>"NUM"</td> <td>"{:}"</td> </tr> <tr> <td><i>aa.b</i></td> <td>"NUM"</td> <td>"{:}"</td> </tr> <tr> <td><i>aa.c</i></td> <td>"NUM"</td> <td>"{:}"</td> </tr> </tbody> </table>	<i>aa.a</i>	"NUM"	"{:}"	<i>aa.b</i>	"NUM"	"{:}"	<i>aa.c</i>	"NUM"	"{:}"
<i>aa.a</i>	"NUM"	"{:}"								
<i>aa.b</i>	"NUM"	"{:}"								
<i>aa.c</i>	"NUM"	"{:}"								
<i>DelVar aa.</i>	<i>Done</i>									
<i>getVarInfo()</i>	"NONE"									

delVoid()Katalog > **delVoid**(*Liste1*) \Rightarrow *liste*

<i>delVoid</i> { { 1,void,3 } }	{ 1,3 }
---------------------------------	---------

Returnerer en liste som har innholdet til *Liste1*, der alle tomme (åpne) elementer er fjernet.

For mer informasjon om tomme elementer, se side 209.

det()Katalog > **det**(*kvadratMatrise*[, *Toleranse*]) \Rightarrow *uttrykk*

Returnerer determinanten til *kvadratMatrise*.

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Toleranse*. Denne toleransen brukes bare hvis matrisen har elementer med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Toleranse*.

- Hvis du bruker `ctrl` `enter` eller stiller modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flyttallsaritmetikk.
- Hvis *Toleranse* utelates eller ikke blir brukt, beregnes standardtoleransen som:

$$5E-14 \cdot \text{maks}(\text{dim}(\text{kvadratMatrise})) \cdot \text{radNorm}(\text{kvadratMatrise})$$

diag()

diag(Liste) ⇒ matrise

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

diag(radMatrise) ⇒ matrise

diag(kolonneMatrise) ⇒ matrise

Returnerer en matrise med verdiene i argumentlisten eller matrise i hoveddiagonalen.

diag(kvadratMatrise) ⇒ radMatrise

Returnerer en radmatrise som inneholder elementene fra hoveddiagonalen til *kvadratMatrise*.

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

kvadratMatrise må være kvadrat.

dim()

dim(Liste) ⇒ heltall

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

Returnerer dimensjonen av *Liste*.

dim(Matrise) ⇒ liste

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

Returnerer matrisens dimensjoner som en to-elements liste {rader, kolonner}.

dim()

Katalog >

dim(*Streng*)⇒*heltall*

Returnerer antallet tegn som er inneholdt i tegnstrengen *Streng*.

dim("Hello")	5
dim("Hello "&"there")	11

Disp (Vis)

Katalog >

Disp *utrElStreng1* [, *utrElStreng2*] ...

Viser argumentene i *Calculator*-loggen. Argumentene vises suksessivt, med korte avstander som skille.

Hovedsakelig nyttig i programmer og funksjoner for å sikre visning av mellomregninger.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

```
Define chars(start,end)=Prgm
  For i,start,end
  Disp i," ",char(i)
EndFor
EndPrgm
```

chars(240,243)

	240	ø
	241	ñ
	242	ò
	243	ó

Done

DispAt

Katalog >

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

DispAt lar deg angi linjen der det angitte uttrykket eller den angitte strengen skal vises på skjermen.

Linjenummeret kan angis som et uttrykk.

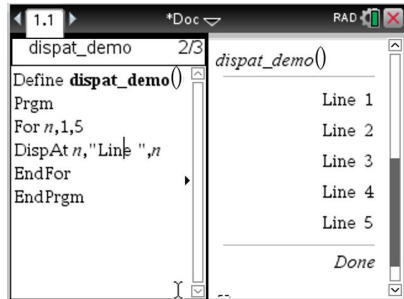
Merk: Linjenummeret gjelder ikke hele skjerm bildet, men kun området som følger umiddelbart etter kommandoen/programmet.

Denne kommandoen lar deg opprette en instrumentbordlignende visning av data fra programmer der verdien til et uttrykk eller en sensoravlesning oppdateres på samme linje.

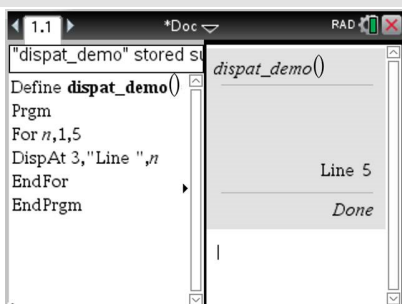
DispAtog Disp kan brukes i samme program.

DispAt

Eksempel



Merk: Maksimumsantall er definert som 8 ettersom dette tilsvarer et fullt display av linjer på grafregnerens skjermbilde, såfremt linjene ikke inneholder matematiske uttrykk i 2D. Det nøyaktige antallet linjer avhenger av innholdet til de viste dataene.



Illustrerende eksempler:

Define z()= Prgm For n,1,3 DispAt 1,"N: ",n Disp "Hallo" EndFor EndPrgm	Utdata z() Iterasjon 1: Linje 1: N:1 Linje 2: Hallo Iterasjon 2: Linje 1: N:2 Linje 2: Hallo Linje 3: Hallo Iterasjon 3: Linje 1: N:3 Linje 2: Hallo Linje 3: Hallo Linje 4: Hallo
Define z1()= Prgm For n,1,3 DispAt 1,"N: ",n EndFor For n,1,4 Disp "Hallo" EndFor EndPrgm	z1() Linje 1: N:3 Linje 2: Hallo Linje 3: Hallo Linje 4: Hallo Linje 5: Hallo

Feilmeldinger:

Feilmelding	Beskrivelse
DispAt-linjenummeret må være mellom 1 og 8	Uttrykk evaluerer linjenummeret utenfor området 1–8 (til og med)
For få argumenter	Funksjonen eller kommandoen mangler et eller flere argumenter.
Ingen argumenter	Det samme som gjeldende Syntaksfeil-dialogboks
For mange argumenter	Begrens argument. Samme feil som Disp.
Ugyldig datatype	Det første argumentet må være et tall.
Åpen: DispAt åpen	Datatypefeilen "Hei alle sammen" iverksettes for den tomme verdien (hvis oppkall er definert)

►DMS (GMS)

Liste ►DMS

I Grader-vinkelmodus:

Matrise ►DMS

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

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @►DMS.

Tolker argumentet som en vinkel og viser ekvivalenten DMS (GGGGG°MM 'SS.ss")-tallet. Se °, ', " (side 189) for DMS-format (grader, minutter, sekunder).

Merk: ►DMS vil omregne fra radianer til grader når det brukes i radian-modus. Hvis inndata blir fulgt av et grader-symbol °, finner det ikke sted noe omregning. Du kan bare bruke ►DMS på slutten av en kommandolinje.

dotP() (prikkP)

dotP(Liste1, Liste2)⇒uttrykk

Returnerer "prikk"produktet av to lister.

dotP(Vektor1, Vektor2)⇒uttrykk

Returnerer “prikk”produktet av to vektorer.



Begge må være radvektorer, eller begge må være kolonnevektorer.

E

e^()

 tast

Merk: Se også **e eksponent-sjablon**, side 2.

Merk: Å trykke på  for å vise e^x er forskjellig fra å trykke på tegnet  på tastaturet.

Du kan legge inn et komplekst tall i $re^{i\theta}$ polar form. Men bruk denne formen bare i radian-vinkelmodus; den forårsaker grunnmengdefeil i grader- eller gradian-vinkelmodus.

$e^{\text{Liste1}} \Rightarrow \text{liste}$

Returnerer tallet **e** opphøyd i potens av hvert element i *Liste1*.

$e^{\text{kvadratMatrise1}} \Rightarrow \text{kvadratMatrise}$

Returnerer kvadratMatrise som er **e** opphøyd i *kvadratMatrise1*. Dette er ikke det samme som å beregne **e** opphøyd i potens av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

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

eff()

Katalog > 

$\text{eff}(\text{nominellRente}, \text{CpY}) \Rightarrow \text{verdi}$

$\text{eff}(5.75, 12)$

5.90398

Finansiell funksjon som omregner den nominelle renten *nominellRente* til en årlig effektiv rente, gitt *CpY* som antall renteperioder per år.

nominellRente må være et reelt tall, og *CpY* må være et reelt tall > 0 .

Merk: Se også **nom()**, side 103.

eigVc() (egenvektor)

eigVc(kvadratMatrise) ⇒ *matrise*

Returnerer en matrise som inneholder egenvektorer for en reell eller kompleks *kvadratMatrise*, der hver kolonne i resultatet samsvarer med en egenverdi. Merk at en egenvektor ikke er entydig; den kan skaleres av enhver konstant faktor. Egenvektorene er normalisert, dvs. at if $V = [x_1, x_2, \dots, x_n]$, then:

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

kvadratMatrise blir først balansert med likhetstransformasjoner til normene for rad og kolonne er så nær den samme verdien som mulig. *KvadratMatrisen* blir så redusert til øvre Hessenberg-form og egenvektorene beregnes via en Schur-faktorisering.

I rektangulært, kompleks format:

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

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

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

eigVl() (egenverdi)

eigVl(kvadratMatrise) ⇒ *liste*

Returnerer en liste over egenverdiene av en reell eller kompleks *kvadratMatrise*.

kvadratMatrise blir først balansert med likhetstransformasjoner til normene for rad og kolonne er så nær den samme verdien som mulig. *KvadratMatrisen* blir så redusert til øvre Hessenberg-form og egenverdiene beregnes fra den øvre Hessenberg-matrisen.

I rektangulær, kompleks format-modus:

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

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

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

If *BoolskUtr1* Then*Blokk1***Elseif** *BoolskUtr2* Then*Blokk2*

⋮

Elseif *Boolsk UtrN* Then*BlokkN***Endif**

⋮

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

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

Return 5

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

Return 3

EndIf

EndFunc

*Done***EndFor**

Se For, side 54.

EndFunc

Se Func, side 58.

EndIf

Se If, side 69.

EndLoop

Se Loop, side 90.

EndPrgm

Se Prgm, side 114.

EndTry

Se Try, side 158.

euler ()

Katalog > 

euler(*Uttr*, *Var*, *avhVar*, {*Var0*, *VarMaks*}, *avhVar0*, *VarTall* [, *eulersIntervall*]) ⇒ *matrise*

euler(*SystemAvUttr*, *Var*, *ListeMedAvhVarer*, {*Var0*, *VarMaks*}, *ListeMedAvhVarer0*, *VarIntervall* [, *eulersIntervall*]) ⇒ *matrise*

euler(*ListeMedUttr*, *Var*, *ListeMedAvhVarer*, {*Var0*, *VarMaks*}, *ListeMedAvhVarer0*, *VarIntervall* [, *eulersIntervall*]) ⇒ *matrise*

Bruker Eulers metode for å løse systemet

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

med *avhVar*(*Var0*)=*avhVar0* på intervallet [*Var0*,*VarMaks*]. Returnerer en matrise, hvor den første raden definerer verdiene i *Var* -resultatet og hvor den andre raden definerer verdien av den første løsningskomponenten ved de tilsvarende *Var* -verdiene, og så videre.

Uttr er høyre side, som definerer den ordinære differensialligningen (ODE).

SystemAvUttr er systemet på høyre side som definerer systemet av ODE-er (tilsvarende til rekkefølgen av avhengige variabler i *ListeMedAvhVarer*).

ListeMedUttr er en liste på høyre side som definerer systemet av ODE-er (tilsvarende til rekkefølgen av avhengige variabler i *ListeMedAvhVarer*).

Var er den uavhengige variabelen.

ListeMedAvhVarer er en liste over avhengige variabler.

Differensialligning:

$$y' = 0,001 \cdot y \cdot (100 - y) \text{ og } y(0) = 10$$

$$\text{euler}(0,001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1)$$

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

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

System av ligninger:

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

med $y1(0) = 2$ og $y2(0) = 5$

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

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

{*Var0*, *VarMaks*} er en liste med to elementer som forteller funksjonen at den skal integrere fra *Var0* til *VarMaks*.

ListeMedAvhVarer er en liste over startverdier for avhengige variabler.

VarIntervall er et tall som ikke er null, slik at **sign(*VarIntervall*) = sign(*VarMaks-Var0*)** og løsninger returneres ved *Var0+i·VarIntervall* for alle $i=0,1,2,\dots$ slik at *Var0+i·VarIntervall* er i [*var0*,*VarMaks*] (det kan hende at det ikke er noen løsningsverdi ved *VarMaks*).

eulersIntervall er et positivt heltall (grunninnstilt på 1) som definerer antallet euler-intervaller mellom resultatverdiene. Den faktiske tallstørrelsen som brukes ved eulers metode, er *VarIntervall/eulersIntervall*.

eval ()

Hub-meny

eval(*Uttr*) ⇒ *streng*

eval() er bare gyldig i TI-Innovator™ Hub kommandoargumentet til programmeringskommandoer **Get**, **GetStr** og **Send**. Programvaren vurderer uttrykk *Uttr* og erstatter formuleringen **eval()** med resultatet som en tegnstreng.

Argumentet *Uttr* må forenkles til et reelt tall.

Sett det blå elementet på RGB LED-skjermen til halv intensitet.

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

Tilbakestill det blå elementet til AV.

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

eval()-argumentet må forenkles til et reelt tall.

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

Programmer for å fade inn det røde elementet

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

Utfør programmet.

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

Selv om resultatet av **eval()** ikke vises, kan du se den resulterende hub-kommandostrengen etter at du har utført kommandoen ved å inspisere hvilken som helst av følgende spesielle variabler.

iostr.SendAns
iostr.GetAns
iostr.GetStrAns

Merk: Les også **Get**(side 60), **GetStr** (side 67) og **Send** (side 134).

Exit (Avslutt)

Exit

Avslutter aktuell **For**, **While**, eller **Loop**-blokk.

Exit er ikke tillatt utenfor de tre løkkestrukturene (**For**, **While**, eller **Loop**).

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

Program:

```
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()

Returnerer **e** opphøyd i *Uttr1*-potens.

Returnerer **e** opphøyd i *Verdil*-potens.

exp()

 tast

Merk: Se også **e** eksponent-sjablon, side 2.

Du kan legge inn et komplekst tall i rei θ polar form. Men bruk denne formen bare i radian-vinkelmodus; den forårsaker grunnmengdefeil i grader- eller gradian-vinkelmodus.

exp(Liste1) ⇒ *liste*

Returnerer tallet **e** opphøyd i potens av hvert element i *Liste1*.

exp(kvadratMatrise1) ⇒ *kvadratMatrise*

Returnerer kvadratMatrise som er **e** opphøyd i *kvadratMatrise1*. Dette er ikke det samme som å beregne **e** opphøyd i potens av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

expr() (uttrykk)

Katalog > 

expr (String) ⇒ *Uttrykk*

Returnerer tegnstrengen som ligger i *Streng* som et uttrykk og utfører den straks.

ExpReg

Katalog > 

ExpReg *X*, *Y* [, [*Frekv*] [, *Kategori*, *Inkluder*]]

Finner den eksponensielle regresjonen $y = a \cdot (b)^x$ for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og *Y* er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt X og Y forekommer. Standardverdien er 1. Alle elementene må være heltall ≥ 0 .

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene.

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).



Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot (b)^x$
stat.a, stat.b	Regresjonskoeffisienter
stat.r ²	Lineær determinasjonskoeffisient for transformerte data
stat.r	Korrelasjonskoeffisient for transformerte data ($x, \ln(y)$)
stat.Resid	Residualene for den eksponensielle modellen
stat.ResidTrans	Rester tilordnet ved lineær tilpasning av transformerte data
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> , og <i>Inkludert kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>inkludert kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

factor() (faktor)Katalog > 

factor(*rasjonalTall*) returnerer det rasjonale tallet faktorisert i primtall. For sammensatte tall øker behandlingstiden eksponensielt med antallet siffer i den nest største faktoren. For eksempel kan det ta mer enn en hel dag å faktorisere et heltall med 30 siffer, og å faktorisere et tall med 100 siffer kan ta mer enn et århundre.

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

Slik stopper du en beregning manuelt,

- **Grafregner:** Hold nede tasten , og trykk på  flere ganger.
- **Windows®:** Hold nede tasten **F12**, og trykk på **Enter** flere ganger.
- **Macintosh®:** Hold nede tasten **F5**, og trykk på **Enter** flere ganger.
- **iPad®:** Applikasjonen viser en ledetekst. Du kan fotsette å vente, eller avbryte.

Hvis du bare vil bestemme om et tall er et primtall, bruk **isPrime()** istedenfor. Det er mye raskere, særlig hvis *rasjonalTall* ikke er et primtall og hvis den nest største faktoren består av mer enn fem siffer.

F Cdf()Katalog > **F Cdf**

(
nedGrense
,*øvGrense*,*dfTeller*,*dfNevner*) \Rightarrow tall hvis
nedGrens og *øvGrens* er tall, *liste* hvis
nedGrens og *øvGrens* er lister

F Cdf

(
nedGrense
,*øvGrense*,*dfTeller*,*dfNevner*) \Rightarrow tall hvis
nedGrens og *øvGrens* er tall, *liste* hvis
nedGrens og *øvGrens* er lister

Beregner F fordelingssannsynligheten mellom *nedGrense* og *ovGrense* for spesifisert *dfTeller* (frihetsgrader) og *dfNevner*.

For $P(X \leq \text{ovGrens})$, set *nedGrens* = 0.

Fill (Fyll)

matriseVar må eksistere allerede.

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

listeVar må eksistere allerede.

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

FiveNumSummary

FiveNumSummary X , [*Frekv*]
[,*Kategori*,*Inkluder*]]

Gir en forkortet versjon av den 1-variabels statistiske observatoren på listen X . En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147.)

X representerer en liste med dataene.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hver korresponderende X -verdi forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende X -dataene.

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

Et tomt (åpent) element i enhver av listene X , $Frekv$ eller $Kategori$ resulterer i et åpent element for det tilsvarende elementet til alle disse listene. For mer informasjon om tomme elementer, se side 209.

Utdata-variabel	Beskrivelse
stat.MinX	Minimum av x-verdiene
stat.Q ₁ X	Første kvartil av x
stat.MedianX	Medianen av x
stat.Q ₃ X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdiene

floor() (nedre)

Returnerer det største heltallet som er \leq argumentet. Denne funksjonen er identisk med **int()**.

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

Argumentet kan være et reelt eller et komplekst tall.

floor(Liste1) \Rightarrow liste

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

floor(Matrise1) \Rightarrow matrise

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

Returnerer en liste eller matrise med nedre verdi for hvert element.

Merk: Se også **ceiling()** og **int()**.

For

For $Var, Lav, Høy [, Intervall]$

Blokk

EndFor

Utfører utsagnene i *Blokk* iterativt for hver verdi av Var , fra Lav til $Høy$, i trinn på $Intervall$.

Var må ikke være en systemvariabel.

```
Define g()=Func
  Local tempsum,step,i
  0  $\rightarrow$  tempsum
  1  $\rightarrow$  step
  For i,1,100,step
    tempsum+i  $\rightarrow$  tempsum
  EndFor
EndFunc
```

$$g() = 5050$$

Intervall kan være positiv eller negativ. Grunnverdien er 1.

Blokk kan enten være et enkelt utsagn eller en sekvens av utsagn som er adskilt med tegnet “.”.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

format()

*format*Streng er en streng og må være av formen: “F[n]”, “S[n]”, “E[n]”, “G[n][c]”, hvor [] viser alternative muligheter.

F[n]: Fast format. n er antallet siffer som vises etter desimalpunktet.

V[n]: Vitenskapelig format. n er antallet siffer som vises etter desimalpunktet.

T[n]: Teknisk format. n er antallet siffer etter det første signifikante sifferet. Eksponenten er tilpasset til et multiplum av tre, og desimalpunktet er flyttet til høyre med sifrene null, ett eller to.

G[n][c]: Samme som fast format, men skiller også sifrene til venstre for basen i grupper på tre. c spesifiserer gruppens og basens skilletegn som et komma. Hvis c er en periode, vises basen som et komma.

[Rk]: Som etterledd bak noen av spesifikantene over kan basemerket Rc tilføyes, der hvor c er et enkelt tegn som spesifiserer hva som erstatter komma.

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() (funksjonsdel)

fPart(*Utt1*) ⇒ *uttrykk*

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

fPart(*Liste1*) ⇒ *liste*

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

fPart(*Matrise*1)⇒*matrise*

Returnerer brøk-delen i argumentet.

For en liste eller matrise, returneres brøk-delene i elementene.

Argumentet kan være et reelt eller et komplekst tall.

FPdf()

FPdf(*XVerdi*,*dfTeller*,*dfNevner*)⇒*tall* hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste

FPdf(*XVerdi*,*dfTeller*,*dfNevner*)⇒*tall* hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste

Beregner F fordelingssannsynligheten mellom *XVerdi* for den spesifiserte *dfTeller* (grader av frihet) og *dfNevner*.

freqTable▶liste()

freqTable▶liste

(*Liste*1,*frekvHeltallListe*)⇒*liste*

Returnerer en liste som inneholder elementene fra *Liste*1 utvidet i henhold til frekvensene i *frekvHeltallListe*.

Denne funksjonen kan brukes til å generere en frekvenstabell for applikasjonen Data og statistikk.

*Liste*1 kan være enhver gyldig liste.

frekvHeltallListe må ha samme dimensjon som *Liste*1 og kun inneholde ikke-negative heltallselementer. Hvert element angir hvor mange ganger det korresponderende *Liste*1-elementet skal gjentas i resultatlisten. En verdi lik null utelater det korresponderende *Liste*1-elementet.

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **freqTable@>list (...)**.

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}

Tomme (åpne) elementer ignorerer. For mer informasjon om tomme elementer, se side 209.

frequency() (frekvens)

frequency(*Liste1*,*stolperListe*)⇒*liste*

Returnerer en liste som inneholder antallet elementer i *Liste1*. Antallet er basert på områder (stolper) som du definerer i *stolperListe*.

Hvis *stolperListe* er {b(1), b(2), ..., b(n)}, er de spesifiserte områdene { $? \leq b(1)$, $b(1) < ? \leq b(2)$, ..., $b(n-1) < ? \leq b(n)$, $b(n) > ?$ }. Den resulterende listen er ett element lenger enn *stolperListe*.

Hvert element av resultatet samsvarer med antallet elementer fra *Liste1* som er i området for den stolpen. Uttrykt med begrep fra **countIf**()-funksjonen er resultatet { countIf(liste, $? \leq b(1)$), countIf(liste, $b(1) < ? \leq b(2)$), ..., countIf(liste, $b(n-1) < ? \leq b(n)$), countIf(liste, $b(n) > ?$)}.

Elementer fra *Liste1* som ikke kan "plasseres i en stolpe" ignorerer. Tomme (åpne) elementer ignoreres også. For mer informasjon om tomme elementer, se side 209.

I applikasjonen Lister og regneark kan du bruke et celleområde istedenfor begge argumentene.

Merk: Se også **countIf**() , side 30.

<i>dataliste</i> ={ 1,2,e,3,π,4,5,6,"hello",7 }	
{ 1,2,2.71828,3,3.14159,4,5,6,"hello",7 }	
frequency(<i>dataliste</i> ,{2.5,4.5})	{2,4,3}

Forklaring til resultat:

2 elementer fra *Dataliste* er $\leq 2,5$

4 elementer fra *Dataliste* er $> 2,5$ og $\leq 4,5$

3 elementer fra *Dataliste* er $> 4,5$

Elementet "hallo" er en streng og kan ikke plasseres i nopen av de definerte stolpene.

F Test_2Samp (2_utvalg F test)

FTest_2Samp*Liste1*,*Liste2*[,*Frekv1*[,*Frekv2* [,*Hypot*]]]

FTest_2Samp *Liste1*,*Liste2*[,*Frekv1*[,*Frekv2* [,*Hypot*]]]

(Dataliste inndata)

FTest_2Samp $sx1, n1, sx2, n2[, Hypot]$

FTest_2Samp $sx1, n1, sx2, n2[, Hypot]$

(Summering statistikk inndata)

Utfører en to-utvalgs F test. En oversikt over resultatene lagres i *stat.results*-variabelen (side 147).

eller $H_a: \sigma_1 > \sigma_2$, sett *Hypoth*>0

For $H_a: \sigma_1 \neq \sigma_2$ (standard), sett *Hypoth* =0

For $H_a: \sigma_1 < \sigma_2$, sett *Hypoth*<0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" på side 209.

Utdata-variabel	Beskrivelse
stat.F	Beregnet \hat{U} -statistikk for datasekvensen
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.dfNumer	frihetsgrad for teller = $n1-1$
stat.dfDenom	frihetsgrad for nevner = $n2-1$
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i <i>Liste 1</i> og <i>Liste 2</i>
stat.x1_bar stat.x2_bar	Utvalgets gjennomsnitt av datasekvenser i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Utvalgenes størrelse

Func (Funk)

Func

Blokk

EndFunc

Sjablon for oppretting av brukerdefinert funksjon.

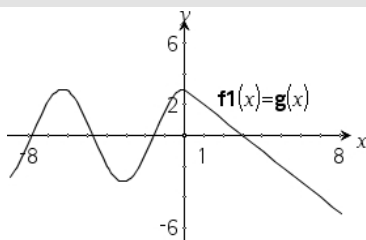
Definere en sammensatt funksjon:

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

Resultat av grafisk fremstilling g(x)

Blokk kan være ett enkelt utsagn, en rekke utsagn adskilt med “:”-tegnet, eller en rekke med utsagn på separate linjer. Funksjonen kan bruke **Returner**-kommandoen for å returnere et spesifikt resultat.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkhåndboken.



G

gcd() (største felles divisor)

gcd(*Tall1*, *Tall2*) \Rightarrow *uttrykk*

gcd(18,33)

3

Returnerer største felles divisor for de to argumentene. **gcd** av to brøker er **gcd** av tellerne dividert med **lcm** av nevnerne.

I modusen **Auto** eller **Tilnærmet** er **gcd** av brøkens flytende desimalpunktall 1,0.

gcd(*Liste1*, *Liste2*) \Rightarrow *liste*

gcd({12,14,16},{9,7,5})

{3,7,1}

Returnerer største felles divisorer av samsvarende deler i *Liste1* og *Liste2*.

gcd(*Matrise1*, *Matrise2*) \Rightarrow *matrise*

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

$\begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}$

Returnerer største felles divisorer av samsvarende deler i *Matrise1* og *Matrise2*.

geomCdf()

geomCdf(*p*,*nedreGrense*,*øvreGrense*) \Rightarrow *tall*
hvis *nedreGrense* og *øvreGrense* er tall,
liste hvis *nedreGrense* og *øvreGrense* er lister

geomCdf(*p*,*øvreGrense*)for $P(1 \leq X \leq \textit{øvreGrense}) \Rightarrow$ *tall* hvis *øvreGrense* er et tall, *liste* hvis *øvreGrense* er en liste

Beregner en kumulativ geometrisk sannsynlighet fra *nedreGrense* til *øvreGrense* med den spesifiserte sannsynligheten for suksess *p*.

For $P(X \leq \text{øvreGrense})$, sett *nedreGrense* = 1.

geomPdf()

geomPdf(*p*, *XVerdi*) \Rightarrow *tall* hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste

Beregner en sannsynlighet ved *XVerdi*, antall forsøk før første suksess inntreffer, for diskret geometrisk fordeling med spesifisert suksess-sannsynligheten *p*.

Get

Hub-meny

Get[*ledetekstStreng*,] *var*[, *statusVar*]

Get[*ledetekstStreng*,] *funk*(*arg1*, ...*argn*)[, *statusVar*]

Programmeringskommando: Henter en verdi fra en tilkoblet TI-Innovator™ Hub og tildeler verdien til variabel *var*.

Verdien må etterspørres:

- På forhånd, gjennom en **Send «LES ...»**-kommando.
–eller–
- Ved å innlemme en «**LES ...**»-forespørsel som det alternative *ledetekstStreng*-argumentet. Denne metoden lar deg bruke en enkel kommando for å etterspørre verdien og hente den.

Implisitt forenkling finner sted. For eksempel tolkes en mottatt streng som «123» som en numerisk verdi. For å bevare strengen, bruker du **GetStr** i stedet for **Get**.

Eksempel: Etterspør nåværende verdi fra hubbens innebygde lysnivåsensor. Bruk **Get** for å hente verdien og tildele den til variabelen *lysver*.

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

Innlem LES-forespørslen i **Hent**-kommandoen.

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

Hvis du inkluderer det valgfrie argumentet *statusVar*, tilordnes det en verdi basert på om operasjonen lyktes eller ikke. En verdi på null betyr at ingen data ble mottatt.

I den andre syntaksen lar argumentet *funk()* et program lagre den mottatte strengen som en funksjonsdefinisjon. Denne syntaksen arbeider som om programmet utførte kommandoen:

Definer *funk(arg1, ...argn) = mottatt streng*

Programmet kan så bruke den definerte funksjonen *funk()*.

Merk: Du kan bruke **Get**-kommandoen i et brukerdefinert program, men ikke i en funksjon.

Merk: Se også **GetStr**, side 67 og **Send**, side 134.

getDenom() (lesNevner)

[Katalog >](#)

Omformer argumentet inn til et uttrykk som har en redusert felles nevner og returnerer så uttrykkets nevner.

getKey()

[Katalog >](#)

getKey([0|1]) ⇒ returnString

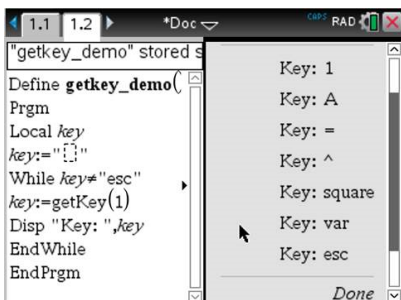
Beskrivelse: **getKey()** – tillater at et TI-Basic-program henter tastaturinndata – grafregner, stasjonær PC og emulator på skrivebordet.

Eksempel:

- `keypressed := getKey()` returnerer en tast eller en tom streng hvis ingen tast er trykket ned. Dette oppkallet returneres umiddelbart.
- `keypressed := getKey(1)` venter til en tast trykkes ned. Dette

getKey()

Eksempel:



oppkallet setter utførelsen av programmet på pause til en tast trykkes ned.

Håndtering av tastetrykk:

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
Esc	Esc	"esc"
Styreplate – klikk oppe	Ikke relevant	"opp"
På	Ikke relevant	"start"
Kladdeapp	Ikke relevant	"kladdeblokk"
Styreplate – venstreklikk	Ikke relevant	"venstre"
Styreplate – midtklikk	Ikke relevant	"midtre"
Styreplate – høyreklikk	Ikke relevant	"høyre"
Dok	Ikke relevant	"dok"
Kategori	Kategori	"kategori"
Styreplate – klikk nede	Piltast ned	"ned"
Meny	Ikke relevant	"meny"
Ctrl	Ctrl	ingen retur
Skift	Skift	ingen retur
Variabel	Ikke relevant	"var"
Del	Ikke relevant	"del"
=	=	"="
trigonometri	Ikke relevant	"trigonometri"
0 til og med 9	0-9	"0" ... "9"
Sjabloner	Ikke relevant	"sjablon"
Katalog	Ikke relevant	"kat"
^	^	"^"
X^2	Ikke relevant	"kvadrat"

Håndholdt enhet / emulator tast	Stasjonær PC	Returverdi
/ (divisjonstast)	/	"/"
* (multiplikasjonstast)	*	"*"
e^x	Ikke relevant	"eksp"
10^x	Ikke relevant	"10potens"
+	+	"+"
-	-	"-"
(("("
))	")"
.	.	". "
(-)	Ikke relevant	"-" (negativ-tegn)
Enter	Enter	"enter"
ee	Ikke relevant	"E" (vitenskapelig notasjon E)
a-z	a-z	alpha = bokstav trykket ned (liten bokstav) ("a"- "z")
skift a-z	skift a-z	alpha = bokstav trykket ned "A"- "Z"
		Merk: ctrl-skift brukes som Caps Lock
?!	Ikke relevant	"?!"
pi	Ikke relevant	"pi"
Flagg	Ikke relevant	ingen retur
,	,	","
Returner	Ikke relevant	"returner"
mellomrom	mellomrom	" " (mellomrom)
Utilgjengelig	Spesialtegn som f.eks. @,!,^ osv.	Tegnet er returnert
Ikke relevant	Funksjonstaster	Ingen returnerte tegn
Ikke relevant	Spesielle	Ingen returnerte tegn

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
	skrivebordskontrolltaster	
Utilgjengelig	Andre skrivebordstaster som ikke er tilgjengelige på kalkulatoren mens getKey() venter på et tastetrykk. ({, }, ;, :, ...)	Det samme tegnet du får i Notat (ikke i en matematikkboks)

Merk: Legg merke til at **getKey()** i et program endrer hvordan hendelser behandles av systemet. Noen av disse beskrives nedenfor.

Avslutte et program og behandle en hendelse – På samme måte som om brukeren skulle avslutte et program ved å trykke på **PÅ**-tasten

"**Støtte**" nedenfor betyr – Systemet fungerer som forventet – programmet fortsetter å kjøre.

Hendelse	Enhet	Stasjonær PC – TI-Nspire™ Student Software
Hurtigspørring	Avslutte et program, behandle en hendelse	Samme som for grafregner (bare TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)
Ekstern filbehandling (Inkl. sende filen 'Exit Press 2 Test' fra en annen grafregner eller skrivebords-grafregner)	Avslutte et program, behandle en hendelse	Samme som for en grafregner. (bare TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)
Avslutt klasse	Avslutte et program, behandle en hendelse	Brukerstøtte (bare TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software)

Hendelse	Enhet	Stasjonær PC – TI-Nspire™ Alle versjoner
TI-Innovator™ Hub koble til / koble fra	Støtte – kan sende kommandoer til TI-Innovator™ Hub. Etter at du har avsluttet programmet, fungerer TI-Innovator™ Hub fremdeles med grafregneren.	Samme som for en grafregner

getLangInfo()

katalog > 

getLangInfo() ⇒ *streng*

`getLangInfo()`

"en"

Returnerer en streng som svarer til kortnavnet på det aktive språket. Du kan for eksempel bruke den i et program eller en funksjon for å finne aktivt språk.

Engelsk = "en"

Dansk = "da"

Tysk = "de"

Finsk = "fi"

Fransk = "fr"

Italiensk = "it"

Nederlandsk = "nl"

Belgisk nederlandsk = "nl_BE"

Norsk = "no"

Portugisisk = "pt"

Spansk = "es"

Svensk = "sv"

getLockInfo()

Katalog > 

getLockInfo(Var) ⇒ *verdi*

`a:=65`

65

Returnerer aktuell låst/opplåst status for variabel *Var*.

`Lock a`

Done

verdi = 0: *Var* er låst opp eller eksisterer ikke.

`getLockInfo(a)`

1

verdi = 1: *Var* er låst opp og kan ikke modifiseres eller slettes.

`a:=75`

"Error: Variable is locked."

`DelVar a`

"Error: Variable is locked."

Se **Lock**, side 87, og **unLock**, side 164.

`Unlock a`

Done

`a:=75`

75

`DelVar a`

Done

GetMode() (lesModus)

Katalog > 

GetMode(ModusNavnHeltall) ⇒ *verdi*

GetMode(0) ⇒ *liste*

GetMode(*ModusNavnHeltall*) returnerer en verdi som representerer aktuell innstilling av *ModusNavnHeltall*-modus.

GetMode(0) returnerer en liste som inneholder tallpar. Hvert par består av et modusheltall og et innstillingsheltall.

For en opplisting av modusene og deres innstillinger, referer til tabellen under.

Hvis du lagrer innstillingene med **GetMode(0)** → *var*, kan du bruke **GetMode(*var*)** i en funksjon eller et program for midlertidig å gjenopprette innstillingene kun innenfor utføringen av funksjonen eller programmet. Se **GetMode()**, side 137.

Modus Navn	Modus Heltall	Innstille heltall
Vis sifre	1	1=Flytende, 2=Flytende1, 3=Flytende2, 4=Flytende3, 5=Flytende4, 6=Flytende5, 7=Flytende6, 8=Flytende7, 9=Flytende8, 10=Flytende9, 11=Flytende10, 12=Flytende11, 13=Flytende12, 14=Fast0, 15=Fast1, 16=Fast2, 17=Fast3, 18=Fast4, 19=Fast5, 20=Fast6, 21=Fast7, 22=Fast8, 23=Fast9, 24=Fast10, 25=Fast11, 26=Fast12
Vinkel	2	1=Radian, 2=Grader, 3=Gradian
Eksponensielt format	3	1=Normal, 2=Vitenskapelig, 3=Teknisk
Reell eller kompleks	4	1=Reell, 2=Rektangulær, 3=Polar
Auto eller tilnærm.	5	1=Auto, 2=Tilnærmet
Vektorformat	6	1=Rektangulær, 2=Sylindrisk, 3=Sfærisk
Grunntall	7	1=Desimal, 2=Heks, 3=Binær

getNum() (lesTeller)

Omformert argumentet til et uttrykk som har en redusert felles nevner og returnerer så uttrykkets teller.

GetStr[*ledetekstStreng*,] *var*[, *statusVar*]

For eksempler, se **Get**.

GetStr[*ledetekstStreng*,] *funk*(*arg1*, ...*argn*)
[, *statusVar*]

Programmeringskommando: Virker på nøyaktig samme måte som kommandoen **Get**, bortsett fra at mottatt verdi alltid tolkes som en streng. I motsetning tolker kommandoen **Get** svaret som et uttrykk, med mindre det er satt i anførselstegn ("").

Merk: Se også **Get**, side 60 og **Send**, side 134.

getType()

Katalog > 

getType(*var*) ⇒ *streng*

Returnerer en streng som angir dataens typevariabel *var*.

Hvis *var* ikke er definert, returnerer strengen "INGEN".

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

getVarInfo()

katalog > 

getVarInfo() ⇒ *matrise* eller *streng*

getVarInfo
(*BibliotekNavnStreng*) ⇒ *matrise* eller *streng*

getVarInfo() returnerer en matrise med informasjon (variabelnavn, type, bibliotektilgjengelighet og låst/opplåst status) for alle variabler og biblioteksobjekter som er definert i den aktuelle oppgaven.

Hvis ingen variabler er definert, returnerer **getVarInfo**() strengen "INGEN".

getVarInfo()	"NONE"												
Define $x=5$	Done												
Lock x	Done												
Define LibPriv $y=\{1,2,3\}$	Done												
Define LibPub $z(x)=3 \cdot x^2 - x$	Done												
getVarInfo()	<table border="1"> <tbody> <tr> <td>x</td> <td>"NUM"</td> <td>"{ }"</td> <td>1</td> </tr> <tr> <td>y</td> <td>"LIST"</td> <td>"LibPriv "</td> <td>0</td> </tr> <tr> <td>z</td> <td>"FUNC"</td> <td>"LibPub "</td> <td>0</td> </tr> </tbody> </table>	x	"NUM"	"{ }"	1	y	"LIST"	"LibPriv "	0	z	"FUNC"	"LibPub "	0
x	"NUM"	"{ }"	1										
y	"LIST"	"LibPriv "	0										
z	"FUNC"	"LibPub "	0										
getVarInfo(<i>tmp3</i>)	"Error: Argument must be a string"												
getVarInfo("tmp3")	[<i>volcy12</i> "NONE" "LibPub " 0]												

getVarInfo

(*BibliotekNavnStreng*)returnerer en matrise med informasjon for alle bibliotekobjektene som er definert i biblioteket *BibliotekNavnStreng*. *BibliotekNavnStreng* må være en streng (tekst omsluttet av anførselstegn) eller en strengvariabel.

Hvis biblioteket *BibliotekNavnStreng* ikke finnes, oppstår det en feil.

Se for eksempel til venstre, der resultatet av **getVarInfo()** tilordnes variabelen *vs*. Hvis du forsøker å vise rad 2 eller 3 av *vs*, returneres en "Ugyldig liste eller matrise"-feil, siden minst ett av elementene i de radene (for eksempel variabel *b*) reevalueres til en matrise.

Denne feilen kan også oppstå når du bruker *Ans* til å reevaluere et **getVarInfo()**-resultat.

Systemet viser ovenstående feil fordi den gjeldende versjonen av programvaren ikke støtter en generalisert matrisestruktur der et element kan være enten en matrise eller en liste.

<i>a</i> :=1	1												
<i>b</i> :=[1 2]	[1 2]												
<i>c</i> :=[1 3 7]	[1 3 7]												
<i>vs</i> :=getVarInfo()	<table border="1"> <tr> <td><i>a</i></td> <td>"NUM"</td> <td>"[]"</td> <td>0</td> </tr> <tr> <td><i>b</i></td> <td>"MAT"</td> <td>"[]"</td> <td>0</td> </tr> <tr> <td><i>c</i></td> <td>"MAT"</td> <td>"[]"</td> <td>0</td> </tr> </table>	<i>a</i>	"NUM"	"[]"	0	<i>b</i>	"MAT"	"[]"	0	<i>c</i>	"MAT"	"[]"	0
<i>a</i>	"NUM"	"[]"	0										
<i>b</i>	"MAT"	"[]"	0										
<i>c</i>	"MAT"	"[]"	0										
<i>vs</i> [1]	[1 "NUM" "[]" 0]												
<i>vs</i> [1,1]	1												
<i>vs</i> [2]	"Error: Invalid list or matrix"												
<i>vs</i> [2,1]	[1 2]												

Goto (Gåtil)**Goto etikettNavn**

Overfører kontroll til navnet *etikettNavn*.

etikettNavn må være definert i samme funksjon med en **Lbl**-instruksjon.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

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

►Grad

Katalog > 

Uttr1 ► **Grad** ⇒ *Uttrykk*

I Grader-vinkelmodus:

Omregner *Uttr1* til gradian vinkelmåling.

(1.5) ► Grad $(1.66667)^g$

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>**Grad**.

I Radian-vinkelmodus:

(1.5) ► Grad $(95.493)^g$

/

identity()

Katalog > 

identity(*Heltall*) ⇒ *matrise*

identity(4)

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Returnerer identitetsmatrisen med dimensjonen *Heltall*.

Heltallet må være et positivt heltall.

Hvis

Katalog > 

Hvis *BooleanExpr*
Utsagn

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

Hvis *BooleanExpr*, så
Blokk

OgHvis

$g(-2)$ 4

Hvis *BooleanExpr* behandles som sann, utføres det enkle utsagnet *Utsagn* eller blokken av utsagn *Blokk* før utførelsen fortsetter.

Hvis *BooleanExpr* behandles som usann, fortsettes utførelsen uten å utføre utsagnet eller blokken av utsagn.

Blokk kan enten være et enkelt utsagn eller en sekvens av utsagn som er adskilt med tegnet «:».

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

Hvis *BooleanExpr*, så
Blokk1

hvis ikke
Blokk2

OgHvis

Hvis *BooleanExpr* behandles som sann, utføres *Blokk1* og utelater så *Blokk2*.

Hvis *BooleanExpr* behandles som usann, utelates *Blokk1*, men *Blokk2* utføres.

Blokk1 og *Blokk2* kan være et enkelt utsagn.

Hvis *BooleanExpr1*, så
Blokk1

EllersHvis *BooleanExpr2*, så
Blokk2

:

EllersHvis *BooleanExprN*, så
BlokkN

OgHvis

Tillater forgreining. Hvis *BooleanExpr1* behandles som sann, utføres *Blokk1*.

Hvis *BooleanExpr1* behandles som usann, behandles *BooleanExpr2*, og så videre.

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

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

Define $g(x)$ =Func	
If $x < 5$ Then	
Return 5	
ElseIf $x > 5$ and $x < 0$ Then	
Return $-x$	
ElseIf $x \geq 0$ and $x \neq 10$ Then	
Return x	
ElseIf $x = 10$ Then	
Return 3	
EndIf	
EndFunc	

	Done
$g(-4)$	4
$g(10)$	3

ifFn()

ifFn(*BooleanExpr*, *Value_If_true*
[, *Value_If_false* [, *Value_If_unknown*]])
 \Rightarrow uttrykk, liste eller matrise

Behandler det boolske uttrykket *BooleanExpr* (eller hvert element fra *BooleanExpr*) og produserer et resultat basert på følgende regler:

- *BooleanExpr* kan teste en enkel verdi, en liste eller en matrise.
- Hvis et element av *BooleanExpr* behandles som sann, returneres det samsvarende elementet fra *Value_If_true*.
- Hvis et element av *BooleanExpr*

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

Testverdi på **1** er mindre enn 2,5, så det er samsvarende

Value_If_True-element av **5** kopieres til resultatlisten.

Testverdi på **2** er mindre enn 2,5, så det er samsvarende

Value_If_True-element av **6** kopieres til resultatlisten.

behandles som usann, returneres det samsvarende elementet fra *Value_If_false*. Hvis du utelater *Value_If_false*, returneres undef.

- Hvis et element av *BooleanExpr* hverken er sant eller usant, returneres det samsvarende elementet *Value_If_unknown*. Hvis du utelater *Value_If_unknown*, returneres undef.
- Hvis det andre, tredje eller fjerde argumentet i **ifFn()**-funksjonen et enkelt uttrykk, brukes det boolske uttrykket i hver posisjon i *BooleanExpr*.

Merk: Hvis det forenklede utsagnet *BooleanExpr* involverer en liste eller matrise, må alle andre liste- eller matriseargumenter ha samme dimensjoner, og resultatet ha samme dimensjoner.

Testverdi på **3** er ikke mindre enn 2,5, så det samsvarende *Value_If_False*-elementet på **10** kopieres til resultatlisten.

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

Value_If_true er en enkel verdi og samsvarer med hvilken som helst valgt posisjon.

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

Value_If_false er ikke spesifisert. Udef brukes.

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

Et element velges fra *Value_If_true*. Et element velges fra *Value_If_unknown*.

imag()

Returnerer den imaginære delen av argumentet.

imag(List I) ⇒ *liste*

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

Returnerer en liste over de imaginære delene av elementene.

imag(Matrix I) ⇒ *matrise*

Returnerer en matrise over de imaginære delene av elementene.

Indireksjon

inString()

Katalog > 

inString(*srcString*, *subString*[, *Start*]) ⇒ *heltall*

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

Returnerer tegnposisjonen i strengen *srcString* der første forekomst av strengen *subString* begynner.

Start, hvis det er inkludert, spesifiserer tegnposisjonen innenfor *srcString* der søket starter. Standard = 1 (det første tegnet i *srcString*).

Hvis *srcString* ikke inneholder *subString* eller *Start* er > lengden av *srcString*, returneres null.

int()

Katalog > 

int(*List1*) ⇒ *liste*

int(*Matrix1*) ⇒ *matrise*

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

Returnerer det største heltallet som er mindre enn eller lik argumentet. Denne funksjonen er identisk med **floor()**.

Argumentet kan være et reelt eller et komplekst tall.

For en liste eller matrise, returneres det største heltallet for hvert element.

intDiv()

Katalog > 

intDiv(*Number1*, *Number2*) ⇒ *heltall*

intDiv(*List1*, *List2*) ⇒ *liste*

intDiv(*Matrix1*, *Matrix2*) ⇒ *matrise*

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

Returnerer heltallsdelen med fortegn av (*Number1* ÷ *Number2*).

For lister og matriser, returneres heltallsdelen med fortegn av (argument 1 ÷ argument 2) for hvert elementpar.

interpoler ()

Katalog >

interpoler($xValue$, $xList$, $yList$, $yPrimeList$) \Rightarrow *liste*

Denne funksjonen gjør følgende:

Gitt $xList$, $yList=f(xList)$ og $yPrimeList=f'(xList)$ for en ukjent funksjon f , brukes en kubisk interpolant for å tilnærme funksjonen f ved $xValue$. Det antas at $xList$ er en liste over monotont stigende eller synkende tall, men denne funksjonen kan returnere en verdi selv om den ikke er det. Denne funksjonen går gjennom $xList$ og ser etter et intervall [$xList[i]$, $xList[i+1]$] som inneholder $xValue$. Hvis den finner et slikt intervall, returnerer den en interpolert verdi for $f(xValue)$, ellers returnerer den **undef**.

$xList$, $yList$ og $yPrimeList$ må være av lik dimensjon ≥ 2 og inneholde uttrykk som forenkles til tall.

Differensialligning:
 $y'=-3\cdot y+6\cdot t+5$ og $y(0)=5$

```
rk:=rk23(-3*y+6*t+5,y,{0,10},5,1)
┌0. 1. 2. 3. 4.
└5. 3.19499 5.00394 6.99957 9.00593 10.00000
```

For å se hele resultatet, trykk på og bruk så og for å bevege markøren.

Bruk den interpolerte() funksjonen for å beregne funksjonens verdier for x-verdilisten:

```
xvaluelist:=seq(i,i,0,10,0.5)
┌0,0,5,1,1,5,2,2,5,3,3,5,4,4,5,5,5,5,6,6,5,5
└0,0,5,1,1,5,2,2,5,3,3,5,4,4,5,5,5,5,6,6,5,5
xlist:=mat▶list(rk[1])
┌0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.
└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
└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
└-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
└5.,2.67062,3.19499,4.02782,5.00394,6.00011
```

invχ²()

Katalog >

invχ²(*Area*,*df*)

invChi2(*Area*,*df*)

Beregner invers kumulativ χ^2 (chi-kvadrat) sannsynlighetsfunksjon spesifisert av frihetsgrad, *df* for et gitt *Område* under kurven.

invF()

Katalog >

invF(*Area*,*dfNumer*,*dfDenom*)

invF(*Area*,*dfNumer*,*dfDenom*)

Beregner invers kumulativ F-fordelingsfunksjon spesifisert av *dfNumer* og *dfDenom* for et gitt *Område* under kurven

invBinom

(CumulativeProb, NumTrials, Prob, OutputForm) ⇒ skalar eller matrise

Invers binomial. Gitt antall forsøk (*NumTrials*) og sannsynligheten for å lykkes for hvert forsøk (*Prob*). Denne funksjonen returnerer minimum antall suksesser, *k*, slik at verdien, *k*, er større eller lik den oppgitte kumulative sannsynligheten (*CumulativeProb*).

OutputForm=0 viser resultat som en skalar (standard).

OutputForm=1 viser resultat som en matrise.

Eksempel: Mary og Kevin spiller med terninger. Mary skal gjette maksimalt antall ganger 6 vises på 30 kast. Hvis 6 vises så mange eller færre ganger, vinner Mary. I tillegg vinner hun mer jo mindre tallet hun gjetter er. Hva er det minste tallet Mary kan gjette hvis hun ønsker en sannsynlighet for å vinne som er større enn 77 %?

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

invBinomN()

invBinomN(CumulativeProb, Prob, NumSuccess, OutputForm) ⇒ skalar eller matrise

Invers binomial med hensyn på *N*. Gitt sannsynligheten for å lykkes med hvert forsøk (*Prob*), og antall suksesser (*NumSuccess*), returnerer denne funksjonen minimum antall forsøk, *N*, slik at verdien, *N*, er mindre eller lik den kumulative sannsynligheten (*CumulativeProb*).

OutputForm=0 viser resultat som en skalar (standard).

OutputForm=1 viser resultat som en matrise.

Eksempel: Monique øver på målskudd for nettbull. Fra erfaring vet hun at det er 70 % sjanse for at hun treffer med hvilket som helst skudd. Hun har tenkt å holde på til hun skårer 50 mål. Hvor mange skudd må hun forsøke for å sikre at sannsynligheten for å treffe med minst 50 skudd er mer enn 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()

invNorm(Area[, μ[, σ]])

Beregner den inverse, kumulative normale fordelingsfunksjonen for et gitt *område* under den normale fordelingskurven som er spesifisert av μ og σ .

invnt(*Area,df*)

Beregner invers kumulativ sannsynlighetsfunksjon for student-t spesifisert av frihetsgrad, *df* for et gitt *Område* under kurven.

iPart()

iPart(*Number*) ⇒ *heltall*

iPart(*List1*) ⇒ *liste*

iPart(*Matrix1*) ⇒ *matrise*

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

Returnerer heltallsdelen av argumentet.

For lister og matriser, returnerer heltallsdelen for hvert element.

Argumentet kan være et reelt eller et komplekst tall.

irr()

irr(*CF0,CFList* [,*CFFreq*]) ⇒ *verdi*

Finansiell funksjon som beregner internrente av retur av en investering.

CF0 er kontantstrømmen ved start kl. 0. Den må være et reelt tall.

CFList er en liste over kontantstrømbeløpene etter den innledende kontantstrømmen *CF0*.

CFFreq er en valgfri liste der hvert element spesifiserer frekvensen av forekomsten for et gruppert (etterfølgende) kontantstrømbeløp, som er det tilsvarende elementet til *CFList*. Standarden er 1. Hvis du legger inn verdier, må dette være positive heltall < 10 000.

Merk: Se også **mirr()**, side 96.

$list1 := \{6000, -8000, 2000, -3000\}$	$\{6000, -8000, 2000, -3000\}$
$list2 := \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$irr(5000, list1, list2)$	-4.64484

isPrime()

Katalog > 

isPrime(*Number*) ⇒ *Boolsk konstant uttrykk*

Returnerer sann eller usann for å vise om *tall* er et helt tall ≥ 2 som bare er delelig med seg selv og 1.

Hvis *Tall* består av mer enn 306 siffer og ikke inneholder noen faktorer, viser ≤ 1021 , **isPrime**(*Number*) en feilmelding

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

isPrime(5)	true
isPrime(6)	false

Funksjon for å finne det neste primtallet etter et spesifisert tall:

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

isVoid()

Katalog > 

isVoid(*Var*) ⇒ *Boolsk konstant uttrykk*
isVoid(*Expr*) ⇒ *Boolsk konstant uttrykk*
isVoid(*List*) ⇒ *liste over boolske konstante uttrykk*

Returnerer sann eller usann for å vise om utsagnet er en åpen (tom) datatype.

For mer informasjon om tomme elementer, se side 209.

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

L

Lbl (Nvn)

Katalog > 

Lbl *etikettNavn*

Definerer en etikett med navnet *etikettNavn* innenfor en funksjon.

Du kan bruke en **Goto** *etikettNavn* - instruksjon for å overføre kontroll til den instruksjonen som umiddelbart følger etter etiketten.

etikettNavn må følge de samme reglene for navn som gjelder for variabelnavn.

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

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

lcm() (mfm)

lcm(Tall1, Tall2) ⇒ uttrykk

lcm(6,9)	18
----------	----

lcm(Liste1, Liste2) ⇒ liste

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

lcm(Matrise1, Matrise2) ⇒ matrise

Returnerer minste felles multiplum av de to argumentene. **lcm** av to brøker er **lcm** av tellerne dividert med **gcd** av nevnerne. **lcm** av brøk som består av flytende desimalpunktall er produktet av teller og nevner.

For to lister eller matriser, returnerer minste felles multiplum for samsvarende elementer.

left() (venstre)

left(kildeStreng[, Num]) ⇒ streng

left("Hello", 2)	"He"
------------------	------

Returnerer de *Num*-tegnene som ligger lengst til venstre i tegnstrengen *kildeStreng*.

Hvis du utelater *Num*, returneres alle i *kildeStreng*.

left(Liste1[, Num]) ⇒ liste

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

Returnerer de *Num*-elementene som ligger lengst til venstre i *Liste1*.

Hvis du utelater *Num*, returneres alle elementer i *Liste1*.

left(Sammenlikning) ⇒ Uttrykk

left(x < 3)	x
-------------	---

Returnerer venstre side av en ligning eller ulikhet.

libShortcut(*BibliotekNavnStreng*,
HurtigtastNavnStreng,
BiblPrivMerke) \Rightarrow liste av variabler

Oppretter en variabelgruppe i den gjeldende oppgaven som inneholder referanser til alle objektene i det angitte bibliotekdokumentet *bibliotekNavnStreng*. Legger også gruppelemmene til i Variablermenyen. Deretter kan du referere til hvert objekt ved å bruke dets *HurtigtastNavnStreng*.

Sett *BiblPrivMerke*=0 hvis du skal ekskludere private bibliotekobjekter (standard)

Sett *BiblPrivMerke*=1 hvis du skal inkludere private bibliotekobjekter

Hvis du skal kopiere en variabelgruppe, se **CopyVar** (side 25).

Hvis du skal slette en variabelgruppe, se **DelVar** (side 39).

Dette eksemplet forutsetter et riktig lagret og oppdatert bibliotekdokument med navnet **linalg2** som inneholder objekter definert som *clearmat*, *gauss1*, og *gauss2*.

```
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 (lineær regresjon)

LinRegBx *X*, *Y*, [*Frekv*], [*Kategori*, *Inkluder*]

Finner den lineære regresjonen $y = a + b \cdot x$ for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 147.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og *Y* er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende *X* og *Y*-dataene.

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a+b \cdot x$
stat.a, stat.b	Regresjonskoeffisienter
stat.r ²	Determinasjonskoeffisient
stat.r	Korrelasjonskoeffisient
stat.Rest	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

LinRegMx (lineær regresjon)

LinRegMx *X*, *Y* [, *Frekv* [, *Kategori*, *Inkluder*]]

Finner den lineære regresjonen $y = m \cdot x + b$ for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (side 147.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og *Y* er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene.

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $m \cdot x + b$
stat.m, stat.b	Regresjonskoeffisienter
stat.r ²	Determinasjonskoeffisient
stat.r	Korrelasjonskoeffisient
stat.Rest	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede X -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede Y -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

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

For stigningstall. Beregner et konfidensintervall med konfidensnivå C for stigningstallet.

LinRegtIntervals $X, Y, F[, 1, Xverd[, CNivå]]$

For respons. Beregner en predikert y -verdi, et prediksjonsintervall med nivå C for én enkelt observasjon, og et konfidensintervall med nivå C for den gjennomsnittlige responsen.

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147.)

Alle listene må ha samme dimensjon.

X og Y er lister av uavhengige og avhengige variabler.

F er en valgfri liste over frekvensverdier. Hvert element i F spesifiserer frekvensen av forekomst for hvert tilsvarende X og Y datapunkt. Standardverdien er 1. Alle elementene må være heltall ≥ 0 .

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a+b \cdot x$
stat.a, stat.b	Regresjonskoeffisienter
stat.gf	Frihetsgrader
stat.r ²	Determinasjonskoeffisient
stat.r	Korrelasjonskoeffisient
stat.Resid	Residualene fra regresjonen

Gjelder kun stigningstall

Utdata-variabel	Beskrivelse
[stat.CLower, stat.CUpper]	Konfidensintervall for stigningstallet
stat.ME	Konfidensintervallets feilmargin
stat.SESlope	Standardfeil for stigningstallet
stat.s	Standardfeil rundt linjen

Gjelder kun responstype

Utdata-variabel	Beskrivelse
[stat.CLower, stat.CUpper]	Konfidensintervall for gjennomsnittlig respons
stat.ME	Konfidensintervallets feilmargin
stat.SE	Standardfeil for gjennomsnittlig respons

Utdata-variabel	Beskrivelse
[stat.LowerPred, stat.UpperPred]	Prediksjonsintervall for én enkeltobservasjon
stat.MEPred	Prediksjonsintervallets feilmargin
stat.SEPred	Standardfeil for prediksjonen
stat.ŷ	$a + b \cdot X$ Verdi

LinRegtTest

katalog > 

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

Beregner en lineær regresjon på X - og Y -listene og en t test på verdien av stigningstallet β og korrelasjonskoeffisienten ρ for ligningen $y = \alpha + \beta x$. Den tester nullhypotesen $H_0: \beta = 0$ (tilsvarende, $\rho = 0$) mot én av tre alternative hypoteser.

Alle listene må ha samme dimensjon.

X og Y er lister av uavhengige og avhengige variabler.

$Frekv$ er en valgfri liste med frekvensverdier. Hvert element i $Frekv$ angir hvor ofte hvert korresponderende datapunkt X og Y forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

$Hypot$ er en valgfri verdi som angir en av tre alternative hypoteser som nullhypotesen ($H_0: \beta = \rho = 0$) skal testes mot.

For $H_a: \beta \neq 0$ og $\rho \neq 0$ (standard), sett $Hypot = 0$

For $H_a: \beta < 0$ og $\rho < 0$, sett $Hypot < 0$

For $H_a: \beta > 0$ og $\rho > 0$, sett $Hypot > 0$

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a + b \cdot x$
stat.t	t -observator for signifikanstest
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Frihetsgrader
stat.a, stat.b	Regresjonskoeffisienter
stat.s	Standardfeil rundt linjen
stat.SESlope	Standardfeil for stigningstallet
stat.r ²	Determinasjonskoeffisient
stat.r	Korrelasjonskoeffisient
stat.Resid	Residualene fra regresjonen

linSolve()

Katalog > 

linSolve(SystemAvLineæreLign, Var1, Var2, ...) ⇒ liste

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

linSolve(LineærLign1 and LineærLign2 and ..., Var1, Var2, ...) ⇒ liste

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

linSolve({LineærLign1, LineærLign2, ...}, Var1, Var2, ...) ⇒ liste

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

linSolve(SystemAvLineæreLign, {Var1, Var2, ...}) ⇒ liste

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

linSolve(LineærLign1 and LineærLign2 and ..., {Var1, Var2, ...}) ⇒ liste

linSolve({LineærLign1, LineærLign2, ...}, {Var1, Var2, ...}) ⇒ liste

Returnerer en liste over løsninger for variablene Var1, Var2, ...

Det første argumentet må behandles som et system av lineære ligninger eller som en lineær ligning. Ellers oppstår det en argumentfeil.

Ved for eksempel å behandle linSolve(x=1 og x=2,x) produserer et "Argumentfeil" -resultat.

Δ list() (liste)

Katalog > 

Δ List(Liste1) \Rightarrow liste

Δ List({20,30,45,70}) {10,15,25}

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive `deltaList (...)`.

Returnerer en liste som inneholder differensene mellom nabo-elementer i *Liste1*. Hvert element i *Liste1* subtraheres fra det neste elementet i *Liste1*. Resultatlisten er alltid ett element kortere enn opprinnelige *Liste1*.

list \blacktriangleright mat()

Katalog > 

list \blacktriangleright mat(Liste [, elementerRerRad]) \Rightarrow matrise

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



Returnerer en matrise fylt rad-for-rad med elementene fra *Liste*.

elementerRerRad, hvis inkludert, spesifiserer antallet elementer per rad. Grunninnstilling er antallet elementer i *Liste* (en rad).

Hvis *Liste* ikke fyller resultatmatrisen, legges det til nuller.

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive `list@>mat (...)`.

ln()

  **taster**

ln(Liste1) \Rightarrow liste

ln(2.) 0.693147

Returnerer argumentets naturlige logaritme.

For en liste, returneres elementenes naturlige logaritmer.

Hvis kompleks formatmodus er reell:

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

Hvis kompleks formatmodus er rektangulær:

ln(kvadratMatrise) ⇒ kvadratMatrise

Returnerer matrisens naturlige logaritme av *kvadratMatrise1*. Dette er ikke det samme som å beregne naturlig logaritme av hvert element. For mer informasjon om beregningsmetode, se etter under **cos()**

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I radian-vinkelmodus og rektangulært, kompleks format:

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

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

LnReg *X*, *Y*, [*Frekv*] [, **Kategori**, **Inkludert**]

Finner den logaritmiske regresjonen $y = a+b \cdot \ln(x)$ for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og *Y* er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende *X* og *Y*-dataene..

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a+b \cdot \ln(x)$

Utdata-variabel	Beskrivelse
stat.a, stat.b	Regresjonskoeffisienter
stat.r ²	Lineær determinasjonskoeffisient for transformerte data
stat.r	Korrelasjonskoeffisient for transformerte data (ln(x), y)
stat.Resid	Residualene for den logaritmiske modellen
stat.ResidTrans	Residualene for den lineære tilpasningen av de transformerte dataene
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

Local

Katalog > 

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

Deklarerer spesifiserte *vars* som lokale variabler. Disse variablene eksisterer kun mens en funksjon utføres og slettes når funksjonen er ferdig utført.

Merk: Lokale variabler sparer plass i minnet, fordi de bare eksisterer midlertidig. Dessuten forstyrrer de ingen eksisterende globale variabelverdier. Bruk lokale variabler for **For** -stigningstall og for midlertidige lagringsverdier i en flerlinjet funksjon, siden endringer på globale variabler ikke er tillatt i en funksjon.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

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 (Lås)

Katalog >

LockVar1 [, Var2] [, Var3] ...

LockVar.

Låser spesifiserte variabler eller variabelgruppe. Låste variabler kan ikke modifiseres eller slettes.

Du kan ikke låse eller låse opp systemvariabelen *Svar*, og du kan ikke låse systemvariabelgruppene *stat.* eller *tvm.*

Merk: Kommandoen **Lås (Lock)** tømmer angre/gjør om-loggen når den brukes på ulåste variabler.

Se **unlock**, side 164 og **getLockInfo()**, side 65.

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()

taster

Merk: Se også **Log-sjablon**, side 2.

Hvis *Uttr2* utelates, brukes 10 som grunntall.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

Hvis grunntall-argumentet utelates, brukes 10 som grunntall.

Hvis kompleks formatmodus er reell:

Hvis kompleks formatmodus er rektangulær:

I radian-vinkelmodus og rektangulært, kompleks format:

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

For å se hele resultatet, trykk på og bruk så og for å bevege markøren.

Logistic

katalog >

Logistic X, Y[, [Frekv] [, Kategori, Inkluder]]

Finner den logistiske regresjonen $y = c / (1 + a \cdot e^{-bx})$ for listene X og Y med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og Y er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt X og Y forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene..

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $c / (1 + a \cdot e^{-bx})$
stat.a, stat.b, stat.c	Regresjonskoeffisienter
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede X -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede Y -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

LogisticD X, Y [, [*Iterasjoner*], [*Frekv*] [, *Kategori*, *Inkluder*]]

Finner den logistiske regresjonen $y = (c / (1+a \cdot e^{-bx})+d)$ for listene X og Y med frekvensen *Frekv*, ved å bruke et angitt antall *Iterasjoner*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og Y er lister av uavhengige og avhengige variabler.

Iterasjoner er en valgfri verdi som angir maksimalt antall ganger det skal gjøres forsøk på å finne en løsning. Hvis utelatt, brukes 64. Vanligvis resulterer større verdier i bedre nøyaktighet men lengre kjøretid, og omvendt.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt X og Y forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene..

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $c/(1+a \cdot e^{-bx})+d$
stat.a, stat.b, stat.c, stat.d	Regresjonskoeffisienter
stat.Resid	Residualene fra regresjonen

Utdata-variabel	Beskrivelse
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluderte kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluderte kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

Loop (Stigningstall)

Katalog > 

Loop

Blokk

EndLoop

Utfører utsagnene gjentatte ganger i *Blokk*. Merk at stigningstallet utføres uendelig, hvis ikke en **Goto** eller **Exit** instruksjon utføres innenfor *Blokk*.

Blokk er en sekvens av utsagn som er adskilt med tegnet.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

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



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

LU *Matrise*, [*Matrise*, *uMatrise*, *pMatrise*], *Tol*]

Beregner Doolittle LU (lower-upper=nedre-øvre) dekomposisjon av en reell eller kompleks matrise. Den nedre trekantede matrisen lagres i *lMatrise*, den øvre trekantede matrisen i *uMatrise* og permutasjonsmatrisen (som beskriver radskiftene som gjøres i løpet av beregningen) i *pMatrise*.

$$lMatrise \cdot uMatrise = pMatrise \cdot matrise$$

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Tol*.

- Hvis du bruker   eller stiller modusen **Auto eller Tilnærmet** på **Tilnærmet**, utføres beregningene med flytende desimalpunktaritmetikk.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
 $5E-14 \cdot \max(\dim(Matrise)) \cdot \text{radNorm}(Matrise)$

Faktoriserende algoritme **LU** bruker delvis pivotering med radutvekslinger.

$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$	$\rightarrow m1$	$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$
LU <i>m1</i> , <i>lower</i> , <i>upper</i> , <i>perm</i>		Done
<i>lower</i>		$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
<i>upper</i>		$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
<i>perm</i>		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

M

mat>list() (matrise til liste)

mat>list(*Matrise*) \Rightarrow *liste*

Returnerer en liste fylt med elementene i *Matrise*. Elementene kopieres fra *Matrise* rad for rad.

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

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **mat@>list(...)**.

max()Katalog > **max(Liste1, Liste2) ⇒ liste**

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

max(Matrise1, Matrise2) ⇒ matrise

Returnerer maksimum (det største) av de to argumentene. Hvis argumentene er to lister eller matriser, returneres en liste eller matrise som inneholder maksimum verdi i hvert par av samsvarende elementer.

max(Liste) ⇒ Uttrykk

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

Returnerer maksimumelementet i liste.

max(Matrise1) ⇒ matrise

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

Returnerer en radvektor som inneholder maksimumselementet av hver kolonne i *Matrise1*.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 209.

Merk: Se også **min()**.**mean() (gjennomsnitt)**Katalog > **mean(Liste[, frekvListe]) ⇒ uttrykk**

$\text{mean}\{\{0.2,0,1,-0.3,0.4\}\}$	0.26
$\text{mean}\{\{1,2,3\},\{3,2,1\}\}$	$\frac{5}{3}$

Returnerer gjennomsnittet av elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

mean(Matrise1[, FrekvMatrise]) ⇒ matrise

I rektangulært vektorformat:

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

Returnerer en radvektor av gjennomsnittet for alle kolonnene i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

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

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 209.

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

median(Liste[, frekvListe]) \Rightarrow uttrykk

$\text{median}\{0.2, 0.1, -0.3, 0.4\}$ 0.2

Returnerer medianen av elementene i Liste.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i Liste.

median(MatriseI[, frekvMatrise]) \Rightarrow matrise

$\text{median}\begin{pmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{pmatrix}$ [0.4 -0.3]

Returnerer en radvektor som inneholder medianer av kolonnene i MatriseI.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i MatriseI.

Merknader:

- Alle inndata i listen eller matrisen må forenkles til tall.
- Tomme (åpne) elementer i listen eller matrisen ignoreres. For mer informasjon om tomme elementer, se side 209.

MedMed

MedMed X,Y [, Frekv] [, Kategori, Inkluder]]

Beregner median-median-linjens $(m \cdot x + b)$ for listene X og Y med frekvens Frekv. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 147.)

Alle listene må ha samme dimensjon bortsett fra Inkluder.

X og Y er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i Frekv angir hvor ofte hvert korresponderende datapunkt X og Y forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene..

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Ligning for median-median-linjen: $m \cdot x + b$
stat.m, stat.b	Modellkoeffisienter
stat.Resid	Residualene fra median-median-linjen
stat.XReg	Liste over de datapunkter i den endrede X -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede Y -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

mid() (midtstreng)

mid(kildeStreng, Start[, Antall]) ⇒ streng

Returnerer *Antall* tegn fra tegnstreng *kildeStreng*, begynnende med tegnnummer *Start*.

Hvis *Antall* utelates eller er større enn dimensjonen på *kildeStreng*, returneres alle tegnene fra *kildeStreng*, begynnende med tegnnummer *Start*.

Antall må være ≥ 0 . Hvis *Antall* = 0, returneres en tom streng.

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

mid() (midtstreng)

Katalog > 

mid(kildeListe, Start [, Antall]) ⇒ liste

Returnerer *Antall* elementer fra *kildeListe*, begynnende med elementnummer *Start*.

Hvis *Antall* utelates eller er større enn dimensjonen på *kildeListe*, returneres alle elementer fra *kildeListe*, begynnende med elementnummer *Start*.

Antall må være ≥ 0 . Hvis *Antall* = 0, returneres en tom liste.

mid(kildeStrengListe, Start[, Antall]) ⇒ liste

Returnerer *Antall* strenger fra listen over strenger *kildeStrengListe*, begynnende med elementnummer *Start*.

$\text{mid}(\{9,8,7,6\},3)$	$\{7,6\}$
$\text{mid}(\{9,8,7,6\},2,2)$	$\{8,7\}$
$\text{mid}(\{9,8,7,6\},1,2)$	$\{9,8\}$
$\text{mid}(\{9,8,7,6\},1,0)$	$\{\}$

$\text{mid}(\{"A","B","C","D"\},2,2)$	$\{"B","C"\}$
---------------------------------------	---------------

min() (minimum)

Katalog > 

min(Liste1, Liste2) ⇒ liste

min(Matrise1, Matrise2) ⇒ matrise

Returnerer minimum (det minste) av de to argumentene. Hvis argumentene er to lister eller matriser, returneres en liste eller matrise som inneholder minimumsverdien i hvert par av samsvarende elementer.

min(Liste) ⇒ Uttrykk

Returnerer minimumselementet av *Liste*.

min(Matrise!) ⇒ matrise

Returnerer en radvektor som inneholder minimumselementet av hver kolonne i *Matrise1*.

Merk: Se også **max()**.

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

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

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

mirr

(
finansRente, *reinvestRente*, *CF0*, *CFListe*
 [, *CFFrekv*])

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

Finansiell funksjon som returnerer modifisert rente av en investering.

finansRente er den renten som du betaler på kontantstrømbeløpene.

reinvestRente er den renten som kontantstrømmen reinvesteres til.

CF0 er kontantstrømmen ved start kl. 0; den må være et reelt tall.

CFListe er en liste over kontantstrømbeløpene etter den innledende kontantstrømmen *CF0*.

CFFrekv er en valgfri liste der hvert element spesifiserer frekvensen av forekomsten for et gruppert (konsekutivt) kontantstrømbeløp, som er det tilsvarende elementet til *CFListe*. Grunninnstilling er 1; hvis du legger inn verdier, må dette være positive heltall < 10.000.

Merk: Se også *irr()*, side 75.

mod()

mod(*Liste1*, *Liste2*) ⇒ *liste*

$\text{mod}(7,0)$	7
-------------------	---

mod(*Matrise1*, *Matrise2*) ⇒ *matrise*

$\text{mod}(7,3)$	1
-------------------	---

Returnerer det første argumentet modulo det andre argumentet slik som definert ved identitetene:

$\text{mod}(-7,3)$	2
--------------------	---

$\text{mod}(7,-3)$	-2
--------------------	----

$\text{mod}(-7,-3)$	-1
---------------------	----

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

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

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

Hvis det andre argumentet ikke er null, er resultatet periodisk i dette argumentet. Resultatet er enten null eller det har samme fortegn som det andre argumentet.

mod()

Katalog > 

Hvis argumentene er to lister eller matriser, returneres en liste eller matrise som inneholder modulen av hvert par av samsvarende elementer.

Merk: Se også `rest()`, side 126

mRow() (mRad)

Katalog > 

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

mRowAdd() (mRadAdd)

Katalog > 

Returnerer en kopi av *Matrise1* med hvert element i rad *Indeks2* i *Matrise1* erstattet med:

MultReg

katalog > 

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

Beregner multiple lineære regresjoner av liste *Y* for listene *X2, X2, ..., X10*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.b0, stat.b1, ...	Regresjonskoeffisienter
stat.R ²	Multipel determinasjonskoeffisient
stat.ŷList	$\hat{y}\text{List} = b_0+b_1 \cdot x_1+ \dots$
stat.Resid	Residualene fra regresjonen

MultRegIntervals $Y, X1[,X2[,X3,...$
 $[,X10]]], XVerListe[,CNivå]$

Beregner en predikert y -verdi, et prediksjonsintervall med nivå C for én enkelt observasjon, og et konfidensintervall med nivå C for den gjennomsnittlige responsen.

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat. \hat{y}	Et punkttestimat: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ for <i>XVerListe</i>
stat.dfError	Feilens frihetsgrader
stat.CLower, stat.CUpper	Konfidensintervall for gjennomsnittlig respons
stat.ME	Konfidensintervall feilmargin
stat.SE	Standardfeil for gjennomsnittlig respons
stat.LowerPred, stat.UpperPred	Prediksjonsintervall for én enkeltobservasjon
stat.MEPred	Prediksjonsintervallets feilmargin
stat.SEPred	Standardfeil for prediksjonen
stat.bList	Liste over regresjonskoeffisienter, $\{b_0, b_1, b_2, \dots\}$
stat.Resid	Residualene fra regresjonen

MultRegTests

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

Multipel lineær regresjonstest beregner en multipel lineær regresjon på de angitte dataene og beregner den globale statistiske F - og t -testobservatoren for koeffisientene.

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.F	Global statistisk F -testobservator
stat.PVal	P-verdi knyttet til global F -observator
stat.R ²	Multipel determinasjonskoeffisient
stat.AdjR ²	Justert multipel determinasjonskoeffisient
stat.s	Standardavvik for feilen
stat.DW	Durbin-Watson-observator. Brukes for å bestemme om første ordens autokorrelasjon er til stede i modellen
stat.dfReg	Regresjonens frihetsgrader
stat.SSReg	Regresjonens kvadratsum
stat.MSReg	Regresjonens kvadratgjennomsnitt
stat.dfError	Feilens frihetsgrader
stat.SKvFeil	Feilens kvadratsum
stat.MSError	Feilens kvadratgjennomsnitt (gjennomsnittlig kvadratavvik)
stat.bList	{ b_0, b_1, \dots } Liste over koeffisienter
stat.tList	Liste over statistiske t -observatorer, én for hver koeffisient i bList
stat.PList	Liste over P-verdier for hver t -observator
stat.SEList	List over standardfeil for koeffisientene i bList
stat.ŷList	\hat{y} List = $b_0+b_1 \cdot x_1+ \dots$
stat.Resid	Residualene fra regresjonen
stat.sResid	Standardiserte residualer. Beregnes ved å dividere en restverdi (residual) med dens standardavvik

Utdata-variabel	Beskrivelse
stat.CookDist	Cooks distanse. Mål for innflytelsen av en observasjon basert på residual og stigning
stat.Leverage	Mål for hvor langt verdiene for den uavhengige variabelen er fra gjennomsnittsverdiene

N

nand (ikke både...og)

ctrl **=** -taster

BoolskUttr1 **nand** *BoolskUttr2*
returnerer *Boolsk uttrykk*

$x \geq 3$ and $x \geq 4$	$x \geq 4$
---------------------------	------------

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

BoolskListe1 **nand** *BoolskListe2*
returnerer *Boolsk liste*

BoolskMatrise1 **nand** *BoolskMatrise2*
returnerer *Boolsk matrise*

Returnerer negasjon av en logisk and-handling på de to argumentene.

Returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Heltall1 **nand** *Heltall2* ⇒ *heltall*

Sammenlikner to reelle heltall bit-for-bit med en and-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 0 hvis begge bitene er 1; ellers er resultatet 1. Den returnerte verdien representerer bit-resultatene og vises i grunntallmodus.

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}
----------------------	------------

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 10).

nCr() (antKomb)Katalog > **nCr(Liste1, Liste2)**⇒liste
$$\text{nCr}(\{5,4,3\}, \{2,4,2\}) \quad \{10,1,3\}$$

Returnerer en liste over kombinasjoner basert på samsvarende elementpar i de to listene. Argumentene må ha samme listestørrelse.

nCr(Matrise1, Matrise2)⇒matrise
$$\text{nCr}\left(\begin{pmatrix} 6 & 5 \\ 4 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}\right) \quad \begin{pmatrix} 15 & 10 \\ 6 & 3 \end{pmatrix}$$

Returnerer en matrise av kombinasjoner basert på samsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.

nDerivative()Katalog > **nDerivative(Uttr1, Var=Verdi [,Orden])**⇒verdi
$$\text{nDerivative}(|x|, x=1) \quad 1$$
$$\text{nDerivative}(|x|, x)|_{x=0} \quad \text{undef}$$
nDerivative(Uttr1, Var[,Orden] | Var=Verdi)⇒verdi
$$\text{nDerivative}(\sqrt{x-1}, x)|_{x=1} \quad \text{undef}$$

Returnerer den numeriske deriverte som er beregnet ved hjelp av automatiske derivasjonsmetoder.

Hvis *verdi* er spesifisert, opphever den eventuell forhåndsstilt verdi eller aktuell “|” erstatning for variabelen.

Den deriverte må være av orden **1** eller **2**.

newList() (nyListe)Katalog > **newList(numElementer)**⇒liste
$$\text{newList}(4) \quad \{0,0,0,0\}$$

Returnerer en liste med en dimensjon lik *numElementer*. Hvert element er null.

newMat() (nyMat)Katalog > **newMat(numRader, numKolonner)**⇒matrise
$$\text{newMat}(2,3) \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Returnerer en matrise med bare nuller med dimensjonen *numRader* og *numKolonner*.

nfMax()Katalog > **nfMax**(*Uttr1*, *Var*) \Rightarrow verdi

$\text{nfMax}(x^2 - 2 \cdot x - 1, x)$	-1.
--	-----

nfMax(*Uttr1*, *Var*, *nedGrense*) \Rightarrow verdi

$\text{nfMax}(0.5 \cdot x^3 - x - 2, x, -5, 5)$	5.
---	----

nfMax(*Uttr1*, *Var*, *nedGrense*,
øvGrense) \Rightarrow verdi**nfMax**(*Uttr1*, *Var*) | *nedGrense* \leq *Var*
 \leq *øvGrense* \Rightarrow verdi

Returnerer et forslag til numerisk verdi av variabel *Var*, der lokalt maksimum av *Uttr1* forekommer.

Hvis du setter *nedGrens* og *øvGrens*, ser funksjonen i det lukkede intervallet [*nedGrens*,*øvGrens*] etter lokalt maksimum.

nfMin()Katalog > **nfMin**(*Uttr1*, *Var*) \Rightarrow verdi

$\text{nfMin}(x^2 + 2 \cdot x + 5, x)$	-1.
--	-----

nfMin(*Uttr1*, *Var*, *nedGrense*) \Rightarrow verdi

$\text{nfMin}(0.5 \cdot x^3 - x - 2, x, -5, 5)$	-5.
---	-----

nfMin(*Uttr1*, *Var*, *nedGrense*,
øvGrense) \Rightarrow verdi**nfMin**(*Uttr1*, *Var*) | *nedGrense* \leq *Var*
 \leq *øvGrense* \Rightarrow verdi

Returnerer et forslag til numerisk verdi av variabel *Var*, der lokalt minimum av *Uttr1* forekommer.

Hvis du setter *nedGrens* og *øvGrens*, ser funksjonen i det lukkede intervallet [*nedGrens*,*øvGrens*] etter lokalt minimum.

nInt()Katalog > **nInt**(*Uttr1*, *Var*, *Nedre*, *Øvre*) \Rightarrow uttrykk

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

Hvis integranden *Uttr1* ikke inneholder andre verdier enn *Var*, og hvis *Nedre* og *Øvre* er konstanter, positiv ∞ , eller negativ ∞ , så returnerer **nInt()** tilnærmet av $\int(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre})$. Denne tilnærmede er et veiet gjennomsnitt av noen utvalgsverdier av integranden i intervallen $\text{Nedre} < \text{Var} < \text{Øvre}$.

Målet er seks signifikante sifre. Den adaptive algoritmen slutter når det er sannsynlig at målet er nådd, eller når det er usannsynlig at ekstra utvalg vil gi nevneverdig forbedring.

Det kommer til syne et varsel ("Tvilsom nøyaktighet") når det ser ut til at målet ikke er nådd.

Nest **nInt()** å utføre multiplenumerisk integrasjon. Integrasjonsgrensene kan avhenge av integrasjonsvariabler utenfor dem.

$$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x\right), x, 0, 1\right) \quad 3.30423$$

nom()

nom(*effektivRente*, *CpY*) \Rightarrow verdi

$$\text{nom}(5.90398, 12) \quad 5.75$$

Finansiell funksjon som omregner den årlige effektive renten *effektivRente* til en nominell rente, gitt *CpY* som antall renteperioder per år.

effektivRente må være et reelt tall, og *CpY* må være et reelt tall > 0 .

Merk: Se også **eff()**, side 44.

nor (verken ... eller)

BoolskUttr1 **nor** *BoolskUttr2* returnerer *Boolsk uttrykk*

$$x \geq 3 \text{ or } x \geq 4 \quad x \geq 3$$

$$x \geq 3 \text{ nor } x \geq 4 \quad x < 3$$

BoolskListe1 **nor** *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrise1 **nor** *BoolskMatrise2* returnerer *Boolsk matrise*

Returnerer negasjon av en logisk or-handling på de to argumentene.
Returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Heltall1 nor *Heltall2* ⇒ *heltall*

Sammenlikner to reelle heltall bit-for-bit med en nor-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis begge bitene er 1; ellers er resultatet 0. Den returnerte verdien representerer bit-resultatene og vises i grunntallmodus.

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 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()

katalog > 

norm(*Matrise*) ⇒ *uttrykk*

norm(*Vektor*) ⇒ *uttrykk*

Returnerer Frobenius-normen.

normCdf()

Katalog > 

normCdf(*nedreGrense*, *øvreGrense*, μ [, σ]) ⇒ *tall* hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er lister

Beregner sannsynligheten i normalfordelingen mellom *nedreGrense* og *øvreGrense* for den angitte μ (standard=0) og σ (standard=1).

normPdf(*XVerdi*[$[\mu, \sigma]$]) \Rightarrow *tall* hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste

Beregner sannsynlighetstetthet for normalfordelingen ved en spesifisert *XVerdi*-verdi for spesifisert μ og σ .

not (ikke)

not *BoolskUttr1* \Rightarrow *Boolsk uttrykk*

Returnerer sann, usann eller en forenklet form av argumentet.

not *Heltall1* \Rightarrow *heltall*

Returnerer tallets komplement av et reelt heltall. Internt er *Heltall1* omregnet til et 64-biters binært tall med fortegn. Verdien av hver bit er forskjøvet (0 blir til 1 og motsatt) for tallets komplement. Resultatene vises i forhold til grunntall-modusen.

Du kan skrive inn heltallet med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten prefiks blir heltallet behandlet som et desimalt tall (grunntall 10).

Hvis du oppgir et desimalt heltall som er for stort for et 64-bit binært tall med fortegn, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se **►Base2**, side 16.

I heksades grunntall-modus:

Viktig: Null, ikke bokstaven O.

not 0h7AC36	0hFFFFFFFFFFFF853C9
-------------	---------------------

I binær grunntall-modus:

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

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

Merk: Et binært innlegg kan bestå av opptil 64 siffer (i tillegg til prefikset 0b). Et heksadesimalt innlegg kan bestå av opptil 16 siffer.

nPr() (antPerm)

nPr(*Liste1*, *Liste2*) \Rightarrow *liste*

Returnerer en liste over permutasjoner basert på samsvarende elementpar i de to listene. Argumentene må ha samme listestørrelse.

nPr({5,4,3}, {2,4,2})	{20,24,6}
-----------------------	-----------

nPr() (antPerm)Katalog > **nPr**(*Matrise1, Matrise2*) \Rightarrow *matrise*

$nPr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$
--	---

Returnerer en matrise av permutasjoner basert på tilsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.

npv()Katalog > **npv**(*Rentefot, CFO, CFListe, CFFrekv*)

$list1 := \{6000, -8000, 2000, -3000\}$	$\{6000, -8000, 2000, -3000\}$
$list2 := \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$npv(10, 5000, list1, list2)$	4769.91

Finansiell funksjon som beregner netto nåverdi; summen av nåverdier for kontantstrøm inn og ut. Et positivt resultat for npv forteller at en investering er lønnsom.

Rentefot er den renten som trekkes fra kontantstrømmene (pengekostnadene) over en periode.

CFO er kontantstrømmen ved start kl. 0; den må være et reelt tall.

CFListe er en liste med kontantstrømbeløp etter den innledende kontantstrømmen *CFO*.

CFFrekv er en liste der hvert element spesifiserer frekvensen av forekomsten for et gruppert (konsekutivt) kontantstrømbeløp, som er det tilsvarende elementet til *CFListe*. Grunninnstilling er 1; hvis du legger inn verdier, må dette være positive heltall < 10.000.

nSolve() (nLøs)Katalog > **nSolve**(*Ligning, Var[=Forslag]*) \Rightarrow *tall eller feil_streng*

$nSolve(x^2 + 5x - 25 = 9, x)$	3.84429
$nSolve(x^2 = 4, x = -1)$	-2.
$nSolve(x^2 = 4, x = 1)$	2.

nSolve(*Ligning, Var [=Forslag], nedGrense, øvGrense*) \Rightarrow *tall eller feil_streng***nSolve**(*Ligning, Var [=Forslag], nedGrense, øvGrense*) \Rightarrow *tall eller feil_streng*

Merk: Hvis det foreligger flere løsninger, kan du bruke et forslag som hjelp for å finne en spesiell løsning.

nSolve(Ligning, Var[=Forslag]) |
nedGrense ≤ Var ≤ øvGrense ⇒ tall eller feil_streng

Søker iterativt etter en tilnærmet reell numerisk løsning for *Ligning* i variabelen. Spesifiser variabelen som:

variabel

– eller –

variabel = reelt tall

For eksempel er x gyldig, og det er $x=3$ også.

nSolve() prøver å bestemme enten ett punkt der rest er null, eller to relativt nære punkter, der rest har motsatte fortegn og størrelsen på resten ikke er for stor. Hvis den ikke kan oppnå dette med et lite antall utvalgspunkter, returnerer den strengen "fant ingen løsning."

$\text{nSolve}(x^2+5\cdot x-25=9, x) x < 0$	-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 (EnVar)

OneVar [1, X], [Frekv], [Kategori, Inkludert]]

OneVar [n, X1, X2 [X3 [... [X20]]]]

Beregner en-variabel-statistikker med opptil 20 lister. En oversikt over resultatene lagres i *stat.results*-variabelen (side 147).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X-argumentene er datalister.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hver korresponderende *X*-verdi forekommer. Standardverdien er 1. Alle elementene må være heltall ≥ 0 .

Kategori er en liste med numeriske kategorikoder for de korresponderende *X*-verdiene.

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

Et tomt (åpent) element i enhver av listene *X*, *Frekv* eller *Kategori* resulterer i et åpent (tomt) element for det tilsvarende elementet til alle disse listene. Et tomt element i enhver av listene fra *X1* til *X20* resulterer i et åpent (tomt) element for det tilsvarende elementet i alle disse listene. For mer informasjon om tomme elementer, se side 209.

Utdata-variabel	Beskrivelse
stat. \bar{x}	Gjennomsnitt av x-verdier
stat. Σx	Sum av x-verdier
stat. Σx^2	Sum av x^2 -verdier
stat.sx	Utvalgets standardavvik av x
stat. x	Populasjonens standardavvik av x
stat.n	Antall datapunkter
stat.MinX	Minimum av x-verdier
stat.Q ₁ X	Første kvartil av x
stat.MedianX	Median av x
stat.Q ₃ X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdier
stat.SSX	Sum av kvadratavvik fra gjennomsnittet av x

or (eller)

BoolskUttr1 or *BoolskUttr2* returnerer
Boolsk uttrykk

BoolskListe1 or *BoolskListe2* returnerer
Boolsk liste

BoolskMatrise1 or *BoolskMatrise2*
returnerer *Boolsk matrise*

```
Define g(x)=Func
    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

Returnerer sann eller usann eller en forenklet form av opprinnelig uttrykk.

Returnerer sann hvis ett eller begge uttrykkene er sanne. Returnerer usann kun hvis begge uttrykkene behandles som usanne.

Merk: Se `xor`.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

Heltall1 or Heltall2 **heltall**

Sammenlikner to reelle heltall bit-for-bit med en or-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis minst en av bitene er 1; resultatet er 0 bare hvis begge bitene er 0. Den returnerte verdien representerer bit-resultatene og vises i grunntall-modus.

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 10).

Hvis du oppgir et desimalt heltall som er for stort for et 64-bit binært tall med fortegn, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se **Base2**, side 16.

Merk: Se `xor`.

I heksades grunntall-modus:

0h7AC36 or 0h3D5F	0h7BD7F
-------------------	---------

Viktig: Null, ikke bokstaven O.

I binær grunntall-modus:

0b100101 or 0b100	0b100101
-------------------	----------

Merk: Et binært innlegg kan bestå av opptil 64 siffer (i tillegg til prefikset 0b). Et heksadesimalt innlegg kan bestå av opptil 16 siffer.

ord() (num. tegnkode)Katalog > **ord**(*Streng*) \Rightarrow *heltall*

ord("hello") 104

ord(*Liste l*) \Rightarrow *liste*

char(104) "h"

ord(char(24)) 24

Returnerer numerisk kode for de første tegnene i tegnstreng *Streng*, eller en liste over de første tegnene i hvert listeelement.

ord({"alpha", "beta"}) {97,98}

P**P@Rx()**Katalog > **P@Rx**(*rUtr*, θ *Utr*) \Rightarrow *uttrykk*

I Radian-vinkelmodus:

P@Rx(*rListe*, θ *Liste*) \Rightarrow *liste***P@Rx**(*rMatrise*, θ *Matrise*) \Rightarrow *matrise*

Returnerer ekvivalent x-koordinat av (*r*, θ)-paret.

Merk: θ -argumentet tolkes enten som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus. Hvis argumentet er et uttrykk, kan du bruke $^\circ$, g eller r for å hoppe over vinkelmodus-innstillingen midlertidig.

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **P@>Rx** (...).

P@Ry()Katalog > **P@Ry**(*rListe*, θ *Liste*) \Rightarrow *liste*

I Radian-vinkelmodus:

P@Ry(*rMatrise*, θ *Matrise*) \Rightarrow *matrise*

Returnerer ekvivalent y-koordinat av (*r*, θ)-paret.

Merk: θ -argumentet tolkes enten som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus.

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive **P@>Ry** (...).

PassErr

For et eksempel på **PassErr**, se eksempel 2 under **Try** -kommandoen, side 158.

Sender en feil til neste nivå.

Hvis systemvariabelen *feilKode* er null, gjør ikke **PassErr** noenting.

Else -leddet i **Try...Else...EndTry**-blokken bør bruke **ClrErr** eller **PassErr**. Hvis feilen skal bearbejdes eller ignoreres, bruk **ClrErr**. Hvis det ikke er kjent hva som skal gjøres med feilen, bruk **PassErr** for å sende den til den neste feilbehandleren. Hvis det ikke er flere ventende **Try...Else...EndTry** feilbehandlere, vises feil-dialogboksen som normalt.

Merk: Se også **ClrErr (SlettFeil)**, side 23, og **Try**, side 158.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

piecewise() (stykkevis)

piecewise(*Uttr1* [, *Betingelse1* [, *Uttr2* [, *Betingelse2* [, ...]]])

Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$	Done
--	------

Returnerer definisjoner for en stykkevis definert funksjon i form av en liste. Du kan også opprette stykkevis definisjoner med en sjablon.

$p(1)$	1
$p(-1)$	undef

Merk: Se også **stykkevis-sjablon**, side 2.

poissCdf()

poissCdf(λ , *nedreGrense*, *øvreGrense*) \Rightarrow *tall*
hvis *nedreGrense* og *øvreGrense* er tall,
liste hvis *nedreGrense* og *øvreGrense* er lister

poissCdf(λ , *øvreGrense*) (for $P(0 \leq X \leq \text{øvreGrense}) \Rightarrow$ *tall* hvis *øvreGrense* er et tall, liste hvis *øvreGrense* er en liste

Beregner en kumulativ sannsynlighet for den diskrete Poisson-fordeling med spesifisert gjennomsnitt λ .

For $P(X \leq \text{øvreGrense})$, sett $\text{nedreGrense}=0$

poissPdf($\lambda, XVerd$) \Rightarrow tall hvis $XVerd$ er et tall, liste hvis $XVerd$ er en liste

Beregner en sannsynlighet for diskret Poisson-fordeling med spesifisert gjennomsnitt λ .

Vektor ► Polar

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive `@>Polar`.

Viser *vektor* i polar form $[r \angle \theta]$. Vektoren må være av dimensjon 2 og kan være en rad eller en kolonne.

Merk: ►Polar er en visningsformat-instruksjon, ikke en omregningsfunksjon. Du kan bare bruke den på slutten av en kommandolinje, og den oppdaterer ikke svar.

Merk: Se også ►Rekt, side 123.

kompleksVerdi ► Polar

Viser *kompleksVektor* i polar form.

- Grader-vinkelmodus returnerer $(r \angle \theta)$.
- Radian-vinkelmodus returnerer $re^{i\theta}$.

I Radian-vinkelmodus:

I Gradian-vinkelmodus:

$(4 \cdot i)$ ►Polar

$(4 \angle 100)$

kompleksVerdi kan ha en hvilken som helst kompleks form. Men hvis du legger inn $re^{i\theta}$, forårsaker dette feil når vinkelmodus er grader.

I Grader-vinkelmodus:

Merk: Du må bruke parentes for å legge inn polar $(r \angle \theta)$.

polyEval(*Liste1*, *Utr1*) \Rightarrow uttrykk

polyEval(*Liste1*, *Liste2*) \Rightarrow uttrykk

Tolker det første argumentet som koeffisienter for et polynom med fallende eksponenter, og returnerer en utregnet verdi av polynomet, innsatt verdien av det andre argumentet.

polyRoots(*Poly*, *Var*) \Rightarrow liste

polyRoots(*KoeffListe*) \Rightarrow liste

Den første syntaksen, **polyRoots**(*Poly*, *Var*), returnerer en liste over sanne røtter av polynom *Poly* med hensyn på variabel *Var*. Hvis det ikke eksisterer noen sanne røtter, returneres en tom liste: {}.

Den andre syntaksen, **polyRoots**(*KoeffListe*), returnerer en liste over sanne røtter for koeffisientene i *KoeffListe*.

Merk: Se også **cPolyRoots()**, side 31.

PowerReg *X*, *Y* [, *Frekv*] [, *Kategori*, *Inkluder*]]

Finner potensregresjonen $y = (a \cdot (x)^b)$ for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 147).

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og *Y* er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene.

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot (x)^b$
stat.a, stat.b	Regresjonskoeffisienter
stat.r ²	Lineær determinasjonskoeffisient for transformerte data
stat.r	Korrelasjonskoeffisient for transformerte data ($\ln(x)$, $\ln(y)$)
stat.Resid	Residualene for potensmodellen
stat.ResidTrans	Residualene for den lineære tilpasningen av de transformerte dataene
stat.XReg	Liste over de datapunkter i den endrede X -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede Y -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

Prgm

Katalog > 

Prgm
Blokk
EndPrgm

Sjablon for å opprette et egendefinert program. Må brukes med kommandoene **Define**, **Define LibPub** eller **Define LibPriv**.

Blokk kan være ett enkelt utsagn, en rekke utsagn adskilt med ":"-tegnet eller en rekke med utsagn på separate linjer.

Beregn GCD og vis mellomresultater.

```
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

PrgmKatalog > 

Blokk kan være ett enkelt utsagn, en rekke utsagn adskilt med “:”-tegnet, eller en rekke med utsagn på separate linjer.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

```
proggcd(4560,450)
```

```
450 60
60 30
30 0
GCD=30
```

```
Done
```

prodSeq()

Se Π(), side 184.

Produkt (Π)

Se Π(), side 184.

Product()Katalog > 

product(*Liste*[, *Start*[, *slutt*]]) ⇒ *uttrykk*

Returnerer produktet av elementene i *Liste*. *Start* og *Slutt* er valgfrie. De spesifiserer et elementområde.

product(*Matrise*1[, *Start*[, *slutt*]]) ⇒ *matrise*

Returnerer en radvektor som inneholder produktet av elementene i kolonnene i *Matrise*1. *Start* og *slutt* er alternativer. De spesifiserer et radområde.

Tomme (åpne) elementer ignorerer. For mer informasjon om tomme elementer, se side 209.

```
product(
  ( 1 2 3 )
  ( 4 5 6 )
  ( 7 8 9 )
) [28 80 162]
```

```
product(
  ( 1 2 3 )
  ( 4 5 6 )
  ( 7 8 9 )
) ,1,2 [4 10 18]
```

propFrac() (ekteBrøk)Katalog > 

propFrac(*rasjonal_tall*) returnerer *rasjonalt_tall* som summen av et heltall og en brøk som har samme fortegn og større nevner enn teller.

```
propFrac(4/3) 1 + 1/3
```

```
propFrac(-4/3) -1 - 1/3
```

propFrac(rasjonalt_uttrykk,Var)

returnerer summen av ekte brøk og et polynom med hensyn på *Var*. Gradene til *Var* i nevneren er større enn gradene til *Var* i telleren i hver ekte brøk. Liknende potenser av *Var* er samlet sammen. Leddene og faktorene deres er sortert med *Var* som hovedvariabel.

Hvis *Var* utelates, utvides den ekte brøken med hensyn på de fleste hovedvariablene. Koeffisientene til den polynomiske delen omgjøres så til "ekte" med hensyn på de fleste hovedvariablene og så videre.

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

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

$$\text{propFrac}(\text{Ans})$$

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



Q

QR

QR Matrise, qMatNavn, rMatNavn(, Tol)

Beregner den faste QR faktoriseringen av en reell eller en kompleks matrise. De resulterende matrisene Q og R lagres til det spesifiserte *MatNavn*. Q-matrisen er enhetlig. R-matrisen er øvre trekantet.

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignorerer *Tol*.

- Hvis du bruker   eller stiller modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flytende desimalpunktaritmetikk.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
 $5E-14 \cdot \text{maks}(\text{dim}(\text{Matrise})) \cdot \text{radNorm}(\text{Matrise})$

Det flytende desimalpunkttalet (9.) i m1 gjøres at resultatene må beregnes i flytende desimalpunkt-form.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix}$
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	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

Faktoriseringsen QR beregnes numerisk med faste transformasjoner. Den symbolske løsningen beregnes med Gram-Schmidt. Kolonnene i $qMatNavn$ er ortonormale grunnvektorer som utspenner rommet som defineres av *matrise*.

$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$	$\rightarrow mI$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
QR mI, qm, rm		Done
qm	$\begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$	
rm	$\begin{bmatrix} \sqrt{m^2+o^2} & \frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{ m \cdot p - n \cdot o }{\sqrt{m^2+o^2}} \end{bmatrix}$	

QuadReg (KvadReg)

QuadReg X, Y [, *Frekv*] [, *Kategori*, *Inkluder*]]

Finder den kvadratiske polynomiske regresjonen $y = a \cdot x^2 + b \cdot x + c$ for listene X og Y med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 147.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og Y er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt X og Y forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene.

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Regresjonskoeffisienter
stat.R ²	Determinasjonskoeffisient
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

QuartReg

Katalog > 

QuartReg *X*, *Y* [, *Frekv*] [, *Kategori*, *Inkluder*]]

Finner den fjerdegrads polynomiske regresjonen

$y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$ for listene *X* og *Y* med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 147.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og *Y* er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt *X* og *Y* forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende *X* og *Y*-dataene..

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Regresjonskoeffisienter
stat.R ²	Determinasjonskoeffisient
stat.Resid	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede <i>X-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

R

R ► Pθ()

R ► Pθ (*xList*, *yList*) ⇒ *liste*

R ► Pθ (*xMatrix*, *yMatrix*) ⇒ *matrise*

Returnerer tilsvarende θ-koordinat for (x,y) par-argumentene.

I Grader-vinkelmodus:

I Gradian-vinkelmodus:

I Radian-vinkelmodus:

Merk: Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av gjeldende vinkelmodus-innstilling.

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive inn

R@>Ptheta (...).

I Radian-vinkelmodus:

R ► Pr ($xList, yList$) ⇒ *liste*R ► Pr ($xMatrix, yMatrix$) ⇒ *matrise*

Returnerer tilsvarende r-koordinat for (x, y) parargumentene.

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive inn **R@>Pr** (...).

► Rad

Omformer argumentet til radian vinkelmåling.

I Grader-vinkelmodus:

(1.5)►Rad	(0.02618) ^r
-----------	------------------------

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive inn **R@>Ptheta** (...).

I Gradian-vinkelmodus:

(1.5)►Rad	(0.023562) ^r
-----------	-------------------------

tilf()

rand() ⇒ *uttrykk*rand(#Trials) ⇒ *liste*

Angir det tilfeldige tallet for start.

rand() returnerer en tilfeldig verdi mellom 0 og 1.

RandSeed 1147	Done
rand(2)	{0.158206, 0.717917}

rand(#Trials) returnerer en liste som inneholder #Trials tilfeldige verdier mellom 0 og 1.

tilfBin()

randBin(n, p) ⇒ *uttrykk*randBin($n, p, \#Trials$) ⇒ *liste*

randBin(n, p) returnerer et tilfeldig reelt tall fra en spesifisert binomisk fordeling.

randBin($n, p, \#Trials$) returnerer en liste som inneholder #Trials tilfeldige reelle tall fra en spesifisert binomisk fordeling.

randInt

(
lowBound,upBound)
 ⇒ uttrykk

randInt

(*lowBound,upBound*
 ,*#Trials*) ⇒ liste

randInt

(
lowBound,upBound)
 returnerer et tilfeldig
 heltall innen
 området som er
 spesifisert av
 heltallsgrensene
lowBound og
upBound.

randInt

(*lowBound,upBound*
 ,*#Trials*) returnerer
 en liste som
 inneholder *#Trials*
 tilfeldige heltall
 innen det spesifiserte
 området.

randMat()

randMat(*numRows, numColumns*) ⇒
 matrise

Returnerer en matrise av heltall mellom -
 9 og 9 av spesifisert dimensjon.

Begge argumentene må forenkles til
 heltall.

RandSeed 1147	Done									
randMat(3,3)	<table border="1"> <tr><td>8</td><td>-3</td><td>6</td></tr> <tr><td>-2</td><td>3</td><td>-6</td></tr> <tr><td>0</td><td>4</td><td>-6</td></tr> </table>	8	-3	6	-2	3	-6	0	4	-6
8	-3	6								
-2	3	-6								
0	4	-6								

Merk: Verdiene i denne matrisen endres hver
 gang du trykker på .

randNorm()

randNorm(μ, σ) ⇒ uttrykk
randNorm($\mu, \sigma, \#Trials$) ⇒ liste

RandSeed 1147	Done
randNorm(0,1)	0.492541
randNorm(3,4.5)	-3.54356

randNorm()

Katalog > 

randNorm(μ , σ) returnerer et desimaltall fra den spesifiserte normalfordelingen. Det kan være et hvilket som helst tall, men vil være sterkt konsentrert i intervallet $[\mu-3\cdot\sigma, \mu+3\cdot\sigma]$.

randNorm(μ , σ , #Trials) returnerer en liste som inneholder #Trials desimaltall fra den spesifiserte normalfordelingen.

randPoly()

Katalog > 

randPoly(Var, Order) \Rightarrow uttrykk

Returnerer et polynom i Var av den spesifiserte Order. Koeffisientene er tilfeldige heltall i området -9 til 9. Den ledende koeffisienten blir ikke null.

Order må være 0-99.

RandSeed 1147	Done
randPoly(x,5)	$-2\cdot x^5+3\cdot x^4-6\cdot x^3+4\cdot x-6$

tilfUtv()

Katalog > 

randSamp(List,#Trials[,noRepl]) \Rightarrow liste

Returnerer en liste som inneholder et tilfeldig utvalg av #Trials forsøk fra List med mulighet for bytte av utvalg (noRepl=0), eller ingen bytte av utvalg (noRepl=1). Grunninnstillingen er med bytte av utvalg.

RandSeed

Katalog > 

RandSeed Tall

Hvis Tall = 0, settes startverdien for tilfeldig tall-generatoren til fabrikkinnstilling. Hvis Tall \neq 0, brukes det for å opprette to startverdier, som lagres i systemvariablene seed1 og seed2.

RandSeed 1147	Done
rand()	0.158206

real()

Katalog > 

Returnerer den reelle delen av argumentet.

real(List1) \Rightarrow liste

real()

Katalog > 

Returnerer den reelle delen av alle elementer.

real(Matrix1) ⇒ *matrise*

Returnerer den reelle delen av alle elementer.

► Rect

Katalog > 

Vector ► **Rect**

Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive inn @>**Rect**.

Viser *Vector* i rektangulær form $[x, y, z]$. Vektoren må være av dimensjon 2 eller 3 og kan være en rad eller en kolonne.

Merk: ► **Rect** er en visningsformat-instruksjon, ikke en omregningsfunksjon. Du kan bare bruke den på slutten av en kommandolinje, og den oppdaterer ikke *ans*.

Merk: Se også ► **Polar**, side 112.

complexValue ► **Rect**

Viser *complexValue* i rektangulær form $a+bi$. *complexValue* kan ha hvilken som helst kompleks form. Men hvis du legger inn $re^{i\theta}$, forårsaker dette feil når vinkelmodus er grader.

Merk: Du må bruke parentes for å legge inn polar ($r\angle \theta$).

I Radian-vinkelmodus:

I Gradian-vinkelmodus:

$((1 \angle 100)) \blacktriangleright \text{Rect}$ *i*

I Grader-vinkelmodus:

Merk: For å skrive \angle , velg den fra symbollisten i Katalogen.

ref()

Katalog > 

ref(Matrix1, Tol) ⇒ *matrise*

Returnerer eliminasjonsformen av *Matrix1*.

$$\text{ref} \left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix} \right) = \begin{bmatrix} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn Tol . Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres Tol .

- Hvis du bruker `ctrl` `enter` eller stiller modusen **Auto eller Tilnærmet** til Tilnærmet, utføres beregningene med flyttallsregning.
- Hvis Tol utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
 $5E-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

Unngå udefinerte elementer i Matrix1 . De kan føre til uventede resultater.

Hvis for eksempel a er udefinert i følgende uttrykk, vises en varselmelding, og resultatet vises som:

$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Varslet vises fordi det generaliserte elementet $1/a$ ikke ville være gyldig for $a=0$.

Dette kan du unngå ved å lagre en verdi til a på forhånd eller ved å bruke begrensningen (" $|$ "), som vist i følgende eksempel.


$$\text{ref}\left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) | a=0 \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Merk: Se også `rref()`, page 133.

RefreshProbeVars

Gir deg tilgang til sensordata fra alle tilkoblede sensorer gjennom TI-Basic-programmet.

StatusVar-verdi**Status**

<i>statusVar</i> =0	Normal (fortsett med programmet) Vernier DataQuest™-applikasjonen er i datainnsamlingsmodus.
<i>statusVar</i> =1	Merk: Vernier DataQuest™-applikasjonen må være i målermodus for at denne kommandoen skal fungere. 
<i>statusVar</i> =2	Vernier DataQuest™-applikasjonen er ikke startet.
<i>statusVar</i> =3	Vernier DataQuest™-applikasjonen er startet, men du har ikke koblet til noen sensorer.

Eksempel

```
Definer temp()=
Prgm
© Sjekk om systemet er klart
RefreshProbeVars-status
Hvis status=0, så
Disp «klar»
For n,1,50
RefreshProbeVars-status
temperatur:=meter.temperatur
Disp «temperatur:
»,temperature
Hvis temperatur>30 så
Disp «for varm»
EndIf
© Vent i 1 sekund mellom
utvalgene
Vent 1
EndFor
Else
Disp «ikke klar. Prøv igjen senere»
EndIf
EndPrgm
```

Merk: Dette kan også brukes med TI-Innovator™-senter.

remain(List1, List2) ⇒ *liste*
remain(Matrix1, Matrix2) ⇒ *matrise*

Returnerer resten av det første argumentet med hensyn på det andre argumentet som definert av identitetene:

$\text{remain}(x,0)$ x
 $\text{remain}(x,y)$ $x - y \cdot \text{iPart}(x/y)$

Som en konsekvens, merk at **remain** $(-x,y) = \text{remain}(x,y)$. Resultatet er enten null eller det har samme fortegn som det første argumentet.

Merk: Se også **mod()**, side 96.

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

$\text{remain}\left(\begin{bmatrix} 9 & -7 \\ 6 & 4 \end{bmatrix}, \begin{bmatrix} 4 & 3 \\ 4 & -3 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$
--	---

Forespør

Forespør *promptString*, *var*[, *DispFlag* [, *statusVar*]]

Forespør *promptString*, *func*(*arg1*, ...*argn*) [, *DispFlag* [, *statusVar*]]

Programmeringskommando: Stopper programmet og viser en dialogboks med meldingen *promptString* og en inndata-boks for brukerens svar.

Når brukeren skriver inn et svar og klikker på **OK**, blir innholdet i inndata-boksen tildelt til variabel *var*.

Hvis brukeren klikker på **Avbryt**, forsetter programmet uten å godta inndata. Programmet bruker forrige verdi for *var* hvis *var* allerede var definert.

Det valgfrie argumentet *DispFlag* kan være et hvilket som helst uttrykk.

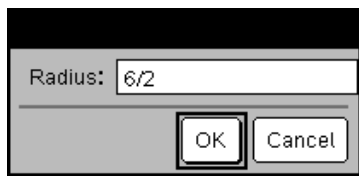
- Hvis *DispFlag* utelates eller behandles til **1**, blir prompt-meldingen og brukerens svar vist i Kalkulator-loggen.
- Hvis *DispFlag* behandles til **0**, vises ikke prompt-meldingen eller svaret i loggen.

Definere et program:

```
Definer forespør_demo()=Prgm
  Forespør "Radius: ",r
  Vis "Area = ",pi*r^2
EndPrgm
```

Kjør programmet og skriv inn et svar:

forespør_demo()



Resultat etter trykk på **OK**:

Stråle: 6/2
 Area = 28,2743

Det valgfrie *statusVar*-argumentet gir programmet en måte å bestemme hvordan brukeren avviste dialogboksen. Merk at *statusVar* krever *DispFlag*-argumentet.

- Hvis brukeren klikket på **OK** eller trykket **Enter** eller **Ctrl+Enter**, settes variabelen *statusVar* til en verdi på **1**.
- Ellers innstilles variabelen *statusVar* til en verdi på **0**.



Argumentet *func()* lar et program lagre brukerens svar som en funksjonsdefinisjon. Denne syntaksen arbeider som om brukeren utførte kommandoen:

Definer *func(arg1, ...argn)* =
brukers svar

Programmet kan så bruke den definerte funksjonen *funk()*. *promptString* skal veilede brukeren i å legge inn et passende *bruker-svar* som fullfører funksjonsdefinisjonen.

Merk: Kalkulatoren *Forespør* -kommandoen inne i et brukerdefinert program, men ikke inne i en funksjon.

Slik stopper du et program som inneholder en **Request**-kommando inne i en infinitt løkke:

- **Grafregner:** Hold nede tasten , og trykk på  flere ganger.
- **Windows®:** Hold nede tasten **F12**, og trykk på **Enter** flere ganger.
- **Macintosh®:** Hold nede tasten **F5**, og trykk på **Enter** flere ganger.
- **iPad®:** Applikasjonen viser en ledetekst. Du kan fotsette å vente, eller avbryte.

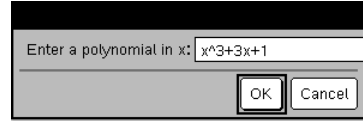
Merk: Se også **ForespørStr**, page 128.

Definere et program:

```
Definer polynom()=Prgm
  Forespør "Legg inn et polynom i
  x:",p(x)
  Vis "Reelle røtter er:",polyRøtter
  (p(x),x)
EndPrgm
```

Kjør programmet og skriv inn et svar:

polynom()



Resultat etter å ha skrevet inn x^3+3x+1 og valgt **OK**:



Reelle røtter er: $\{-0,322185\}$

ForespørStr *promptString*, var[, *DispFlag*]

Programmeringskommando: Arbeider identisk med den første syntaksen i **Request**-kommandoen, bortsett fra at brukerens svar alltid tolkes som en streng. Som kontrast tolker **Request**-kommandoen svaret som et uttrykk hvis ikke brukeren setter det i anførselstegn ("").

Merk: Du kan bruke **RequestStr** -kommandoen inne i et brukerdefinert program, men ikke inne i en funksjon.

Slik stopper du et program som inneholder en **ForespørStr**-kommando i en infinitt løkke:

- **Grafregner:** Hold nede tasten , og trykk på  flere ganger.
- **Windows®:** Hold nede tasten **F12**, og trykk på **Enter** flere ganger.
- **Macintosh®:** Hold nede tasten **F5**, og trykk på **Enter** flere ganger.
- **iPad®:** Applikasjonen viser en ledetekst. Du kan fotsette å vente, eller avbryte.

Merk: Se også **Forespør**, page 126.

Definere et program:

```
Definer forespørStr_demo()=Prgm
  ForespørStr "Navnet ditt:",navn,0
  Vis "Forespør har ",dim(navn),"
  tegn."
EndPrgm
```

Kjør programmet og skriv inn et svar:

forespørStr_demo()



Resultat etter klikk på **OK** (Merk at argumentet *DispFlag* på **0** utelater prompten og svaret fra loggen):

forespørStr_demo()

Svaret har 5 tegn.

Return

Return [*Expr*]

Returnerer *Expr* som resultatet av funksjonen. Bruk i en **Func...EndFunc**-blokk.

Merk: Bruk **Return** uten et argument i en **Prgm...EndPrgm**-blokk for å avslutte et program.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

```
Define factorial (nn)=
Func
Local answer,counter
1 → answer
For counter,1,nn
answer = counter → answer
EndFor
Return answer}
EndFunc

factorial (3) 6
```

right()**right(List1[, Num])** ⇒ *liste* $\text{right}(\{1,3,-2,4\},3)$ $\{3,-2,4\}$ Returnerer *Num*-elementene som ligger lengst til høyre i *List1*.Hvis du utelater *Num*, returneres alle elementer i *List1*.**right(sourceString[, Num])** ⇒ *streng* $\text{right}(\text{"Hello"},2)$ "lo"Returnerer *Num*-tegnene som ligger lengst til høyre i tegnstreng *sourceString*.Hvis du utelater *Num*, returneres alle elementer i *sourceString*.**right(Comparison)** ⇒ *uttrykk* $\text{right}(x<3)$ 3

Returnerer høyre side av en ligning eller ulikhet.

rk23 ()**rk23(Expr, Var, depVar, {Var0, VarMax}, depVar0, VarStep [, diftol])** ⇒ *matrise*

Differensialligning:

 $y' = 0,001 \cdot y \cdot (100 - y)$ og $y(0) = 10$ $\text{rk23}(0,001 \cdot y \cdot (100 - y), t, y, \{0,100\}, 10, 1)$

0.	1.	2.	3.	4.
10.	10.9367	11.9493	13.042	14.2

rk23(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol]) ⇒ *matrise*

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

rk23(ListOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol]) ⇒ *matrise*Samme ligning med *diftol* satt til $1.E-6$ $\text{rk23}(0,001 \cdot y \cdot (100 - y), t, y, \{0,100\}, 10, 1, 1.E-6)$

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

Bruker Runge-Kutta-metoden for å løse systemet

 $\frac{d \text{ depVar}}{d \text{ Var}} = \text{Expr}(\text{Var}, \text{depVar})$ med $\text{depVar}(\text{Var}0) = \text{depVar}0$ i intervallet $[\text{Var}0, \text{VarMax}]$. Returnerer en matrise, hvis første rad definerer resultatverdiene av *Var* som definert av *VarStep*. Den andre raden definerer verdien av den første løsningskomponenten for de tilsvarende *Var*-verdiene, og så videre.

System av ligninger:

$$\begin{cases} y1' = -y1 + 0.1 \cdot y1 \cdot y2 \\ y2' = 3 \cdot y2 - y1 \cdot y2 \end{cases}$$
med $y1(0) = 2$ og $y2(0) = 5$

Expr er høyre side, som definerer den ordinære differensialligningen (ODE).

SystemOfExpr er et system på høyre side som definerer systemet av ODE-er (tilsvare rekkefølgen av avhengige variabler i *ListOfDepVars*).

ListOfExpr er en liste over høyresider som definerer systemet av ODE-er (tilsvare rekkefølgen av avhengige variabler i *ListOfDepVars*).

Var er den uavhengige variabelen.

ListOfDepVars er en liste over avhengige variabler.

$\{Var0, VarMax\}$ er en liste med to elementer som forteller funksjonen at den skal integrere fra *Var0* til *VarMax*.

ListOfDepVars0 er en liste over opprinnelige verdier for avhengige variabler.

Hvis *VarStep* beregnes til et tall som ikke er null: $tegn(VarStep) = tegn(VarMax - Var0)$ og løsninger returneres ved $Var0 + i * VarStep$ for alle $i=0,1,2,\dots$ slik at $Var0 + i * VarStep$ er i $[var0, VarMax]$ (får kanskje ikke en løsningsverdi ved *VarMax*).

Hvis *VarStep* beregnes til null, returneres løsningene ved "Runge-Kutta" *Var*-verdiene.

diftol er feiltoleransen (grunnverdi på 0,001).

$$rk23 \left(\begin{array}{l} -y1+0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{array}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1 \right)$$

0.	1.	2.	3.	4.
2.	1.94103	4.78694	3.25253	1.82848
5.	16.8311	12.3133	3.51112	6.27245

root()

Merk: Se også **mal for N.-rot**, side 1.

rotate()

rotate(*Integer* *I*, *#ofRotations*) \Rightarrow *heltall* | binær grunntall-modus:

rotate()

Katalog > 

Hvis $\#ofRotations$ er positiv, skjer rotasjonen til venstre. Hvis $\#ofRotations$ er negativ, skjer rotasjonen til høyre. Grunninnstilling er -1 (roterer ett tegn mot høyre).

round()

Katalog > 

Returnerer argumentet avrundet til spesifisert antall sifre etter desimalpunktet.

$\text{round}(1.234567, 3)$ 1.235

$digits$ må være et heltall i området $1 - 12$. Hvis $digits$ ikke er inkludert, returneres argumentet avrundet til 12 signifikante sifre.

Merk: Visning av siffermodus kan ha innvirkning på hvordan dette vises.

$\text{round}(List1[, digits]) \Rightarrow$ liste

$\text{round}(\{\pi, \sqrt{2}, \ln(2)\}, 4)$
 $\{3.1416, 1.4142, 0.6931\}$

Returnerer en liste over elementer avrundet til spesifisert antall sifre.

$\text{round}(Matrix1[, digits]) \Rightarrow$ matrise

$\text{round}\left(\begin{bmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{bmatrix}, 1\right)$ $\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$

Returnerer en matrise over elementer avrundet til spesifisert antall sifre.

rowAdd()

Katalog > 

$\text{rowAdd}(Matrix1, rIndex1, rIndex2) \Rightarrow$ matrise

Returnerer en kopi av $Matrix1$ med rad $rIndex2$ erstattet med summen av rader $rIndex1$ og $rIndex2$.

rowDim()

Katalog > 

$\text{rowDim}(Matrix) \Rightarrow$ uttrykk

Returnerer antallet rader i $Matrix$.

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

Merk: Se også $\text{colDim}()$, side 24.

rowNorm()

Katalog > 

rowNorm(*Matrix*) ⇒ *uttrykk*

Returnerer den største summen av absoluttverdiene for elementene i radene i *Matrix*.

Merk: Alle matriseelementene må forenkles til tall. Se også **colNorm()**, side 24.

$$\text{rowNorm}\left(\begin{pmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{pmatrix}\right) = 25$$

rowSwap()

Katalog > 

rowSwap(*Matrix1*, *rIndex1*, *rIndex2*)
⇒ *matrise*

Returnerer *Matrix1* med rader *rIndex1* og *rIndex2* ombyttet.

$$\begin{array}{l} \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \rightarrow \text{mat} \\ \text{rowSwap}(\text{mat}, 1, 3) \end{array} \quad \begin{array}{l} \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \\ \begin{pmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{pmatrix} \end{array}$$

rref()

Katalog > 



rref(*Matrix1*[, *Tol*]) ⇒ *matrise*

Returnerer den reduserte eliminasjonsformen av *Matrix1*.

$$\text{rref}\left(\begin{pmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{pmatrix}\right) = \begin{pmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{pmatrix}$$

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Tol*.

$$\text{rref}\left(\begin{pmatrix} a & b \\ c & d \end{pmatrix}\right) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- Hvis du bruker   eller stiller modusen **Auto eller Tilnærmet** til Tilnærmet, utføres beregningene med flyttallsregning.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
 $5E-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

Merk: Se også `ref()`, page 123.

S

sec()

 tast

`sec(Liste1)` ⇒ *liste*

I Grader-vinkelmodus:

Merk: Argumentet tolkes som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling. Du kan bruke $^{\circ}$, $^{\text{G}}$ eller $^{\text{r}}$ for å hoppe over vinkelmodusen midlertidig.

 $\sec^{-1}()$  tast

`sec-1(Liste1)` ⇒ *liste*

I Grader-vinkelmodus:

Merk: Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling.

`sec-1(1)` 0

I Gradian-vinkelmodus:

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive `arcsec (...)`.

`sec-1(√2)` 50

I Radian-vinkelmodus:

sech()

Katalog > 

`sech(Liste1)` ⇒ *liste*

 $\text{sech}^{-1}()$ Katalog > 

`sech-1(Liste1)` ⇒ *liste*

I Radian-vinkelmodus og Rectangular-kompleksmodus:

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive `arcsech (...)`.

Send

Hub-meny

`Send uttrEllerStreng1[, uttrEllerStreng2] ...`

Eksempel: Slå på det blå elementet i den innebygde RGB LED-skjermen i 0,5 sekunder.

Programmeringskommando: Sender én eller flere TI-Innovator™ Hub kommandoer til en tilkoblet hub.

uttrEllerStreng må være en gyldig TI-Innovator™ Hub kommando. Vanligvis inneholder *uttrEllerStreng* en "INNSTALL ..." -kommando for å kontrollere en enhet eller en «LES ...»-kommando for å etterspørre data.

Argumentene sendes suksessivt til hubben.

Merk: Du kan bruke **Send**-kommandoen i et brukerdefinert program, men ikke i en funksjon.

Merk: Se også **Get** (side 60), **GetStr** (side 67), og **eval()** (side 48).

```
Send "SET COLOR.BLUE ON TIME .5"
```

Done

Eksempel: Etterspør nåværende verdi fra hubbens innebygde lysnivåsensor. En **Get**-kommando henter verdien og tildeler den til variabelen *lysver*.

```
Send "READ BRIGHTNESS"
```

Done

```
Get lightval
```

Done

```
lightval
```

0.347922

Eksempel: Send en kalkulert frekvens til hubbens innebygde høyttaler. Bruk den spesielle variabelen *iostr.SendAns* for å vise hubkommandoen med uttrykket som er evaluert.

```
n:=50
```

50

```
m:=4
```

4

```
Send "SET SOUND eval(m·n)"
```

Done

```
iostr.SendAns
```

"SET SOUND 200"

seq() (sekv)

Katalog > 

$\text{seq}(\text{Uttr}, \text{Var}, \text{Lav}, \text{Høy}, \text{Trinn}) \Rightarrow \text{liste}$

Øker *Var* fra *Lav* til *Høy* med trinn på *Intervall*, behandler *Uttr* og returnerer resultatene som en liste. Det opprinnelige innholdet i *Var* er fremdeles der etter at **seq()** er fullført.

Grunnverdien for *Intervall* = 1.

$$\text{seq}(n^2, n, 1, 6) \quad \{1, 4, 9, 16, 25, 36\}$$

$$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right) \quad \left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$$


$$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right) \quad \frac{1968329}{1270080}$$

Merk: For å tvinge frem et tilnærmet desimalresultat,

Grafregner: Trykk på  .

Windows®: Trykk på **Ctrl+Enter**.

Macintosh®: Trykk på **⌘+Enter**.

iPad®: Hold på **enter**, og velg .

$$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right) \quad 1.54977$$

seqGen(*Uttr*, *Var*, *avhVar*, {*Var0*, *VarMaks*}, *ListeAvInnlLedd* [, *VarIntervall* [, *Lofverdi*]]) ⇒ *liste*

Genererer en liste over ledd for tallfølge *avhVar*(*Var*)=*Uttr* som følger: Øker uavhengig variabel *Var* fra *Var0* til *VarMaks* med *VarIntervaller*, beregner *avhVar*(*Var*) for tilsvarende verdier av *Var* ved hjelp av *Uttr*-formel og *ListeAvInnlLedd*, og returnerer resultatene som en liste.

seqGen(*ListeEllerSystemAvUttr*, *Var*, *ListeMedAvhVarer*, {*Var0*, *VarMaks*} [, *MatriseAvInnlLedd* [, *VarIntervall* [, *Lofverdi*]]) ⇒ *matrise*

Genererer en matrise av ledd for et system (eller en liste) av tallfølger *ListeMedAvhVarer* (*Var*)=*ListeEllerSystemAvUttr* som følger: Øker uavhengig variabel *Var* fra *Var0* til *VarMaks* med *VarIntervall*, behandler *ListeMedAvhVarer*(*Var*) for tilsvarende verdier av *Var* ved hjelp av *ListeEllerSystemAvUttr*-formel og *MatriseAvInnlLedd*, og returnerer resultatene som en matrise.

Opprinnelig innhold i *Var* er uendret etter at **seqGen()** er fullført.

Grunnverdien for *VarIntervall* = 1.

Genererer de første 5 leddene i tallfølgen $u(n) = u(n-1)^2/2$, med $u(1)=2$ og *VarIntervall*=1.

$$\text{seqGen}\left(\frac{u(n-1)^2}{n}, n, u, \{1,5\}, \{2\}\right)$$

$$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$$

Eksempel der *Var0*=2:

$$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2,5\}, \{3\}\right)$$

$$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$$

System av to tallfølger:

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u^2(n-1)}{2} + u(n-1)\right\}, n, \{u1, u2\}, \{1,5\}, \left[\begin{array}{c} 1 \\ 2 \end{array}\right]\right)$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 3 & 4 & 5 \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix}$$

Merk: Tomrommet () i matrisen med innledende ledd over brukes for å angi at det innledende leddet for $u1(n)$ er beregnet ved hjelp av den eksplisitte tallfølgeformelen $u1(n)=1/n$.

seqn(*Uttr*{*u*, *n* [, *ListeMedInnlLedd* [, *nMaks* [, *Lofverdi*]]]) ⇒ *liste*

Genererer en liste over ledd for tallfølge $u(n)=\text{Uttr}(u, n)$ som følger: Øker *n* fra 1 til *nMaks* med 1, beregner $u(n)$ for tilsvarende verdier av *n* ved hjelp av formel *Uttr*(*u*, *n*) og *ListeMedInnlLedd*, og returnerer resultatene som en liste.

seqn(*uttr*{*n* [, *nMaks* [, *Lofverdi*]]}) ⇒ *liste*

Genererer de første 6 leddene i tallfølgen $u(n) = u(n-1)/2$, med $u(1)=2$.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

Genererer en liste over ledd for en ikke-rekursiv tallfølge $u(n) = Uttr(n)$ som følger: Øker n fra 1 til $nMaks$ med 1, beregner $u(n)$ for tilsvarende verdier av n ved hjelp av formelen $Uttr(n)$, og returnerer resultatene som en liste.

Hvis $nMaks$ mangler, innstilles $nMaks$ på 2500

Hvis $nMax=0$, innstilles $nMaks$ på 2500

Merk: `seqn()` kaller `seqGen()` med $n0=1$ og $nintervall=1$

SetMode() (lesModus)

`SetMode(modusNavnHeltall, innstillingHeltall) ⇒ heltall`

`SetMode(liste) ⇒ heltallsliste`

Kun gyldig innenfor en funksjon eller et program.

`SetMode(modusNavnHeltall, innstillingHeltall)` setter foreløpig modus `modusNavnHeltall` til den nye innstillingen `innstillingHeltall`, og returnerer et heltall som samsvarer med den opprinnelige innstillingen av den modusen. Endringen er begrenset til hvor lenge det varer å utføre programmet/funksjonen.

`modusNavnHeltall` spesifiserer hvilken modus du vil stille inn. Det må være en av modus-heltallene fra tabellen nedenfor.

`innstilleHeltall` spesifiserer den nye innstillingen for modusen. Det må være en av innstillingsheltallene fra listen nedenfor for den spesifikke modusen som du stiller inn.

Vis tilnærmet verdi av π ved hjelp av grunninnstillingen for Vis sifre, og vis så π med en innstilling på Fast2. Kontroller for å se at grunninnstillingen gjenoprettes etter at programmet utføres.

SetMode(*liste*) lar deg endre flere innstillinger. *liste* inneholder tallpar med modusheltall og innstillingeheltall.

SetMode(*liste*) returnerer en liknende liste med heltallpar som representerer de opprinnelige modusene og innstillingene.

Hvis du har lagret alle modusinnstillinger med **SetMode(0)** → *var*, kan du bruke **SetMode(*var*)** for å gjenopprette disse innstillingene til funksjonen eller programmet lukkes. Se **SetMode()**, side 65.

Merk: De aktuelle modusinnstillingene sendes til påkalte under rutiner. Hvis en under rutine endrer en modusinnstilling, går modusinnstillingen tapt når kontrollen går tilbake til påkallingsrutinen.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

Modus Navn	Modus Heltall	Innstillte heltall
Vis sifre	1	1=Flytende, 2=Flytende1, 3=Flytende2, 4=Flytende3, 5=Flytende4, 6=Flytende5, 7=Flytende6, 8=Flytende7, 9=Flytende8, 10=Flytende9, 11=Flytende10, 12=Flytende11, 13=Flytende12, 14=Fast0, 15=Fast1, 16=Fast2, 17=Fast3, 18=Fast4, 19=Fast5, 20=Fast6, 21=Fast7, 22=Fast8, 23=Fast9, 24=Fast10, 25=Fast11, 26=Fast12
Vinkel	2	1=Radian, 2=Grader, 3=Gradian
Eksponensielt format	3	1=Normal, 2=Vitenskapelig, 3=Teknisk
Reell eller kompleks	4	1=Reell, 2=Rektangulær, 3=Polar
Auto eller tilnærm.	5	1=Auto, 2=Tilnærmet
Vektorformat	6	1=Rektangulær, 2=Sylindrisk, 3=Sfærisk
Grunntall	7	1=Desimal, 2=Heks, 3=Binær

shift(*Heltall1* [, #*avSkift*]) ⇒ *heltall*

Forskyver (skifter) bitene i et binært heltall. Du kan legge inn *Heltall1* med hvilket som helst grunntall. Det konverteres automatisk til 64-bit binær form med fortegn. Hvis *Heltall1* er for stort for denne formen, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se **Base2**, side 16.

Hvis #*avSkift* er positiv, er skift til venstre. Hvis #*avSkift* er negativ, er skift til høyre. Grunninnstilling er -1 (skiftes én bit mot høyre).

I et høyre-skift er biten helt til høyre droppet og 0 eller 1 lagt inn for å stemme overens med den venstre biten. I et venstre-skift er biten helt til venstre droppet og 0 er lagt inn som høyre-bit.

For eksempel i et høyre-skift:

Hver bit skifter mot høyre.

0b0000000000000111101011000011010

Setter inn 0 hvis biten helt til venstre er 0, eller 1 hvis biten helt til venstre er 1.

produserer:

0b0000000000000111101011000011010

Resultatene vises i forhold til grunntall-modusen. Ledende nuller vises ikke.

shift(*Liste1* [, #*avSkift*]) ⇒ *liste*

Returnerer en kopi av *Liste1* skiftet til høyre eller til venstre av #*avSkift*-elementer. Endrer ikke *Liste1*.

Hvis #*avSkift* er positiv, er skift til venstre. Hvis #*avSkift* er negativ, er skift til høyre. Grunninnstillingen er -1 (skiftes et element til høyre).

I binær grunntall-modus:

shift(0b1111010110000110101)	0b111101011000011010
shift(256,1)	0b1000000000

I heksades grunntall-modus:

shift(0h78E)	0h3C7
shift(0h78E,-2)	0h1E3
shift(0h78E,2)	0h1E38

Viktig: Hvis du vil skrive inn et binært eller heksadesimalt tall, må du alltid bruke prefikset 0b eller 0h (null, ikke bokstaven O).

I desimalt grunntall-modus:

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() (skift)

Katalog > 

Elementer som introduseres ved begynnelsen eller slutten av *liste* ved skiftet er satt til symbolet "ude".

shift(*Streng* *l*, #*avSkift*)(\Rightarrow *streng*)

Returnerer en kopi av *Streng* *l* skiftet mot høyre eller mot venstre av #*ofShifts*-tegn. Endrer ikke *Streng* *l*.

Hvis #*avSkift* er positiv, er skift til venstre. Hvis #*avSkift* er negativ, er skift til høyre. Grunninnstillingen er -1 (skiftes et tegn mot høyre).

Tegn som introduseres ved begynnelsen eller slutten av *streng* ved skiftet er satt til et mellomrom.

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

sign() (fortegn)

Katalog > 

sign(*Liste* *l*) \Rightarrow *list e*

sign(*Matrise* *l*) \Rightarrow *matrise*

Hvis kompleks formatmodus er reell:

sign(0) returnerer ± 1 hvis kompleks formatmodus er Reell; ellers returnerer den seg selv.

sign(0) representerer enhets sirkelen i den komplekse grunnmengden.

For en liste eller matrise returneres fortegnene for alle elementene.

simult()

Katalog > 

simult(*koeffMatrise*, *konstVektor*{, *Tol*}) \Rightarrow *matrise*

Løs mhp. x og y :

$$x + 2y = 1$$

$$3x + 4y = -1$$

Returnerer en kolonnevektor som inneholder løsningene til et system av lineære ligninger.

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

Merk: Se også **linSolve()**, side 83.

koeffMatrise må være en kvadratmatrise som inneholder ligningskoeffisientene.

Løsningen er $x = -3$ og $y = 2$.

Løs:

konstVektor må ha samme antall rader (samme dimension) som *koeffMatrise* og inneholde konstantene.

$$ax + by = 1$$

$$cx + dy = 2$$

Alternativt kan ethvert matriseelement behandles som null hvis absoluttverdien er mindre enn *Tol*. Denne toleransen brukes bare hvis matrisen er lagt inn med flytende desimalpunkt og ikke inneholder noen symbolske variabler som ikke er tildelt noen verdi. Ellers ignoreres *Tol*.

- Hvis du bruker modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flytende desimalpunktaritmetikk.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
 $5E-14 \cdot \max(\dim(\text{koeffMatrise})) \cdot \text{radNavn}(\text{koeffMatrise})$

simult(koeffMatrise, konstMatrise, Tol) ⇒ *matrise*

Løser multiple systemer av lineære ligninger, hvor hvert system har samme ligningskoeffisienter men forskjellige konstanter.

Hver kolonne i *konstMatrise* må inneholde konstantene for et ligningssystem. Hver kolonne i resultatmatrisen inneholder løsningen for det tilsvarende systemet.

Løs:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left(\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ -1 & -3 \end{pmatrix}\right) \quad \begin{bmatrix} -3 & -7 \\ 2 & \frac{9}{2} \end{bmatrix}$$

For det første systemet er $x=-3$ og $y=2$. For det andre systemet er $x=-7$ og $y=9/2$.

sin(Liste1) ⇒ *liste*

I Grader-vinkelmodus:

sin(Liste1) returnerer en liste over sinus til alle elementer i *Liste1*.

I Gradian-vinkelmodus:

sin()

 **tast**

Merk: Argumentet tolkes som grader, gradianer eller som radianer, avhengig av aktuell vinkelmodus. Du kan bruke $^{\circ}$, G , eller r for å hoppe over vinkelmodusen midlertidig.

$\sin(\text{kvadratMatrise1}) \Rightarrow \text{kvadratMatrise}$

Returnerer matrisens sinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne sinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

I Radian-vinkelmodus:

$$\sin \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

$\sin^{-1}()$

 **tast**

$\sin^{-1}(\text{Liste1}) \Rightarrow \text{liste}$

$\sin^{-1}(\text{Liste1})$ returnerer en liste over invers sinus til hvert element i *Liste1*.

Merk: Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling.

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arcsin(...)**.

$\sin^{-1}(\text{kvadratMatrise1}) \Rightarrow \text{kvadratMatrise}$

Returnerer matrisens inverse sinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers sinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Grader-vinkelmodus:

I Gradian-vinkelmodus:

I Radian-vinkelmodus:

I radian-vinkelmodus og rektangulær, kompleks modus:

$$\sin^{-1} \begin{pmatrix} 1 & 5 \\ 4 & 2 \end{pmatrix} = \begin{bmatrix} -0.174533 - 0.12198 \cdot i & 1.74533 - 2.35591 \cdot i \\ 1.39626 - 1.88473 \cdot i & 0.174533 - 0.593162 \cdot i \end{bmatrix}$$

sinh()

Katalog > 

$\sinh(\text{Liste1}) \Rightarrow \text{liste}$

$$\begin{array}{ll} \sinh(1.2) & 1.50946 \\ \sinh(\{0,1,2,3\}) & \{0,1.50946,10.0179\} \end{array}$$

sinh (*Liste1*) returnerer en liste over hyperbolsk sinus til hvert element i *Liste1*.

sinh

(*kvadratMatrise1*) \Rightarrow *kvadratMatrise*

Returnerer matrisens hyperbolske sinus for *kvadratMatrise1*. Dette er ikke det samme som å beregne hyperbolsk sinus for hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

$$\sinh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$

sinh⁻¹()

sinh⁻¹(*Liste1*) \Rightarrow *liste*

sinh⁻¹(*Liste1*) returnerer en liste over invers hyperbolsk sinus til hvert element i *Liste1*.

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arcsinh** (...).

sinh⁻¹

(*kvadratMatrise1*) \Rightarrow *kvadratMatrise*

Returnerer matrisens inverse hyperbolske sinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers hyperbolsk sinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

SinReg

SinReg *X*, *Y* [, [*Iterasjoner*],[*Periode*] [, [*Kategori*,*Inkluder*]

Finner sinusregresjonen for listene X og Y .
En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 147.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og Y er lister av uavhengige og avhengige variabler.

Iterasjoner er en verdi som angir maksimalt antall ganger (1 til 16) det skal gjøres forsøk på å finne en løsning. Hvis utelatt, brukes 8. Vanligvis resulterer større verdier i bedre nøyaktighet men lengre kjøretid, og omvendt.

Periode spesifiserer en estimert periode. Hvis utelatt, bør forskjellen mellom verdiene i X være like og i sekvensiell rekkefølge. Hvis du spesifiserer *Periode*, kan forskjellene mellom x -verdiene være ulike.

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene..

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

Resultatet av **SinReg** er alltid i radianer, uavhengig av innstilling for vinkelmodus.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Regresjonskoeffisienter
stat.Rest	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede X -listen som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>

Utdata-variabel	Beskrivelse
stat.YReg	Liste over de datapunkter i den endrede <i>Y-listen</i> som faktisk brukes i regresjonen basert på begrensninger i <i>Frekv</i> , <i>Kategoriliste</i> og <i>Inkluder kategorier</i>
stat.FreqReg	Liste over frekvenser som samsvarer med <i>stat.XReg</i> og <i>stat.YReg</i>

SortA (SorterSt)

Katalog > 

SortA *Liste1* [, *Liste2*] [, *Liste3*]...

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
---------------------------------	---------------

SortA *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

SortA <i>list1</i>	Done
--------------------	------

Sorterer elementene i det første argument i stigende rekkefølge.

<i>list1</i>	$\{1,2,3,4\}$
--------------	---------------

Hvis du inkluderer andre argumenter, sorteres elementene av hvert slik at den nye posisjonen deres stemmer overens med den nye posisjonen til elementene i det første argumentet.

$\{4,3,2,1\} \rightarrow list2$	$\{4,3,2,1\}$
---------------------------------	---------------

Alle argumentene må være navn på lister eller vektorer. Alle argumentene må ha like dimensjoner.

SortA <i>list2,list1</i>	Done
--------------------------	------

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 209.

<i>list2</i>	$\{1,2,3,4\}$
--------------	---------------

<i>list1</i>	$\{4,3,2,1\}$
--------------	---------------

SortD (SorterSy)

Katalog > 

SortD *Liste1* [, *Liste2*] [, *Liste3*] ...

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
---------------------------------	---------------

SortD *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
---------------------------------	---------------

Identisk med **SortA**, bortsett fra at **SortD** sorterer elementene i fallende rekkefølge.

SortD <i>list1,list2</i>	Done
--------------------------	------

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 209.

<i>list1</i>	$\{4,3,2,1\}$
--------------	---------------

<i>list2</i>	$\{3,4,1,2\}$
--------------	---------------

►Sphere (sfærisk)

Katalog > 

Vektor ►Sphere

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @>Sphere.

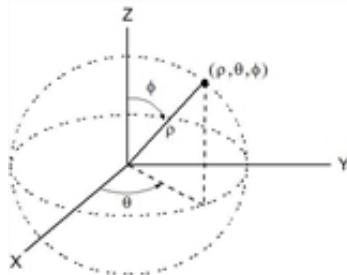
Viser rad- eller kolonnevektor i sfærisk form [ρ \angle θ \angle ϕ].

Vektor må være av dimensjon 3 og kan enten være en rad- eller en kolonnevektor.

Merk: ►Sphere er en visningsformat-instruksjon, ikke en omregningsfunksjon. Du kan bare bruke den på slutten av en kommandolinje.

```
[1 2 3]►Sphere  
[3.74166 1.10715 0.640522]
```

```
([2 1.5708 3])►Sphere  
[3.60555 0.785398 0.588003]
```



sqrt() (kvdrt)

Katalog > 

sqrt(Liste1)⇒liste

Returnerer kvadratroten til argumentet.

For en liste, returneres kvadratroten til alle elementene i Liste1.

Merk: Se også **Kvadratrot-sjablon**, side 1.

stat.results

Viser resultater fra en statistisk beregning.

Resultatene vises som et sett av navn-verdi-par. De spesifikke navnene som vises er avhengige av den aller siste brukte statistikkfunksjonen eller kommandoen.

Du kan kopiere et navn eller en verdi og lime den inn i andre posisjoner.

Merk: Unngå å definere variabler som bruker de samme navnene som de som brukes for statistisk analyse. I noen tilfeller kan det oppstå feilbetingelse. I tabellen nedenfor finner du en liste over variabelnavn som brukes for statistisk analyse.

 $xlist := \{1,2,3,4,5\}$ $\{1,2,3,4,5\}$
 $ylist := \{4,8,11,14,17\}$ $\{4,8,11,14,17\}$
LinRegMx $xlist, ylist, 1$: stat.results

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r ² "	0.996109
"r"	0.998053
"Resid"	"{...}"

stat.values	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	"{-0.4,0.4,0.2,0,-0.2}"

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBLOCK
stat.AdjR ²	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r ²	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.ox	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.oy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.ox1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.ox2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.x̄
stat.b9	stat.FBlock	stat.ŷ	stat.Σx ²	stat.x̄1
stat.b10	stat.Fcol	stat.ŷ1	stat.Σxy	stat.x̄2
stat.bList	stat.FInteract	stat.ŷ2	stat.Σy	stat.x̄Diff
stat.χ ²	stat.FreqReg	stat.ŷDiff	stat.Σy ²	stat.x̄List

stat.c	stat.Frow	stat.PList	stat.s	stat.XReg
stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat.y
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat.z
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat.zList
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

Merk: Hver gang applikasjonen Lister og regneark beregner statistiske resultater, kopierer den "stat."-gruppevariablene til en "stat#."-gruppe, der # er et tall som økes automatisk. På den måten kan du bevare tidligere resultater mens du utfører flere beregninger.

stat.values (stat.verdier)

Katalog > 

stat.values

Se `stat.results` -eksemplet.

Viser en matrise av verdiene som er beregnet for siste behandlede statistikkfunksjon eller kommando.

I motsetning til `stat.results` utelater `stat.values` navnene som assosieres med verdiene.

Du kan kopiere en verdi og lime dette inn i andre posisjoner.

stDevPop() (stAvvPop)

Katalog > 

`stDevPop(Liste[, frekvListe])` ⇒ uttrykk

I radian-vinkelmodus og automatisk modus:

Returnerer populasjonens standardavvik for elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

Merk: *Liste* må ha minst to elementer. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 209

stDevPop(*Matrise1*[,
FrekvMatrise]) \Rightarrow *matrise*

Returnerer en radvektor av populasjonens standardavvik i kolonnene i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

Merk: *Matrise1* må ha minst to rader. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 209.

stDevSamp(*Liste*[,*frekvListe*]) \Rightarrow *uttrykk*

Returnerer utvalgets standardavvik av elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

Merk: *Liste* må ha minst to elementer. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 209

stDevSamp(*Matrise1*[,
frekvMatrise]) \Rightarrow *matrise*

Returnerer en radvektor av utvalgets standardavvik av kolonnene i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

Merk: *Matrise1* må ha minst to rader. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 209.

Stop (Stopp)

Katalog > 

Stop

Programmeringskommando: Avslutter programmet.

Stop er ikke tillatt i funksjoner.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

$i:=0$	0
Define $prog1()$ =Prgm	Done
For $i,1,10,1$	
If $i=5$	
Stop	
EndFor	
EndPrgm	
$prog1()$	Done
i	5

Lagre

Se → (lagre), side 192.

String() (Streng)

Katalog > 

$string(Uttr) \Rightarrow string$

Forenkler $Uttr$ og returnerer resultatet som en tegnstreng.

subMat() (undermatrise)

Katalog > 

$subMat(MatriseI[, startRad] [, startKol] [, endRad] [, endKol]) \Rightarrow matrise$

Returnerer den spesifiserte undermatrisen til $MatriseI$.

Grunninnstillinger: $startRad=1$, $startKol=1$, $endRad=siste\ rad$, $endKol=siste\ kolonne$.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
$subMat(m1,2,1,3,2)$	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
$subMat(m1,2,2)$	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)

Se G(), side 185.

sum()

Katalog > 

$sum(Liste[, Start[, Slutt]]) \Rightarrow uttrykk$

Returnerer summen av elementene i *Liste*.

Start og *Slutt* er alternativer. De spesifiserer et elementområde.

Et tomt utsagn produserer et tomt resultat. Tomme (åpne) elementer i *Liste* ignoreres. For mer informasjon om tomme elementer, se side 209.

sum(MatriseI[, Start[, Slutt]]) ⇒ *matrise*

Returnerer en radvektor som inneholder summene av elementene i kolonnene i *MatriseI*.

Start og *Slutt* er alternativer. De spesifiserer et radområde.

Et tomt utsagn produserer et tomt resultat. Tomme (åpne) elementer i *MatriseI* ignoreres. For mer informasjon om tomme elementer, se side 209.

sum	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$	[5 7 9]
sum	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	[12 15 18]
sum	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, 2, 3$	[11 13 15]

sumlf(Liste, Kriterium[, SumListe]) ⇒ *verdi*

Returnerer samlet sum av alle elementene i *Liste* som møter de spesifiserte *Kriterier*. Eventuelt kan du spesifisere en endringsliste, *sumListe*, for å hente de elementene som skal samles (akkumuleres).

Liste kan være et uttrykk, en liste eller en matrise. *SumListe*, hvis spesifisert, må ha samme dimensjon(er) som *Liste*.

Kriterium kan være:

- En verdi, et uttrykk eller en streng. For eksempel, **34** samler kun de elementene i *Liste* som forenkles til verdien 34.
- Et boolsk uttrykk som inneholder symbolet **?** som plassholder for hvert element. For eksempel, **? < 10** samler kun de elementene i *Liste* som er mindre enn 10.

sumIf()

Katalog > 

Hvis et *Liste*-element møter *Kriteriene*, legges dette elementet til den samlede summen. Hvis du inkluderer *sumListe*, legges tilsvarende element fra *sumListe* til summen istedenfor.

I applikasjonen *llster* og regneark kan du bruke et celleområde istedenfor *Liste* og *sumListe*.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 209.

Merk: Se også `countIf()`, side 30.

sumSeq()

Se $\Sigma()$, side 185.

system()

Katalog > 

Returnerer et ligningssystem, formateret som en liste. Du kan også opprette et system med en sjablon.

$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right) \quad x=4 \text{ and } y=-4$$

Keep CAS image?

T

T(transponert)

katalog > 

Matrise $I^T \Rightarrow$ *matrise*

Returnerer den komplekse konjugerte transponerte av *Matrise*1.

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive @t.

tan()

 **tast**

`tan(Liste1)` \Rightarrow *liste*

I Grader-vinkelmodus:

`tan(Liste1)` returnerer en liste over tangensene til alle elementene i *Liste1*.

I Gradian-vinkelmodus:

tan()



Merk: Argumentet tolkes som grader, gradianer eller som radianer, avhengig av aktuell vinkelmodus. Du kan bruke $^{\circ}$, G , eller r for å hoppe over vinkelmodusen midlertidig.

tan(kvadratMatrise1)⇒kvadratMatrise

Returnerer matrisetangensen av kvadratMatrise1. Dette er ikke det samme som å beregne tangens for hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

I Radian-vinkelmodus:

$$\tan \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

tan⁻¹()



tan⁻¹(Liste1)⇒liste

tan⁻¹(Liste1) returnerer en liste over de inverse tangenser til hvert element i Liste1.

Merk: Resultatet returneres som en vinkel i enten grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling.

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arctan (...)**.

tan⁻¹

(kvadratMatrise1)⇒kvadratMatrise

Returnerer matrisens inverse tangens til kvadratMatrise1. Dette er ikke det samme som å beregne invers tangens til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Grader-vinkelmodus:

$$\tan^{-1}(1) = 45$$

I Gradian-vinkelmodus:

$$\tan^{-1}(1) = 50$$

I Radian-vinkelmodus:

$$\tan^{-1}(\{0,0.2,0.5\}) = \{0,0.197396,0.463648\}$$

I Radian-vinkelmodus:

$$\tan^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

tanh(Liste1) ⇒ liste

tanh(Liste1) returnerer en liste av hyperbolske tangenser til hvert element i *Liste1*.

tanh
(*kvadratMatrise1*) ⇒ *kvadratMatrise*

Returnerer matrisens hyperbolske tangens til *kvadratMatrise1*. Dette er ikke det samme som å beregne hyperbolsk tangens til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

$$\tanh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

tanh⁻¹(Liste1) ⇒ liste

tanh⁻¹(Liste1) returnerer en liste over invers hyperbolsk tangens til hvert element i *Liste1*.

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **arctanh** (...).

tanh⁻¹
(*kvadratMatrise1*) ⇒ *kvadratMatrise*

Returnerer matrisens inverse hyperbolske tangens til *kvadratMatrise1*. Dette er ikke det samme som å beregne invers hyperbolsk tangens til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I rektangulært, kompleks format:

I radian-vinkelmodus og rektangulært, kompleks format:

$$\tanh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.099353+0.164058 \cdot i & 0.267834-1.4908 \\ -0.087596-0.725533 \cdot i & 0.479679-0.94730 \\ 0.511463-2.08316 \cdot i & -0.878563+1.7901 \end{bmatrix}$$

For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

tCdf(nedGrense,øvGrense,df) ⇒ tall hvis

nedGrens og *øvGrens* er tall, *liste* hvis *nedGrens* og *øvGrens* er lister

Beregner student-*t*-fordelingens sannsynlighet mellom *nedGrens* og *øvGrens* for spesifisert grader av frihet *df*.

Text

Text*promptStreng*[, *VisFlagg*]

Programmeringskommando: Stopper programmet og viser tegnstrengen *promptStreng* i en dialogboks.

Når brukeren klikker på **OK**, fortsetter programmet å utføre.


Det valgfrie *flagg* -argumentet kan være et hvilket som helst uttrykk.

- Hvis *VisFlagg* utelates eller behandles til **1**, blir tekstmeldingen lagt til i Kalkulator-loggen.
- Hvis *VisFlagg* behandles til **0**, blir tekstmeldingen ikke lagt til i loggen.

Hvis programmet trenger et skrevet svar fra brukeren, kan du se etter på **Request**, side 126 eller **RequestStr**, side 128.

Merk: Du kan bruke denne kommandoen inne i et brukerdefinert program, men ikke inne i en funksjon.

Definer et program som stopper for å vise hver av de fem tilfældige tallene i en dialogboks.

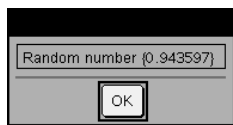
Innenfor malen `Prgm...EndPrgm` fullfører du hver linje ved å trykke på  istedenfor på `enter`. På tastaturet på datamaskinen, hold nede **Alt** og trykk på **Enter**.

```
Define tekst_demo()=Prgm
  For i,1,5
    stringo:="Random number " &
    string(rand(i))
    Text stringo
  EndFor
EndPrgm
```

Kjør program:

tekst_demo()

Eksempel på dialogboks:



Then (Så)

tInterval *Liste* [, *Frekv* [, *CNivå*]]

(Dataliste-inndata)

tInterval \bar{x} , *sx*, *n* [, *CNivå*]

(Oppsummerende statistikk-inndata)

Beregner et *t*-konfidensintervall. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 147.)

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall for et ukjent populasjonsgjennomsnitt
stat. \bar{x}	Utvalgets gjennomsnitt av datasekvensen fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.df	Grader-av-frihet
stat. σ_x	Utvalgets standardavvik
stat.n	Lengde av datasekvensen med utvalgsgjennomsnitt

tInterval_2Samp

tInterval_2Samp *Liste1*, *Liste2* [, *Frekv1* [, *Frekv2* [, *CNivå* [, *Felles*]]]]

(Dataliste inndata)

tInterval_2Samp $\bar{x}1$, *sx1*, *n1*, $\bar{x}2$, *sx2*, *n2* [, *CNivå*, *Felles*]

(Summering statistikk inndata)

Beregner et to-utvalgs *t* konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147.)

Felles=1 fellesvarianser; *Felles*=0 gjør ikke fellesvarianser.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling
stat. $\bar{x}1$ - $\bar{x}2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.df	Grader-av-frihet
stat. $\bar{x}1$, stat. $\bar{x}2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat. $\sigma x1$, stat. $\sigma x2$	Utvalgets standardavvik for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antall utvalg i datasekvenser
stat.sp	Det felles standardavviket. Beregnet når <i>Felles</i> = JA.

tPdf()

Katalog > 

tPdf(*XVerd*,*df*) \Rightarrow *tall* hvis *XVerd* er et tall,
liste hvis *XVerd* er en liste

Beregner sannsynlighetstetthetsfunksjonen (pdf) for Student-*t*-fordelingen ved en spesifisert *x*-verdi med spesifiserte grader av frihet *df*.

trace()

katalog > 

Returnerer diagonalsummen (summen av alle elementene på hoveddiagonalen) til *kvadratMatrise*.

Try*blokk1***Else***blokk2***EndTry**

Utfører *blokk1* med mindre det oppstår en feil. Programmet overfører til *blokk2* hvis en feil oppstår i *blokk1*. Systemvariabelen *feilKode* inneholder feilkoden, dermed kan programmet utføre retting av feil. For en liste over feilkoder, se "Feilkoder og feilmeldinger," side 219.

blokk1 og *blokk2* kan enten være et enkelt utsagn eller en sekvens av utsagn som er adskilt med tegnet "·".

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjete program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

Eksempel 2

For å se kommandoene **Try**, **ClrErr** og **PassErr** i drift, legg inn `egenverdier()` - programmet som vist til høyre. Kjør programmet ved å utføre hver av følgende uttrykk.

$$\text{egenvals} \left(\begin{pmatrix} -3 \\ -41 \\ 5 \end{pmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix} \right)$$

Merk: Se også **ClrErr**, side 23, og **PassErr**, side 111.

```
Define progI()=Prgm
  Try
    z:=z+1
    Disp "z incremented."
  Else
    Disp "Sorry, z undefined."
  EndTry
EndPrgm
```

Done

```
z:=1:progI()
.....
z incremented.
```

Done

```
DelVar z:progI()
.....
Sorry, z undefined.
```

Done

```
Define egenverdier(a,b)=Prgm
```

© Programmet `egenverdier(A,B)` viser egenverdier av A·B

Try

Disp "A=" ,a

Disp "B=" ,b

Disp " "

Disp "Egenverdier av A·B er:",egVd(a*b)

Else

If feilKode=230 Then

Disp "Feil: Produkt av A·B må være en kvadratmatrise"

ClrErr

Else

PassErr

Endlf

EndTry

EndPrgm

tTest**tTest** $\mu_0, \text{Liste}, [\text{Frekv}], [\text{Hypot}]$

(Dataliste inndata)

tTest $\mu_0, \bar{x}, sx, n, [\text{Hypot}]$

(Summering statistikk inndata)

Utfører en hypotesetest for ett enkelt ukjent populasjonsgjennomsnitt μ når populasjonens standardavvik σ er ukjent. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147).

Test $H_0: \mu = \mu_0$, mot ett av følgende:For $H_a: \mu < \mu_0$, sett *Hypot*<0For $H_a: \mu \neq \mu_0$ (standard), sett *Hypot*=0For $H_a: \mu > \mu_0$, set *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.t	$(\bar{x} - \mu_0) / (\text{stdev} / \text{sqrt}(n))$
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader-av-frihet
stat. \bar{x}	Utvalgsgjennomsnitt av datasekvensen i <i>Liste</i>
stat.sx	Utvalgets standardavvik av datasekvensen
stat.n	Utvalgenes størrelse

tTest_2Samp *Liste1, Liste2[, Frekv1[, Frekv2
[, Hypot[, Felles]]]]*

(Dataliste inndata)

tTest_2Samp $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2$ [*Hypot
[, Felles]*]

(Summering statistikk inndata)

Beregner en to-utvalgs t -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147).

Test $H_0: \mu_1 = \mu_2$, mot ett av følgende:

For $H_a: \mu_1 < \mu_2$, sett *Hypot*<0

For $H_a: \mu_1 \neq \mu_2$ (standard), sett *Hypot*=0

For $H_a: \mu_1 > \mu_2$, sett *Hypot*>0

Felles=1 fellesvarianser

Felles =0 gir ikke fellesvarianser

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.t	Standard normalverdi beregnet for forskjellen i gjennomsnitt
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.df	Grader av frihet for t-statistikken
stat. $\bar{x}1$, stat. $\bar{x}2$	Utvalgets gjennomsnitt av datasekvenser i <i>Liste1</i> og <i>Liste2</i>
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i <i>Liste 1</i> og <i>Liste2</i>
stat.n1, stat.n2	Utvalgenes størrelse
stat.sp	Det felles standardavviket. Beregnet når <i>Felles</i> =1.

tvmFV()

tvmFV $(N, I, PV, Pmt, [PpY], [CpY], [PmtAt]) \Rightarrow verdi$

tvmFV(120,5,0,-500,12,12)

77641.1

tvmFV()

Katalog > 

Finansiell funksjon som beregner fremtidig verdi for penger.

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 162. Se også **amortTbl()**, side 7.

tvmI()

Katalog > 

tvmI($N, PV, Pmt, FV, [PpY], [CpY], [PmtAt]$) \Rightarrow verdi

tvmI(240,100000,-1000,0,12,12) 10.5241

Finansiell funksjon som beregner rentefoten per år.

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 162. Se også **amortTbl()**, side 7.

tvmN()

Katalog > 

tvmN($I, PV, Pmt, FV, [PpY], [CpY], [PmtAt]$) \Rightarrow verdi

tvmN(5,0,-500,77641,12,12) 120.

Finansiell funksjon som beregner antallet betalingsperioder.

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 162. Se også **amortTbl()**, side 7.

tvmPmt()

Katalog > 

tvmPmt($N, I, PV, FV, [PpY], [CpY], [PmtAt]$) \Rightarrow verdi

tvmPmt(60,4,30000,0,12,12) -552.496

Finansiell funksjon som beregner beløpet for hver betaling.

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 162. Se også **amortTbl()**, side 7.

tvmPV(*N,I,Pmt,FV,[PpY],[CpY],[PmtAt]*)⇒verdi

tvmPV(48,4,-500,30000,12,12) -3426.7

Finansiell funksjon som beregner nåverdien.

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 162. Se også **amortTbl()**, side 7.

TVM-argument*	Beskrivelse	Datatype
N	Antall betalingsperioder	reelt tall
I	Årlig rente (rentefot)	reelt tall
PV	Nåverdi	reelt tall
Pmt	Betalingsbeløp	reelt tall
FV	Fremtidig verdi	reelt tall
PpY	Antall betalinger pr. år, grunninnstilling=1	heltall > 0
CpY	Antall renteperioder pr. år, grunninnstilling=1	heltall > 0
PmtAt	Betaling som forfaller ved slutten eller begynnelsen av hver periode, grunninnstilling=avslutt	heltall (0=avslutte, 1=begynne)

* Disse tidsverdi-for-penger-argumentnavnene likner TVM-variabelnavnene som f.eks. **tvm.pv** og **tvm.pmt** som brukes av *Calculator* applikasjonens finansløser. Men finansielle funksjoner lagrer ikke argumentverdiene eller resultatene til TVM-variablene.

TwoVar *X, Y*, [*Frekv*] [, *Kategori*, *Inkluder*]]

Beregner 2-variabls statistiske observatorer. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 147.)

Alle listene må ha samme dimensjon bortsett fra *Inkluder*.

X og *Y* er lister av uavhengige og avhengige variabler.

Frekv er en valgfri liste med frekvensverdier. Hvert element i *Frekv* angir hvor ofte hvert korresponderende datapunkt X og Y forekommer. Standardverdien er 1. Alle elementene må være heltall 0.

Kategori er en liste over kategorikoder for de tilsvarende X og Y -dataene..

Inkluder er en liste med én eller flere av kategorikodene. Bare dataelementene med kategorikode som er i listen blir inkludert i beregningen.

Et tomt (åpent) element i enhver av listene X , *Frekv* eller *Kategori* resulterer i et tomt (åpent) element for det tilsvarende elementet til alle disse listene. Et tomt element i enhver av listene fra $X1$ til $X20$ resulterer i et tomt (åpent) element for det tilsvarende elementet i alle disse listene. For mer informasjon om tomme elementer, se side 209.

Utdata-variabel	Beskrivelse
stat. \bar{x}	Gjennomsnitt av x-verdier
stat. x	Sum av x-verdier
stat. x2	Sum av x2 verdier
stat.sx	Utvalgets standardavvik til x
stat. x	Populasjonens standardavvik til x
stat.n	Antall datapunkter
stat. \bar{y}	Gjennomsnitt av y-verdier
stat. y	Sum av y-verdier
stat. y ²	Sum av y2-verdier
stat.sy	Utvalgets standardavvik til y
stat. y	Populasjonens standardavvik til y
stat. xy	Sum av x · y -verdier
stat.r	Korrelasjonskoeffisient
stat.MinX	Minimum av x-verdier

Utdata-variabel	Beskrivelse
stat.Q ₁ X	Første kvartil av x
stat.MedianX	Median av x
stat.Q ₃ X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdier
stat.MinY	Minimum av y-verdier
stat.Q ₁ Y	Første kvartil av y
stat.MedY	Median av y
stat.Q ₃ Y	Tredje kvartil av y
stat.MaxY	Maksimum av y-verdier
stat. (x-) ²	Sum av kvadratavvik fra gjennomsnittet av x
stat. (y-) ²	Sum av kvadrat for avvik fra gjennomsnittet av y

U

unitV() (enhetsV)

Katalog > 

unitV(*Vektor1*) ⇒ vektor

Returnerer enten en rad- eller kolonne-
enhetsvektor, avhengig av formen på
Vektor1.

Vektor1 må være enten en enkel-rad-
matrise eller en enkel-kolonne-matrise.

unlock

Katalog > 

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

unlock*Var*.

Låser opp spesifisert variabel eller
variabelgruppe. Låste variabler kan ikke
modifiseres eller slettes.

Se **lock**, side 87, og **getLockInfo()**, side 65.

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

varPop()Katalog > **varPop**(*Liste*[,*frekvListe*]) \Rightarrow uttrykkReturnerer populasjonsvariansen for *Liste*.Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.**Merk:** *Liste* må inneholde minst to elementer.

Hvis et element i en av listene er tomt (åpent), ignoreres dette elementet, og det tilsvarende elementet i den andre listen ignoreres også. For mer informasjon om tomme elementer, se side 209.

varSamp() (utvalgets varians)Katalog > **varSamp**(*Liste*[,*frekvListe*]) \Rightarrow uttrykkReturnerer utvalgets varians for *Liste*.Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.**Merk:** *Liste* må inneholde minst to elementer.

Hvis et element i en av listene er tomt (åpent), ignoreres dette elementet, og det tilsvarende elementet i den andre listen ignoreres også. For mer informasjon om tomme elementer, se side 209.

varSamp(*Matrise1*[,
frekvMatrise]) \Rightarrow matriseReturnerer en radvektor som inneholder utvalgets varians for hver kolonne i *Matrise1*.Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

$$\text{varSamp} \left(\begin{array}{ccc|cc} 1 & 2 & 5 & 4.75 & 1.03 & 4 \\ -3 & 0 & 1 & & & \\ .5 & .7 & 3 & & & \end{array} \right)$$

$$\text{varSamp} \left(\begin{array}{cc|cc} -1.1 & 2.2 & 6 & 3 \\ 3.4 & 5.1 & 2 & 4 \\ -2.3 & 4.3 & 5 & 1 \end{array} \right) \left[3.91731 \quad 2.08411 \right]$$

Hvis et element i en av matrisene er tomt (åpent), ignoreres dette elementet, og det tilsvarende elementet i den andre matrisen ignoreres også. For mer informasjon om tomme elementer, se side 209.

Merk: *Matrise1* må inneholde minst to rader.

W

Wait



Wait tidenISekunder

Utsetter utførelsen i en periode på *tidenISekunder* sekunder.

Wait er spesielt nyttig i et program som krever en kort forsinkelse for at anmodede data skal bli tilgjengelige.

Argumentet *tidenISekunder* må være et uttrykk som forenkler til en desimalverdi i området 0 til og med 100. Kommandoen avrunder denne verdien opp til nærmeste 0,1 sekunder.

For å avbryte en **Wait** som pågår,

- **Grafregner:** Hold nede tasten , og trykk på  flere ganger.
- **Windows®:** Hold nede tasten **F12**, og trykk på **Enter** flere ganger.
- **Macintosh®:** Hold nede tasten **F5**, og trykk på **Enter** flere ganger.
- **iPad®:** Applikasjonen viser en ledetekst. Du kan fortsette å vente, eller avbryte.

Merk: Du kan bruke kommandoen **Wait** innenfor et brukerdefinert program, men ikke innenfor en funksjon.

For å vente 4 sekunder:

Wait 4

For å vente 1/2 sekund:

Wait 0.5

For å vente 1,3 sekunder med bruk av variabelen *seccount*:

seccount:=1.3
Wait seccount

Dette slår på en grønn indikatorlampe i 0,5 sekunder og slår den deretter av.

Send "SET GREEN 1 ON"

Wait 0.5

Send "SET GREEN 1 OFF"

warnCodes ()

warnCodes(*Uttr1, StatusVar*)⇒*uttrykk*

Behandler uttrykk *Uttr1*, returnerer resultatet, og lagrer kodene for alle genererte varsler i listevariabelen *StatusVar*. Hvis ingen varsler er generert, tildeler denne funksjonen *StatusVar* en tom liste.

Uttr1 kan være et hvilket som helst gyldig matematisk uttrykk i TI-Nspire™ eller TI-Nspire™ CAS. Du kan ikke bruke en kommando eller tildeling som *Uttr1*.

StatusVar må være et gyldig variabelnavn.

Se side 227 for en liste over varselkoder og assosierte meldinger.

when() (når)

when(*Betingelse*, *santResultat* [, *usantResultat*][, *ukjentResultat*])
⇒ uttrykk

Returnerer *santResultat*, *usantResultat*, eller *ukjentResultat*, avhengig av om *Betingelse* er sann, usann eller ukjent. Returnerer inndata hvis det er for få argumenter til å spesifisere korrekt resultat.

Utelat både *usantResultat* og *ukjentResultat* for å definere et uttrykk bare i det området der *Betingelse* er sann.

Bruk et **undef** *usantResultat* for å definere et uttrykk som bare plotter grafen på et intervall.

when() er nyttig for å definere rekursive funksjoner.

$\text{when}(x < 0, x + 3) x = 5$	undef
-------------------------------------	-------

$\text{when}(n > 0, n \cdot \text{factorial}(n - 1), 1) \rightarrow \text{factorial}(n)$	Done
$\text{factorial}(3)$	6
3!	6

While *Betingelse**Blokk***EndWhile**

Utfører utsagnene i *Blokk* så lenge som *Betingelse* er sann.

Blokk kan enten være et enkelt utsagn eller en sekvens av utsagn som er adskilt med tegnet.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

```

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

```

<i>sum_of_recip</i> (3)	Done $\frac{11}{6}$
-------------------------	------------------------

X**xor (enten ...eller ...)**

BoolskUttr1 **xor** *BoolskUttr2* returnerer *Boolsk uttrykk*

true xor true	false
5 > 3 xor 3 > 5	true

BoolskListe1 **xor** *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrise1 **xor** *BoolskMatrise2* returnerer *Boolsk matrise*

Returnerer sann hvis *BoolskUttr1* er sant og *BoolskUttr2* er usant eller omvendt.

Returnerer usann hvis begge argumentene er sanne eller hvis begge er usanne. Returnerer et forenklet *Boolsk uttrykk* hvis ikke noen av argumentene kan avgjøres som sanne eller usanne.

Merk: Se or, side 108.

Heltall1 **xor** *Heltall2* ⇒ *heltall*

I heksades grunntall-modus:

Viktig: Null, ikke bokstaven O.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

I binær grunntall-modus:

Sammenlikner to reelle heltall bit-for-bit med en **xor**-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis en av bitene (men ikke begge) er 1; resultatet er 0 hvis begge bitene er 0 eller begge biter er 1. Returnert verdi representerer bit-resultatene og vises i grunntallmodus.

0b100101 xor 0b100

0b100001

Merk: Et binært innlegg kan bestå av opptil 64 siffer (i tillegg til prefikset 0b). Et heksadesimalt innlegg kan bestå av opptil 16 siffer.

Du kan skrive inn heltallene med hvilket som helst grunntall. Hvis du skriver inn en binær eller heksadesimal verdi, må du bruke hhv. prefiks 0b eller 0h. Uten slik prefiks blir heltall behandlet som desimalt (grunntall 10).

Hvis du oppgir et desimalt heltall som er for stort for et 64-bit binært tall med fortegn, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se **►Base2**, side 16.

Merk: Se **or**, side 108.

Z

zInterval

zInterval σ ,*Liste*[,*Frekv*[,*CNivå*]]

(Dataliste inndata)

zInterval σ , \bar{x} ,*n* [,*CNivå*]

(Summering statistikk inndata)

Beregner et *z* konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen (side 147).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall for et ukjent populasjonsgjennomsnitt

Utdata-variabel	Beskrivelse
stat. \bar{x}	Utvalgets gjennomsnitt av datasekvensen fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.sx	Utvalgets standardavvik
stat.n	Lengde av datasekvensen med utvalgsgjennomsnitt
stat. σ	Kjent populasjons standardavvik for datasekvens <i>Liste</i>

zInterval_1Prop

Katalog > 

zInterval_1Prop x, n [,CNivå]

Beregner et en-proporsjons z konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147).

x er et ikke-negativt heltall.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling
stat. \hat{p}	Beregnet andel (brøkdel) av suksesser
stat.ME	Feilmargin
stat.n	Antall utvalg i datasekvens

zInterval_2Prop

Katalog > 

zInterval_2Prop $x1, n1, x2, n2$ [,CNivå]

Beregner et to-proporsjons z konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147).

$x1$ og $x2$ er ikke-negative heltall.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling
stat. \hat{p} Diff	Beregnet differanse mellom andeler (brøkdeler)
stat.ME	Feilmargin
stat. \hat{p} 1	Beregnet andel av suksesser i utvalg 1
stat. \hat{p} 2	Beregnet andel av suksesser i utvalg 2
stat.n1	Utvalgsstørrelse i datasekvens 1
stat.n2	Utvalgsstørrelse i datasekvens 2

zInterval_2Samp

Katalog > 

zInterval_2Samp $\sigma_1, \sigma_2, \text{Liste1}, \text{Liste2}$
[,Frekv1[,Frekv2,[CNivå]]]

(Dataliste inndata)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}, n2$ [CNivå])

(Summering statistikk inndata)

Beregner et to-utvalgs z konfidensintervall.
En oversikt over resultatene lagres i
stat.results-variabelen. (Se side 147).

For informasjon om effekten av tomme
elementer i en liste, se "Tomme (åpne)
elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling
stat. $\bar{x}1 - \bar{x}2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat. $\bar{x}1$, stat. $\bar{x}2$	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat. $\sigma x1$, stat. $\sigma x2$	Utvalgets standardavvik for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antall utvalg i datasekvenser
stat.r1, stat.r2	Kjent populasjons standardavvik for datasekvens <i>Liste 1</i> og <i>Liste 2</i>

zTest $\mu_0, \sigma, \text{Liste}, [\text{Frekv}, \text{Hypot}]$

(Dataliste inndata)

zTest $\mu_0, \sigma, \bar{x}, n, \text{Hypot}$

(Summering statistikk inndata)

Utfører en z -test med frekvens *frekvlste*. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147).

Test $H_0: \mu = \mu_0$, mot ett av følgende:

For $H_a: \mu < \mu_0$, sett *Hypot*<0

For $H_a: \mu \neq \mu_0$ (standard), sett *Hypot*=0

For $H_a: \mu > \mu_0$, sett *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.z	$(\bar{x} - \mu_0) / (\sigma / \text{sqrt}(n))$
stat.P-Verdi	Minste sannsynlighet som null-hypotesen kan forkastes ved
stat. \bar{x}	Utvalgsgjennomsnitt av datasekvensen i <i>Liste</i>
stat.sx	Utvalgets standardavvik av datasekvensen. Returneres kun for inndata <i>Data</i> .
stat.n	Utvalgenes størrelse

zTest_1Prop

zTest_1Prop p_0, x, n, Hypot

Beregner en en-proporsjons z -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147).

x er et ikke-negativt heltall.

Test $H_0: p = p_0$ mot ett av følgende:

For $H_a: p > p_0$, sett *Hypot*>0

For $H_a: p \neq p_0$ (standard), sett *Hypot*=0

For $H_a: p < p_0$, sett *Hypot*<0

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 209).

Utdata-variabel	Beskrivelse
stat.p0	Hypotesisk populasjonsandel
stat.z	Standard normalverdi beregnet for andelen
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat. \hat{p}	Beregnet andel av suksesser
stat.n	Utvalgenes størrelse

zTest_2Prop $x1, n1, x2, n2[, Hypot]$

Beregner en to-proporsjons *z*-test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147).

$x1$ og $x2$ er ikke-negative heltall.

Test $H_0: p1 = p2$ mot ett av følgende:

For $H_a: p1 > p2$, sett *Hypot*>0

For $H_a: p1 \neq p2$ (standard), sett *Hypot*=0

For $H_a: p < p_0$, sett *Hypot*<0

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 209).

Utdata-variabel	Beskrivelse
stat.z	Standard normalverdi beregnet for differansen av andelene
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat. $\hat{p}1$	Beregnet andel av suksesser i utvalg 1
stat. $\hat{p}2$	Beregnet andel av suksesser i utvalg 2
stat. \hat{p}	Beregnet samlet andel av suksesser

Utdata-variabel	Beskrivelse
stat.n1, stat.n2	Antall utvalg som er tatt i forsøk 1 og 2

zTest_2Samp

Katalog > 

zTest_2Samp $\sigma_1, \sigma_2, \text{Liste1}, \text{Liste2}, [\text{Frekv1}$
 $[\text{,Frekv2}, \text{Hypot}]]]$

(Dataliste inndata)

zTest_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2, [\text{Hypot}]$

(Summering statistikk inndata)

Beregner en to-utvalgs z -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 147).

Test $H_0: \mu_1 = \mu_2$, mot ett av følgende:

For $H_a: \mu_1 < \mu_2$, sett *Hypot*<0

For $H_a: \mu_1 \neq \mu_2$ (standard), sett *Hypot*=0

For $H_a: \mu_1 > \mu_2$, *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 209).

Utdata-variabel	Beskrivelse
stat.z	Standard normalverdi beregnet for forskjellen i gjennomsnitt
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat. \bar{x} 1, stat. \bar{x} 2	Utvalgets gjennomsnitt av datasekvenser i <i>Liste1</i> og <i>Liste2</i>
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i <i>Liste 1</i> og <i>Liste2</i>
stat.n1, stat.n2	Utvalgenes størrelse

Symboler

+ (addere)

+ tast

Returnerer summen av de to argumentene.

56	56
56+4	60
60+4	64
64+4	68
68+4	72

$Liste1 + Liste2 \Rightarrow liste$

$Matrise1 + Matrise2 \Rightarrow matrise$

Returnerer en liste (eller matrise) som inneholder summene av tilsvarende elementer i $Liste1$ og $Liste2$ (eller $Matrise1$ og $Matrise2$).

Dimensjonene i argumentene må være like.

$15 + \{10, 15, 20\}$	$\{25, 30, 35\}$
$\{10, 15, 20\} + 15$	$\{25, 30, 35\}$

Merk: Bruk $+$ (prikk pluss) for å addere et uttrykk til hvert element.

$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
---	--

- (subtrahere)

- tast

$Liste1 - Liste2 \Rightarrow liste$

$Matrise1 - Matrise2 \Rightarrow matrise$

Subtraherer hvert element i $Liste2$ (eller $Matrise2$) fra tilsvarende element i $Liste1$ (eller $Matrise1$), og returnerer resultatene.

Dimensjonene i argumentene må være like.

$15 - \{10, 15, 20\}$	$\{5, 0, -5\}$
$\{10, 15, 20\} - 15$	$\{-5, 0, 5\}$

Merk: Bruk $-$ (prikk minus) for å subtrahere et uttrykk fra hvert element.

$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$
---	--

•(multiplisere)

 tast

Returnerer produktet av de to argumentene.

$Liste1 \cdot Liste2 \Rightarrow liste$

Returnerer en liste som inneholder produktene av samsvarende elementer i $Liste1$ og $Liste2$.

Dimensjonene i listene må være like.

$Matrise1 \cdot Matrise2 \Rightarrow matrise$

Returnerer matriseproduktet av $Matrise1$ og $Matrise2$.

Antallet kolonner i $Matrise1$ må være likt antallet rader i $Matrise2$.

Merk: Bruk \cdot (prikk multipliser) for å multiplisere et uttrykk med hvert element.

/ (divider)

 tast

Merk: Se også **Brøk-sjablon**, side 1.

$Liste1 / Liste2 \Rightarrow liste$

Returnerer en liste som inneholder kvotientene av $Liste1$ dividert med $Liste2$.

$$\frac{\{1, 2, 3\}}{\{4, 5, 6\}} \quad \left\{ 0.25, \frac{2}{5}, \frac{1}{2} \right\}$$

Dimensjonene i listene må være like.

Merk: Bruk $/$ (prikk divider) for å dividere et uttrykk med hvert element.

^ (potens)

 tast

$Liste1 \wedge Liste2 \Rightarrow liste$

Returnerer det første argument opphøyd i det andre argumentet.

Merk: Se også **EkspONENT-sjablon**, side 1.

For en liste, returneres elementene i $Liste1$ opphøyd i tilsvarende elementer i $Liste2$.

^ (potens)



I reell grunnmengde bruker brøkpotens som har forkortet eksponent med oddetall i nevner en rell forgreining i motsetning til hovedforgreining for kompleks modus.

kvadratMatrise1 ^ *heltall* ⇒ *matrise*

Returnerer *kvadratMatrise1* opphøyd i *heltall* -potens.

kvadratMatrise1 må være en kvadratmatrise.

Hvis *heltall* = -1, beregnes invers matrise.

Hvis *heltall* < -1, beregnes invers matrise opphøyd i en korrekt positiv potens.

$$\{1,2,3,4\}^{-2} \quad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \quad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

x² (kvadrat)



Returnerer kvadratet av argumentet.

*Liste1*² ⇒ *liste*

Returnerer en liste med kvadrater av elementene i *Liste1*.

*kvadratMatrise1*² ⇒ *matrise*

Returnerer matrisens kvadrat av *kvadratMatrise1*. Dette er ikke det samme som å beregne kvadratet av hvert element. Bruk ² for å beregne kvadratet av hvert element.

$$4^2 \quad 16$$

$$\{2,4,6\}^2 \quad \{4,16,36\}$$

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2 \quad \begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^{\wedge 2} \quad \begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$$

.+ (prikk adder)





taster

Matrise1 .+ *Matrise2* ⇒ *matrise*

Matrise1 .+ *Matrise2* returnerer en matrise som er summen av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

.- (prikk subt.)

  taster

Matrise1 .- *Matrise2* \Rightarrow *matrise*

Matrise1 .- *Matrise2* returnerer en matrise som er differansen mellom hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

.• (prikk mult.)

  taster

Matrise1 .• *Matrise2* \Rightarrow *matrise*

Matrise1 .• *Matrise2* returnerer en matrise som er produktet av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.



. / (prikk divider)

  taster

Matrise1 ./ *Matrise2* \Rightarrow *matrise*

Matrise1 ./ *Matrise2* returnerer en matrise som er kvotient av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

.^ (prikk potens)

  taster

Matrise1 .^ *Matrise2* \Rightarrow *matrise*

Matrise1 .^ *Matrise2* returnerer en matrise, der hvert element i *Matrise2* er eksponenten for samsvarende element i *Matrise1*.

-(negere)

 tast

-*Liste1* \Rightarrow *liste*

-*Matrise1* \Rightarrow *matrise*

Returnerer argumentets negasjon.

For en liste eller matrise returneres alle elementene negert.

I binær grunntall-modus:

Viktig: Null, ikke bokstaven O

```
-0b100101
0b11111111111111111111111111111111▶
```


For å se hele resultatet, trykk på ▲ og bruk så ◀ og ▶ for å bevege markøren.

- (negere)

 **tast**

Hvis argumentet er et binært eller heksadesimalt heltall, gir negasjonen komplementet til to.

% (prosent)

  **taster**

Liste1 % \Rightarrow *liste*

Matrise1 % \Rightarrow *matrise*

argument

Returterer **100**


For en liste eller matrise, returneres en liste eller matrise med hvert element dividert med 100.

Merk: For å tvinge frem et tilnærmet desimalresultat,

Grafregner: Trykk på  .

Windows®: Trykk på **Ctrl+Enter**.

Macintosh®: Trykk på **⌘+Enter**.

iPad®: Hold på **enter**, og velg .

13% 0.13

$\{\{1,10,100\}\}\%$ $\{0.01,0.1,1.\}$

= (er lik)

 **tast**

Uttr1 = Uttr2 \Rightarrow *Boolsk uttrykk*

Liste1 = Liste2 \Rightarrow *Boolsk liste*

Matrise1 = Matrise2 \Rightarrow *Boolsk matrise*

Returterer sann hvis *Uttr1* er bestemt å være lik *Uttr2*.

Returterer usann hvis *Uttr1* er bestemt å være ulik *Uttr2*.

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktboken.

Eksempel på funksjon som bruker matematiske testsymboler: =, ≠, <, ≤, >, ≥

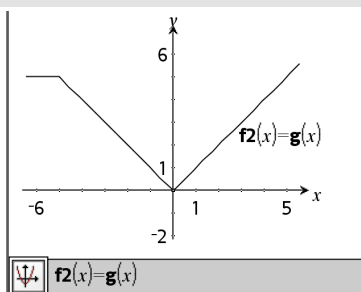
```
Define g(x)=Func
  If x<=5 Then
    Return 5
  ElseIf x>5 and x<0 Then
    Return -x
  ElseIf x≥0 and x≠10 Then
    Return x
  ElseIf x=10 Then
    Return 3
  EndIf
EndFunc
```

Done

Resultat av grafisk fremstilling g(x)

= (er lik)

 tast



≠ (ulik)

  taster

$Uttr1 \neq Uttr2 \Rightarrow$ Boolsk uttrykk

Se "=" (er lik) eksempel.

$Liste1 \neq Liste2 \Rightarrow$ Boolsk liste

$Matrise1 \neq Matrise2 \Rightarrow$ Boolsk matrise

Returnerer sann hvis $Uttr1$ er bestemt å være ulik $Uttr2$.



Returnerer usann hvis $Uttr1$ er bestemt å være lik $Uttr2$.

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive /=

< (mindre enn)

  taster

$Uttr1 < Uttr2 \Rightarrow$ Boolsk uttrykk

Se "=" (er lik) eksempel.



$Liste1 < Liste2 \Rightarrow$ Boolsk liste

$Matrise1 < Matrise2 \Rightarrow$ Boolsk matrise

Returnerer sann hvis $Uttr1$ er bestemt å være mindre enn $Uttr2$.

Returnerer usann hvis $Uttr1$ er bestemt å være større enn eller lik $Uttr2$.

< (mindre enn)

  taster

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

≤ (mindre enn eller lik)

  taster

$Uttr1 \leq Uttr2 \Rightarrow$ *Boolsk uttrykk*

Se "=" (er lik) eksempel.

$Liste1 \leq Liste2 \Rightarrow$ *Boolsk liste*

$Matrise1 \leq Matrise2 \Rightarrow$ *Boolsk matrise*

Returnerer sann hvis $Uttr1$ er bestemt å være mindre enn eller lik $Uttr2$.

Returnerer usann hvis $Uttr1$ er bestemt å være større enn $Uttr2$.

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive <=

> (større enn)

  taster

$Uttr1 > Uttr2 \Rightarrow$ *Boolsk uttrykk*

Se "=" (er lik) eksempel.

$Liste1 > Liste2 \Rightarrow$ *Boolsk liste*

$Matrise1 > Matrise2 \Rightarrow$ *Boolsk matrise*

Returnerer sann hvis $Uttr1$ er bestemt å være større enn $Uttr2$.

Returnerer usann hvis $Uttr1$ er bestemt å være mindre enn eller lik $Uttr2$.

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

≥ (større enn eller lik med)

ctrl = taster

$Uttr1 \geq Uttr2 \Rightarrow$ *Boolsk uttrykk*

Se “=” (er lik) eksempel.

$Liste1 \geq Liste2 \Rightarrow$ *Boolsk liste*

$Matrise1 \geq Matrise2 \Rightarrow$ *Boolsk matrise*

Returnerer sann hvis $Uttr1$ er bestemt å være større enn eller lik $Uttr2$.

Returnerer usann hvis $Uttr1$ er bestemt å være mindre enn eller lik $Uttr2$.

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive >=

\Rightarrow (logisk implikasjon)

ctrl = -taster

$BoolskUttr1 \Rightarrow BoolskUttr2$ returnerer *Boolsk uttrykk*

$5 > 3$ or $3 > 5$ true

$5 > 3 \Rightarrow 3 > 5$ false

$BoolskListe1 \Rightarrow BoolskListe2$ returnerer *Boolsk liste*

3 or 4 7

$3 \Rightarrow 4$ -4

$BoolskMatrise1 \Rightarrow BoolskMatrise2$ returnerer *Boolsk matrise*

{1,2,3} or {3,2,1} {3,2,3}

{1,2,3} \Rightarrow {3,2,1} {-1,-1,-3}

$Heltall1 \Rightarrow Heltall2$ returnerer *Heltall*

Behandler uttrykket **not <argument1> or <argument2>** og returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive =>

⇔ (logisk dobbel implikasjon, XNOR)

ctrl [=]-taster

BoolskUttr1 ⇔ *BoolskUttr2* returnerer *Boolsk uttrykk*

5>3 xor 3>5 true

BoolskListe1 ⇔ *BoolskListe2* returnerer *Boolsk liste*

5>3 ⇔ 3>5 false

BoolskMatrise1 ⇔ *BoolskMatrise2* returnerer *Boolsk matrise*

3 xor 4 7

3 ⇔ 4 -8

Heltall1 ⇔ *Heltall2* returnerer *Heltall*

{1,2,3} xor {3,2,1} {2,0,2}

{1,2,3} ⇔ {3,2,1} {-3,-1,-3}

Returnerer negasjon av en **XOR** Boolsk handling på de to argumentene.
Returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive <=>

! (fakultet)

[?!>] tast

Liste1! ⇒ *liste*

5! 120

Matrise1! ⇒ *matrise*

{{5,4,3}}! {120,24,6}

Returnerer argumentets fakultet.

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ $\begin{pmatrix} 1 & 2 \\ 6 & 24 \end{pmatrix}$

For en liste eller matrise, returneres en liste eller matrise av elementenes fakulteter.

& (legg til)

ctrl [📖] taster

Streng1 & *Streng2* ⇒ *streng*

"Hello "&"Nick" "Hello Nick"

Returnerer en tekststreng som er *Streng2* lagt til *Streng1*.

d () (derivert)

Katalog > 

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **derivative** (...).

$\sqrt{}$ (kvadratrot)

  **taster**

$\sqrt{Liste1} \Rightarrow liste$

Returnerer kvadratroten til argumentet.

For en liste, returneres kvadratroten til alle elementene i *Liste1*.

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **sqrt** (...)

Merk: Se også **Kvadratrot-sjablon**, side 1.

$\Pi()$ (prodSeq)

Katalog > 

$\Pi(Uttr1, Var, Nedre, Øvre) \Rightarrow uttrykk$

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **prodSeq** (...).

Finner *Uttrykk1* for hver verdi av *Var* fra *Nedre* til *Øvre*, og returnerer produktet av resultatene.

Merk: Se også **Produkt-sjablon (Π)**, side 5.

$\Pi(Uttr1, Var, Nedre, Nedre-1) \Rightarrow 1$

$\Pi(Uttr1, Var, Nedre, Øvre) \Rightarrow 1/\Pi(Uttr1, Var, Øvre+1, Nedre-1)$ hvis $Øvre < Nedre - 1$

Produktformlene som er brukt er hentet fra følgende referanse:

Ronald L. Graham, Donald E. Knuth, og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\frac{1}{\prod_{k=4}^3 \binom{1}{k}} \quad 1$$

$$\frac{1}{\prod_{k=4}^1 \binom{1}{k}} \quad 6$$

$$\frac{1}{\prod_{k=4}^1 \binom{1}{k}} \cdot \frac{4}{\prod_{k=2}^4 \binom{1}{k}} \quad \frac{1}{4}$$

$\Sigma(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow \text{uttrykk}$

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **sumSeq(...)**.

Behandler *Uttrykk1* for hver verdi av *Var* fra *Nedre* til *Øvre*, og returnerer summen av resultatene.

Merk: Se også **Sum-sjablon**, side 5.

$\Sigma(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Nedre}-1) \Rightarrow 0$

$\Sigma(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow -\Sigma(\text{Uttr1}, \text{Var}, \text{Øvre}+1, \text{Nedre}-1)$ hvis $\text{Øvre} < \text{Nedre}-1$

Summeringsformlene som er brukt er hentet fra følgende referanse:

Ronald L. Graham, Donald E. Knuth, og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\sum_{k=4}^3 (k) = 0$$

$$\sum_{k=4}^1 (k) = -5$$

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) = 4$$

 $\Sigma\text{Int}()$

$\Sigma\text{Int}(\text{NPmt1}, \text{NPmt2}, \text{N}, \text{I}, \text{PV}, [\text{Pmt}], [\text{FV}], [\text{PpY}], [\text{CpY}], [\text{PmtAt}], [\text{avrundVerdi}]) \Rightarrow \text{verdi}$

$\Sigma\text{Int}(1, 3, 12, 4, 75, 20000, \dots, 12, 12) = -218.11$

 ΣInt

$(\text{NPmt1}, \text{NPmt2}, \text{amortTabell}) \Rightarrow \text{verdi}$

Amortiseringsfunksjon som beregner rentesummen i løpet av en spesifisert rekke av betalinger.

NPmt1 og *NPmt2* definerer start- og sluttgrensene for betalingsrekken.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 162.

- Hvis du utelater *Pmt*, grunninnstilles den til $\text{Pmt}=\text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.

<i>tbt</i> :=amortTbl(12,12,4,75,20000,,,12,12)			
0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

$\Sigma\text{Int}(1, 3, \text{tbt}) = -218.11$

- Hvis du utelater FV , grunninnstilles den til $FV=0$.
- Grunninnstillingene for PpY , CpY og $PmtAt$ er de samme som for TVM-funksjonene.

avrundVerdi spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

$\Sigma\text{Int}(NPmt1, NPmt2, amortTabell)$ beregner rentesummen basert på amortiseringstabell *amortTabell*. Argumentet *amortTabell* må være en matrise i den form som er beskrevet under **amortTbl()**, side 7.

Merk: Se også $\Sigma\text{Prn}()$, nedenfor, og **Bal()**, side 15.

 $\Sigma\text{Prn}()$

$\Sigma\text{Prn}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [avrundVerdi]) \Rightarrow \text{verdi}$

$\Sigma\text{Prn}(1, 3, 12, 4.75, 20000, \dots, 12, 12)$ -4911.47

 ΣPrn

$(NPmt1, NPmt2, amortTable) \Rightarrow \text{verdi}$

Amortiseringsfunksjon som beregner summen av hovedbetalinger i løpet av en spesifisert rekke av betalinger.

$NPmt1$ og $NPmt2$ definerer start- og sluttgrensene for betalingsrekken.

$N, I, PV, Pmt, FV, PpY, CpY$ og $PmtAt$ er beskrevet i tabellen med TVM-argumenter, side 162.

- Hvis du utelater Pmt , grunninnstilles den til $Pmt=\text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.
- Hvis du utelater FV , grunninnstilles den til $FV=0$.
- Grunninnstillingene for PpY , CpY og $PmtAt$ er de samme som for TVM-funksjonene.

tbl:=amortTbl(12, 12, 4.75, 20000, \dots, 12, 12)

0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

$\Sigma\text{Prn}(1, 3, \text{tbl})$ -4911.47

avrundVerdi spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

$\Sigma\text{Prn}(\text{NPmt1}, \text{NPmt2}, \text{amortTabell})$ beregner summen av hovedbetalinger basert på amortiseringstabell *amortTabell*. Argumentet *amortTabell* må være en matrise i den form som er beskrevet under **amortTbl()**, side 7.

Merk: Se også $\Sigma\text{Int}()$, over, og **Bal()**, side 15.

(Indir.ref)

  **taster**

varNavnStreng

Oppretter eller refererer til variabelen xyz.

Refererer til variabelen med navnet *varNavnStreng*. På denne måten kan du bruke strenger for å opprette variabelnavn "innenfra" en funksjon.

$10 \rightarrow r$	10
"r" $\rightarrow s1$	"r"
#s1	10

Returnerer verdien av variabelen (r) som har et navn som er lagret i variabel s1.

E (vitenskapelig tallnotasjon)

 **tast**

mantissaEksponent

23000.	23000.
Legger inn et tall i vitenskapelig fremstilling. Tallet blir tolket som en <i>mantissa</i> $\times 10^{\text{eksponent}}$.	23000000000.+4.1E15
	4.1E15
	$3 \cdot 10^4$
	30000

Tips: Hvis du vil legge inn en potens av 10 uten å forårsake desimale verdier i resultatet, bruk 10^{heltall} .

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive **EE**. Eksempel: Skriv **2.3EE4** for å legge inn $2.3E4$.

g (gadian)

 **tast**

Uttr1g \Rightarrow *Uttrykk*

l grader, gadian eller radian modus:

g (gradian)

1 tast

$Uttr1g \Rightarrow Uttrykk$

$Liste1g \Rightarrow liste$

$Matrise1g \Rightarrow matrise$

Denne funksjonen gir deg en mulighet til å spesifisere en vinkel i gradianer mens du er i grader- eller gradian-modus.

I radian-vinkelmodus, multipliseres $Uttr1$ med $\pi/200$.

I grader-vinkelmodus, multipliseres $Uttr1$ med $g/100$.

I gradian modus, returneres $Uttr1$ uendret.

Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @g.

r (radian)

1 tast

$Liste1r \Rightarrow liste$

I grader, gradian eller radian modus:

$Matrise1r \Rightarrow matrise$

Denne funksjonen gir deg en mulighet til å spesifisere en gradian vinkel mens du er i grader- eller radian-modus.

I grader-vinkelmodus, multipliseres argumentet med $180/\pi$.

I radian-vinkelmodus, returneres argumentet uendret.

I gradian modus, multipliseres argumentet med $200/\pi$.

Tips: Bruk r hvis du vil tvinge radianer inn en funksjonsdefinisjon uavhengig av hvilken modus som er i bruk når du bruker funksjonen.

Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @r.

° (grader)

 tast

ListeI°⇒liste

I grader, gradian eller radian modus:

MatriseI°⇒matrise

I radian modus:

Denne funksjonen gir deg en mulighet til å spesifisere en grader-vinkel mens du er i gradian eller radian modus.

$$\cos\left\{\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right\}$$

$$\{1, 0.707107, 0., 0.864976\}$$

I radian-vinkelmodus, multipliseres argumentet med $\pi/180$.

I grader-vinkelmodus, returneres argumentet uendret.

I gradian-vinkelmodus, multipliseres argumentet med 10/9.

Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @d.

°, ', " (grader/minutter/sekunder)

  taster

gg°mm'ss.ss"⇒Uttrykk

I Grader-vinkelmodus:

gg Et positivt eller negativt tall

$$25^\circ 13' 17.5'' \quad 25.2215$$

mm Et ikke-negativt tall

$$25^\circ 30' \quad \frac{51}{2}$$

ss.ss Et ikke-negativt tall

Returnerer $gg+(mm/60)+(ss.ss/3600)$.

Dette grunntall -60-formatet lar deg:

- Sette inn en vinkel i grader/minutter/sekunder uten hensyn til aktuell vinkelmodus.
- Sette inn tid, som timer/minutter/sekunder.

Merk: Sett to apostrofer (") etter ss.ss, ikke et anførselstegn (').

∠ (vinkel)

  taster

[Radius,∠θ_Vinkel]⇒vektor

I radian modus og vektorformat innstilt på: rektangulær

(polar inndata)

∠ (vinkel)

ctrl  taster

[Radius, ∠θ_Vinkel, Z_Koordinat] ⇒ vektor

sylindrisk

(sylindrisk inndata)

sfærisk

[Radius, ∠θ_Vinkel, ∠θ_Vinkel] ⇒ vektor

(sfærisk inndata)

Returnerer koordinater som en vektor, avhengig av vektorformatets modusinnstilling: rektangulær, sylindrisk eller sfærisk.

Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @<.

(Størrelse ∠ Vinkel) ⇒ kompleks Verdi

I radian-vinkelmodus og rektangulært, komplekst format:

(polar inndata)

$$5+3\cdot i - \left(10 \angle \frac{\pi}{4}\right) = -2.07107 - 4.07107\cdot i$$

Setter inn en kompleks verdi i ($r\angle\theta$) polar form. *Vinkelen* tolkes avhengig av aktuell vinkelmodus-innstilling.

_ (senket strek som et tomt element)

Se "Tomme (åpne) elementer" |, side 209.

10^()

Katalog > 

10^ (Liste1) ⇒ liste

Returnerer 10 opphøyd i argumentets potens.

For en liste, returneres 10 opphøyd i elementenes potens i *Liste1*.

10^(kvadratMatrise) ⇒ kvadratMatrise

Returnerer 10 opphøyd i potensen av *kvadratMatrise1*. Dette er ikke det samme som å beregne 10 opphøyd i potens av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} = \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

*kvadratMatrise*1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

^-1(resiprok)

Liste1 $\wedge^{-1} \Rightarrow$ liste

Returnerer den inverse verdien av et argument.

For en liste, returneres den inverse verdien av elementene i *Liste*1.

*kvadratMatrise*1 $\wedge^{-1} \Rightarrow$ *kvadratMatrise*

Returnerer den inverse verdien av *kvadratMatrise*1.

*kvadratMatrise*1 må være en ikke-singulær kvadratisk matrise.

| (begrensningsoperator)

Uttr | *BoolskUttr*1 [and *BoolskUttr*2]...

Uttr | *BoolskUttr*1 [or *BoolskUttr*2]...

Begrensningssymbolet ("|") fungerer som en binær operator. Operanden til venstre for | er et uttrykk. Operanden til høyre for | spesifiserer en eller flere relasjoner som kan ha innvirkning på forenklingen av uttrykket. Flere forbindelser etter | må sammenføres av logiske "and" eller "or" operatorene.

Med begrensningsoperatoren har du tre utgangstyper av funksjonalitet:

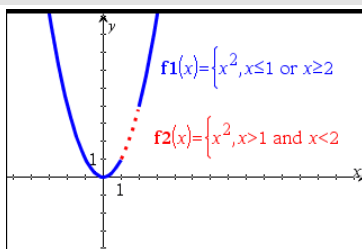
- Erstatninger
- Intervallbegrensninger
- Eksklusjoner

En erstatning har form som en ligning, som $x=3$ eller $y=\sin(x)$. For at den skal være mest effektiv, bør den venstre siden være en enkel variabel. Uttr | *Variabel* = *verdi* vil erstatte *verdi* for hver forekomst av *Variabel* i Uttr.

| (begrenningsoperator)

ctrl  -taster

Intervallbegrensninger tar form som en eller flere ulikheter som er føyd sammen av logiske "and" eller "or" operatore. En intervallbegrensning tillater også forenkling som ellers kan være ugyldig eller ikke mulig å beregne.



Eksklusjoner bruker relasjons-operatoren "ulik" (\neq or \neq) for å ekskludere en spesifikk verdi fra å komme i betraktning.

$$\text{solve}(x^2 - 1 = 0, x) | x \neq 1 \quad x = -1$$

→ (lagre)

ctrl var tast

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive =: som en snarvei. Eksempel: Skriv `pi/4 =: minvar`.

:= (tildele)

ctrl  taster

Var := Liste

Var := Matrise

Funksjon(Param1,...) := Utr

Funksjon(Param1,...) := Liste

Funksjon(Param1,...) := Matrise

© (kommentar)

  **taster**

© [tekst]

© fremstiller *tekst* som en kommentarlinje som lar deg kommentere funksjoner og programmer som du oppretter.

© kan plasseres ved begynnelsen eller hvor som helst på linjen. Alt som er til høyre for ©, til slutten av linjen, er kommentaren.

Merk for å legge inn eksemplet: For anvisninger om hvordan du legger inn flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produktåndboken.

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

© *Declare variables*

Local *i,result*

result:=0

For *i,1,n,1* ©Loop *n times*

result:=result+i²

EndFor

Return *result*

EndFunc

Done

$g(3)$

14

Ob, Oh

  **taster,   **taster****

Ob *binærTall*

I desimalt grunntall-modus:

0b10+0hF+10

27

Oh *heksadesimalTall*

Markerer hhv. et binært eller heksadesimalt tall. For å sette inn et binært eller heksadesimalt tall må du sette inn prefikset Ob eller Oh uavhengig av grunninnstillingsmodus. Uten prefiks blir et tall behandlet som et desimaltall (grunntall 10).

I binær grunntall-modus:

0b10+0hF+10

0b11011

Resultatene vises i forhold til grunntall-modusen.

I heksades grunntall-modus:

0b10+0hF+10

0h1B

TI-Nspire™ CX II – Tegnekommandoer

Dette er et tilleggsdokument for TI-Nspire™-referanseguiden og TI-Nspire™ CAS-referanseguiden. Alle TI-Nspire™ CX II-kommandoene vil bli innlemmet og publisert i versjon 5.1 av TI-Nspire™-referanseguiden og TI-Nspire™ CAS-referanseguiden.

Grafikkprogrammering

Nye kommandoer er lagt til på TI-Nspire™ CX II-grafregnerne og TI-Nspire™-skrivebordsprogrammer for grafikkprogrammering.

TI-Nspire™ CX II-grafregnerne vil bytte til denne grafikkmodusen mens den utfører grafikkkommandoer og bytte tilbake til konteksten der programmet ble utført etter at programmet er ferdig.

Skjermen viser «Kjører ...» i topplinjen mens programmet utføres. Det viser «Ferdig» når programmet er fullført. Ethvert tastetrykk tar systemet ut av grafikkmodus.

- Overgangen til grafikkmodus utløses automatisk når en av Draw (grafikk)-kommando oppstår under utførelsen av TI-Basic-programmet.
- Denne overgangen vil bare skje når du kjører et program fra kalkulator; i et dokument eller kalkulator i kladdeark.
- Overgangen ut av grafikkmodus skjer ved avslutning av programmet.
- Grafikkmodusen er bare tilgjengelig på TI-Nspire™ CX II-grafregnerne og skrivebordsvisningen for TI-Nspire™ CX II CAS-grafregneren. Dette betyr at det ikke er tilgjengelig i datamaskindokumentvisningen på skrivebordet eller i iOS.
 - Hvis det oppdages en grafikkkommando mens du kjører et TI-Basic-program fra feil kontekst, vises en feilmelding og TI-Basic-programmet avsluttes.

Grafikkskjerm

Grafikkskjermbildet vil inneholde en overskrift øverst på skjermen som det ikke kan skrives i av grafikkkommandoer.

Tegningsområdet for grafikkskjermbildet vil bli slettet (farge = 255, 255, 255) når grafikkskjermbildet initialiseres.

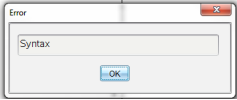
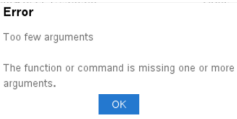
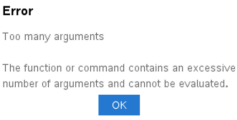
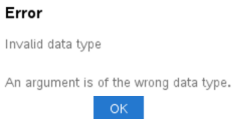
Grafikkskjerm	Standard
Høyde	212
Bredde	318
Farge	hvit: 255,255,255

Standardvisning og innstillinger

- Statusikonene i topplinjen (batteristatus, trykk-for-å-teste-status, nettverksindikator etc.) vil ikke være synlige når et grafikkprogram kjører.
- Standard tegnefarge: Black (0,0,0)
- Standard pennestil – normal, glatt
 - Tykkelse: 1 (tynn), 2 (normal), 3 (tykkest)
 - Stil 1 (jevn), 2 (stiplet), 3 (punkt)
- Alle tegnekommandoer vil bruke de nåværende innstillingene for farge og penn; enten standardverdier eller de som ble angitt via TI-Basic-kommandoer.
- Skrivestilen er fastsatt og kan ikke endres.
- Eventuelle utdata til grafikkskjerm bildet vil bli tegnet i et klippevindu som er på størrelse med tegneområdet på grafikkskjermen. Tegnete utdata som strekker seg utenfor dette klippede grafikkskjermområdet, tegnes ikke. Ingen feilmelding vises.
- Alle x,y-koordinater angitt for tegnekommandoer er definert slik at 0,0 er øverst til venstre i grafikkskjermens tegneområde.
 - **Unntak:**
 - **DrawText** bruker koordinatene som nedre venstre hjørne av markeringsrammen for teksten.
 - **SetWindow** bruker nedre venstre hjørne av skjermen
- Alle parametere for kommandoene kan gis som uttrykk, som evaluerer til et tall som deretter rundes av til nærmeste heltall.

Feilmeldinger på grafikkskjerm

Hvis valideringen mislykkes, vises en feilmelding.

Feilmelding	Beskrivelse	Vise
Feil Syntaks	Dersom syntakskontrollen oppdager syntaksfeil, vises en feilmelding, og kontrollen prøver å plassere markøren nær den første feilen så du kan korrigere den.	
Feil For få argumenter	Funksjonen eller kommandoen mangler et eller flere argumenter	
Feil For mange argumenter	Funksjonen eller kommandoen inneholder for mange argumenter og kan ikke behandles.	
Feil Ugyldig datatype	Et argument er av feil datatype.	

Ugyldige kommandoer i grafikkmodus

Noen kommandoer er ikke tillatt når programmet bytter til grafikkmodus. Hvis disse kommandoene oppstår i grafikkmodus, vises feil og programmet avsluttes.

Forbudt kommando	Feilmelding
Forespør	Forespørselen kan ikke utføres i grafikkmodus
ForespørStr	RequestStr kan ikke utføres i grafikkmodus
Tekst	Tekst kan ikke utføres i grafikkmodus

Kommandoene som skriver ut tekst til kalkulatoren – **disp** og **dispAt** – vil være støttede kommandoer i grafikkonteksten. Teksten fra disse kommandoene vil bli sendt til Kalkulator-skjermbildet (ikke på Grafikk) og vil være synlig etter at programmet er avsluttet og systemet skifter tilbake til kalkulator-appen

Slett

Tøm *x, y, bredde, høyde*

Slett

Tømmer hele skjermen hvis ingen parametere er spesifisert.

Tømmer hele skjerm

Hvis *x, y, bredde* og *høyde* er spesifisert, blir rektangelet som er definert av parametrene slettet.

Clear 10,10,100,50

Tømmer et rektangelområde med øverste venstre hjørne på (10, 10), med bredde 100 og høyde 50

DrawArc

 Katalog > 
 CXII

DrawArc $x, y, \text{bredde}, \text{høyde}, \text{startAngle}, \text{arcAngle}$

Tegn en bue i det definerte avgrensende rektangelet med de angitte start- og buevinklene.

x, y : øvre venstre koordinat for avgrensende rektangel

bredde, høyde: dimensjoner av avgrensende rektangel

«arc angle» definerer vinkelåpningen for buen.

Disse parametere kan gis som uttrykk, som evaluerer til et tall som deretter rundes av til nærmeste heltall.

DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180



Se også: [FillArc](#)

DrawCircle

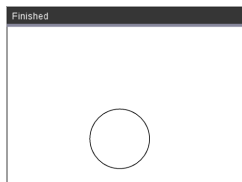
 Katalog > 
 CXII

DrawCircle x, y, radius

x, y : koordinat for sentrum

radius: radiusen på sirkelen

DrawCircle 150,150,40



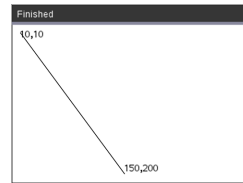
Se også: [FillCircle](#)

DrawLine $x1, y1, x2, y2$ Tegn en linje fra $x1, y1, x2, y2$.

Uttrykk som evaluerer til et tall som deretter rundes av til nærmeste heltall.

Skjermgrenser: Hvis de angitte koordinatene fører til at en del av linjen tegnes utenfor grafikkskjermen, blir den delen av linjen klippet og ingen feilmelding vises.

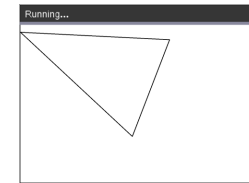
DrawLine 10,10,150,200

**DrawPoly**

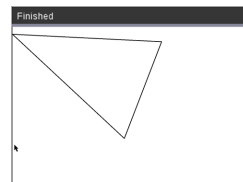
Kommandoene har også to varianter:

DrawPoly $xlist, ylist$

eller

DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$ **Merk:** DrawPoly $xlist, ylist$ Form vil koble $x1, y1$ til $x2, y2$, $x2, y2$ til $x3, y3$ og så videre.**Merk:** DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$ xn, yn vil **IKKE** kobles til $x1, y1$ automatisk.Uttrykk som evaluerer til en liste over reelle flyttall
 $xlist, ylist$ Uttrykk som evaluerer til ett enkelt reelt flyttall
 $x1, y1...xn, yn$ = koordinater for polygonhjørnene $xlist:={0,200,150,0}$ $ylist:={10,20,150,10}$ DrawPoly $xlist,ylist$ 

DrawPoly 0,10,200,20,150,150,0,10



Merk: DrawPoly: Inndata-størrelsesdimensjoner (bredde/høyde) i forhold til tegnede linjer.

Linjene tegnes i en avgrensningsboks rundt den angitte koordinaten og dimensjoneres slik at den faktiske størrelsen på den tegnede polygon vil være større enn det bredden og høyden indikerer.

Se også: [FillPoly](#)

DrawRect

DrawRect *x, y, bredde, høyde*

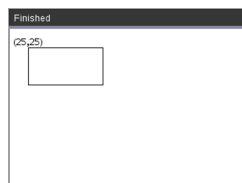
x, y: øvre venstre koordinat for rektangel

bredde, høyde: bredde og høyde for rektangel (rektangel tegn rett ned og til høyre fra startkoordinat).

Merk: Linjene tegnes i en avgrensningsboks rundt den angitte koordinaten og dimensjoneres slik at den faktiske størrelsen på det tegnede rektangelet vil være større enn det bredden og høyden indikerer.

Se også: [FillRect](#)

DrawRect 25,25,100,50



DrawText

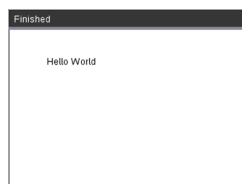
DrawText *x, y, exprOrString1*
[,exprOrString2]...

x, y: koordinat for tekstutdata

Tegner teksten i *exprOrString* på den angitte *x, y*-koordinatposisjonen.

Reglene for *exprOrString* er de samme som for **Disp** – **DrawText** kan ta flere argumenter.

DrawText 50,50,«Hello World»



FillArc

 Katalog >  CXII

FillArc x, y , bredde, høyde, startAngle, arcAngle

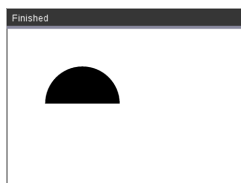
x, y : øvre venstre koordinat for avgrensede rektangel

Tegn og fyll en bue i det definerte avgrensede rektangelet med de angitte start- og buevinklene.

Standard fyllfarge er svart. Fyllfargen kan angis av [SetColor](#)-kommando

«arc angle» definerer vinkelåpningen for buen

FillArc 50,50,100,100,0,180



FillCircle

 Katalog >  CXII

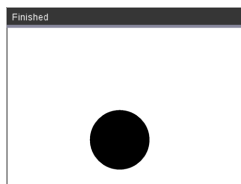
FillCircle x, y , radius

x, y : koordinat for sentrum

Tegn og fyll en sirkel på det angitte senteret med den angitte radiusen.

Standard fyllfarge er svart. Fyllfargen kan angis av [SetColor](#)-kommando.

FillCircle 150,150,40



Her!

FillPoly

 Katalog >  CXII

FillPoly $xlist, ylist$

eller

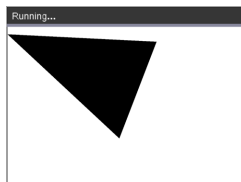
FillPoly $x1, y1, x2, y2, x3, y3...xn, yn$

Merk: Linjen og fargen er spesifisert av [SetColor](#) og [SetPen](#)

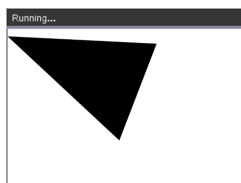
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

FillPoly $xlist,ylist$



FillPoly 0,10,200,20,150,150,0,10

**FillRect****FillRect** *x, y, bredde, høyde**x, y*: øvre venstre koordinat for rektangel*bredde, høyde*: bredde og høyde for rektangelTegn og fyll et rektangel med øverste venstre hjørne ved koordinaten spesifisert av (x,y) Standard fyllfarge er svart. Fyllfargen kan angis av [SetColor](#)-kommandoen**Merk:** Linjen og fargen er spesifisert av [SetColor](#) og [SetPen](#)

FillRect 25,25,100,50



G

getPlatform()

Katalog > 
CXII

getPlatform()

getPlatform()

"dt"

Returnerer:

«dt» på skrivebordsprogrammer

«hh» på TI-Nspire™ CX-grafregnere

«ios» på TI-Nspire™ CX iPad®-app

PaintBuffer

Tegn grafikkbuffer på skjerm

Denne kommandoen brukes sammen med UseBuffer for å øke visningshastigheten på skjermen når programmet genererer flere grafiske objekter.

UseBuffer

```
For n,1,10
```

```
x:=randInt(0,300)
```

```
y:=randInt(0,200)
```

```
radius:=randInt(10,50)
```

```
Wait 0,5
```

```
DrawCircle x,y,radius
```

```
EndFor
```

```
PaintBuffer
```

Dette programmet viser alle de 10 sirklene samtidig.

Hvis kommandoen «UseBuffer» fjernes, vises hver sirkel slik den er tegnet.

Se også: [UseBuffer](#)

PlotXY $x, y, form$

x, y koordinat for plottform

form: skriv inn et tall mellom 1 og 13 for å spesifisere formen

- 1 – Fylt sirkel
- 2 – Tom sirkel
- 3 – Fylt firkant
- 4 – Tom firkant
- 5 – Kryss
- 6 – Pluss
- 7 – Tynn
- 8 – middels punkt, fast
- 9 – middels punkt, tomt
- 10 – større punkt, fast
- 11 – større punkt, tomt
- 12 – største punkt, fast
- 13 – største punkt, tomt

PlotXY 100,100,1

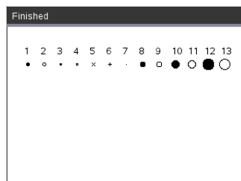


For n,1,13

DrawText 1+22*n,40,n

PlotXY 5+22*n,50,n

EndFor



SetColor
 Katalog > 
CXII
SetColor

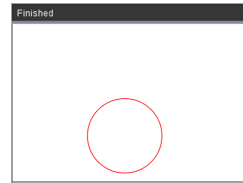
Red-value, Green-value, Blue-value

Gyldige verdier for rød, grønn og blå er mellom 0 og 255

Angir fargen for påfølgende Tegnekommandoer

SetColor 255,0,0

DrawCircle 150,150,100

**SetPen**
 Katalog > 
CXII
SetPen

tykkelse, stil

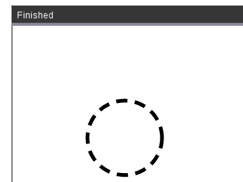
tykkelse: 1 <= tykkelse <= 3 | 1 er tynneste, 3 er tykkeste

stil: 1 = Jevn, 2 = Stiplet, 3 = Punkt

Angir pennestilen for påfølgende Tegnekommandoer

SetPen 3,3

DrawCircle 150,150,50

**SetWindow**
 Katalog > 
CXII
SetWindow

xMin, xMax, yMin, yMax

Etablerer et logisk vindu som settes inn på grafikktegneområdet. Alle parametere er nødvendige.

Hvis deler av et tegnet objekt er utenfor vinduet, blir utdataene klippet (ikke vist) og ingen feilmelding vises.

SetWindow 0,160,0,120

vil sette utskriftsvinduet til 0,0 i nederste venstre hjørne med en bredde på 160 og en høyde på 120

DrawLine 0,0,100,100

SetWindow 0,160,0,120

SetPen 3,3

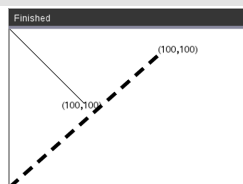
DrawLine 0,0,100,100

Hvis x_{\min} er større enn eller lik x_{\max} eller y_{\min} er større enn eller lik y_{\max} , vises en feilmelding.

Eventuelle objekter som er tegnet før en SetWindow-kommando, blir ikke tegnet på nytt i den nye konfigurasjonen.

For å tilbakestille vindusparametrene til standardinnstillingene, bruk:

SetWindow 0,0,0,0



UseBuffer

Tegn til grafikkbuffer utenfor skjermen i stedet for til skjerm (for å øke ytelsen)

Denne kommandoen brukes sammen med PaintBuffer for å øke visningshastigheten på skjermen når programmet genererer flere grafiske objekter.

Med UseBuffer vises all grafikk først etter at neste PaintBuffer-kommando er utført.

UseBuffer trenger bare å bli anropt én gang i programmet, dvs. hver bruk av PaintBuffer trenger ikke en tilsvarende UseBuffer

Se også: [PaintBuffer](#)

UseBuffer

```
For n,1,10
```

```
x:=randInt(0,300)
```

```
y:=randInt(0,200)
```

```
radius:=randInt(10,50)
```

```
Wait 0,5
```

```
DrawCircle x,y,radius
```

```
EndFor
```

```
PaintBuffer
```

Dette programmet viser alle de 10 sirklene samtidig.



Hvis kommandoen «UseBuffer» fjernes, vises hver sirkel slik den er tegnet.

Tomme (åpne) elementer

Når du analyserer reelle data, kan det hende at du ikke alltid har et komplett datasett. TI-Nspire™ tillater tomme eller åpne dataelementer, slik at du kan fortsette med data som nesten er komplette istedenfor å måtte starte på nytt eller forkaste oppgaver som ikke er fullført.

Under “*Plotte graf fra regnearkdata*” i kapitlet Lister og regneark finner du et eksempel på data som involverer tomme elementer.

Med funksjonen **delVoid()** kan du fjerne tomme elementer fra en liste. Med funksjonen **isVoid()** kan du teste for et tomt element. For detaljer, se **delVoid()**, side 39, og **isVoid()**, side 76.

Merk: For å legge inn et tomt element manuelt i et matematisk uttrykk, skriv “_” eller nøkkelordet **void**. Nøkkelordet **void** konverteres automatisk til et “_”-symbol når uttrykket blir behandlet. For å skrive inn “_” på grafregneren, trykk på  .

Beregninger som involverer åpne elementer

De fleste beregninger som involverer et åpent (tomt) innlegg, vil produsere et åpent (tomt) resultat. Se spesialtilfeller nedenfor.

$_$	$_$
$\gcd(100, _)$	$_$
$3 + _$	$_$
$\{5, _, 10\} - \{3, 6, 9\}$	$\{2, _, 1\}$

Listeutsagn som inneholder åpne (tomme) elementer

Følgende funksjoner og kommandoer ignorerer (hopper over) åpne (tomme) elementer som blir funnet i listeutsagn.

count, **countif**, **cumulativeSum**, **freqTable**→**list**, **frequency**, **max**, **mean**, **median**, **product**, **stDevPop**, **stDevSamp**, **sum**, **sumlf**, **varPop**, og **varSamp**, samt regresjonsberegninger, **OneVar**, **TwoVar** og **FiveNumSummary** statistikk, konfidensintervaller og statistikktester

$\text{sum}(\{2, _, 3, 5, 6, 6\})$	16.6
$\text{median}(\{1, 2, _, _, 3\})$	2
$\text{cumulativeSum}(\{1, 2, _, 4, 5\})$	$\{1, 3, _, 7, 12\}$
$\text{cumulativeSum}\left(\begin{pmatrix} 1 & 2 \\ 3 & _ \\ 5 & 6 \end{pmatrix}\right)$	$\begin{pmatrix} 1 & 2 \\ 4 & _ \\ 9 & 8 \end{pmatrix}$

SortA og **SortD** flytter alle åpne (tomme) elementer innenfor det første utsagnet til bunnen.

$\{5, 4, 3, _, 1\} \rightarrow \text{list1}$	$\{5, 4, 3, _, 1\}$
$\{5, 4, 3, 2, 1\} \rightarrow \text{list2}$	$\{5, 4, 3, 2, 1\}$
$\text{SortA list1, list2}$	<i>Done</i>
list1	$\{1, 3, 4, 5, _ \}$
list2	$\{1, 3, 4, 5, 2\}$

Listeutsagn som inneholder åpne (tomme) elementer

I regresjoner introduserer en åpning i en X- eller Y-liste en åpning for det tilsvarende elementet i en rest.

$\{1,2,3,_,5\} \rightarrow list1$	$\{1,2,3,_,5\}$
$\{1,2,3,4,5\} \rightarrow list2$	$\{1,2,3,4,5\}$
SortD list1,list2	Done
list1	$\{5,3,2,1,_\}$
list2	$\{5,3,2,1,4\}$
<hr/>	
$I1:=\{1,2,3,4,5\}; I2:=\{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
LinRegMx I1,I2	Done
stat.Resid	$\{0.434286,_, -0.862857, -0.011429, 0.44\}$
stat.XReg	$\{1,_,3,4,5\}$
stat.YReg	$\{2,_,3,5,6,6\}$
stat.FreqReg	$\{1,_,1,1,1,1\}$

En utelatt kategori i en regresjon introduserer en åpning (tomt element) for det tilsvarende elementet i en rest.

$I1:=\{1,3,4,5\}; I2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
$cat:=\{"M","M","F","F"\}; incl:=\{"F"\}$	$\{"F"\}$
LinRegMx I1,I2,1,cat,incl	Done
stat.Resid	$\{_,_,0,0,0\}$
stat.XReg	$\{_,_,4,5\}$
stat.YReg	$\{_,_,5,6,6\}$
stat.FreqReg	$\{_,_,1,1,1\}$

En frekvens på 0 i en regresjon introduserer en åpning (tomt element) for det tilsvarende elementet i en rest.

$I1:=\{1,3,4,5\}; I2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx I1,I2,\{1,0,1,1\}	Done
stat.Resid	$\{0.069231,_, -0.276923, 0.207692\}$
stat.XReg	$\{1,_,4,5\}$
stat.YReg	$\{2,_,5,6,6\}$
stat.FreqReg	$\{1,_,1,1,1\}$

Snarveier/hurtigtaster for å legge inn matematiske uttrykk

Hurtigtaster lar deg legge inn matematiske uttrykk ved å skrive i stedet for å bruke katalogen eller symbolpaletten. Eksempel: Når du skal legge inn uttrykket $\sqrt{6}$, kan du skrive `sqrt (6)` på kommandolinjen. Når du trykker på `enter`, endres uttrykket `sqrt (6)` til $\sqrt{6}$. Noen hurtigtaster kan brukes både fra kalkulatoren og fra tastaturet på datamaskinen. Andre er først og fremst nyttige fra tastaturet på datamaskinen.

Fra kalkulatoren eller datamaskintastaturet

Hvis du skal legge inn dette:	Skriv dette:
π	<code>pi</code>
θ	<code>theta</code>
∞	<code>infinity</code>
\leq	<code><=</code>
\geq	<code>>=</code>
\neq	<code>/=</code>
\Rightarrow (logisk implikasjon)	<code>=></code>
\Leftrightarrow (logisk dobbel implikasjon, XNOR)	<code><=></code>
\rightarrow (lagre-operator)	<code>=:</code>
$ $ (absoluttverdi)	<code>abs (...)</code>
$\sqrt{()}$	<code>sqrt (...)</code>
$\Sigma()$ (Sum-sjablon)	<code>sumSeq (...)</code>
$\Pi()$ (Produkt-sjablon)	<code>prodSeq (...)</code>
$\sin^{-1}()$, $\cos^{-1}()$, ...	<code>arcsin (...)</code> , <code>arccos (...)</code> , ...
$\Delta\text{List}()$	<code>deltaList (...)</code>

Fra tastaturet på datamaskinen

Hvis du skal legge inn dette:	Skriv dette:
i (imaginær konstant)	<code>@i</code>
e (naturlig log-grunntall e)	<code>@e</code>
E (vitenskapelig notasjon)	<code>@E</code>
\top (transponert)	<code>@t</code>

Hvis du skal legge inn dette:	Skriv dette:
$^{\circ}$ (radianer)	@r
$^{\circ}$ (grader)	@d
$^{\circ}$ (gadianer)	@g
\angle (vinkel)	@<
► (konvertering)	@>
►Decimal, ►approxFraction (), osv.	@>Decimal, @>approxFraction(), osv.

EOS™ (Ligningsoperativsystem)-hierarkiet

Dette avsnitt beskriver Equation Operating System (ligningsoperativsystem) (EOS™) som brukes av TI-Nspire™ -teknologien for undervisning i matematikk og realfag. Tall, variabler og funksjoner legges inn i en enkel, ukomplisert sekvens. EOS™ -programvaren behandler uttrykk og ligninger ved hjelp av parentetisk gruppering og i samsvar med de prioriteringene som beskrives over.

Rekkefølge av beregning

Nivå	Operator
1	Parentes (), hakeparentes [], buet parentes { }
2	Omregning (#)
3	Oppkalling av funksjon
4	Postoperatorer: grader-minutter-sekunder ([°] , ' , ") , fakultet (!) , prosent (%) , radianer (^r) , senket skrift ([]) , transponert (^T)
5	Eksponensiering, potens-operator (^)
6	Negasjon (L)
7	Sett sammen streng (&)
8	Multiplikasjon (•) , divisjon (/)
9	Addisjon (+) , subtraksjon (-)
10	Likhetsrelasjoner: lik (=) , ulik (≠ eller /≠) , mindre enn (<) , mindre enn eller lik (≤ eller <=) , større enn (>) , større enn eller lik (≥ eller >=)
11	Logisk not
12	Logisk and
13	Logiske or
14	enten...eller, verken ...eller, ikke både ...og
15	Logisk implikasjon (⇒)
16	Logisk dobbel implikasjon, XNOR (↔)
17	Begrensningsoperator (" ")
18	Lagre (→)

Parenteser, hakeparenteser, buede parenteser

Først behandles alle beregninger som står i parentes, hakeparentes eller buet parentes. I uttrykket $4(1+2)$ behandler EOS™ -programvaren for eksempel først den delen av uttrykket som står i parenteser, $1+2$, og multipliserer deretter resultatet, 3, med 4.

Antallet åpne- og lukkeparenteser, åpne- og lukke-hakeparenteser og buede åpne- og lukkeparenteser må være det samme innenfor ett uttrykk eller én ligning. Hvis ikke,

vises en feilmelding, som angir det manglende elementet. For eksempel vil $(1+2)/(3+4)$ vise feilmeldingen "Mangler)."

Merk: Siden TI-Nspire™ -programvaren gjør at du kan definere dine egne funksjoner, blir et variabelnavn fulgt av et uttrykk i parentes betraktet som en "oppkalling av funksjon" istedenfor halvveis skjult multiplikasjon. For eksempel i $a(b+c)$ blir funksjonen a beregnet for verdien $b+c$ (av den variable). For å multiplisere uttrykket $b+c$ med variabelen a, må du bruke eksplisitt multiplikasjon: $a*(b+c)$.

Omregning

Omregnings-operatoren (#) omregner en streng til et variabel- eller funksjonsnavn. For eksempel oppretter #("x"&"y"&"z") variabelnavnet xyz. Omregning lar deg også opprette og modifisere en variabel mens du er inne i et program. Hvis for eksempel $10 \rightarrow r$ og " r " $\rightarrow s1$, så er $\#s1=10$.

Postoperatorer

En postoperator er en operator som kommer direkte etter et argument, som f.eks. 5!, 25%, eller $60^\circ 15' 45''$. Et argument som er fulgt av en postoperator blir behandlet ved fjerde prioritetsnivå. I uttrykket $4^3!$ blir for eksempel 3! behandlet først. Resultatet, 6, blir så eksponenten av 4 for å oppnå 4096.

Eksponensiering

Eksponensiering (^) og element-for-element-eksponensiering (.^) blir behandlet fra høyre til venstre. Uttrykket 2^3^2 blir for eksempel behandlet som det samme som 2^4 (3^2) for å produsere 512. Dette er forskjellig fra $(2^3)^2$, som er 64.

Negasjon

For å legge inn et negativt tall, trykk på $(-)$ fulgt av tallet. Postoperasjoner og eksponensiering utføres før negasjon. Resultatet av $-x^2$ er for eksempel et negativt tall, og $-9^2 = -81$. Bruk parenteser for å opphøye et negativt tall i annen potens, som f.eks. $(-9)^2$ for å produsere 81.

Begrensning ("|")

Argumentet som følger etter ("|")-operator gir et sett av begrensninger som påvirker hvordan argumentet som står foran operatoren blir behandlet.

TI-Nspire CX II – TI-Basic programmeringsfunksjoner

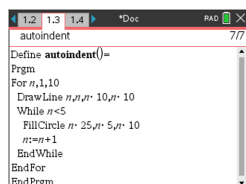
Auto-innrykk i Programmeringseditor

TI-Nspire™ progeditor gir automatisk innrykking av utsagn i en blokkkommando.

Blokkkommandoer er If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

Programeditor vil automatisk legge til mellomrom foran programkommandoer i en blokkkommando. Den avsluttende kommandoen til blokken blir innrettet likt med åpningskommandoer.

Eksemplet nedenfor viser automatisk innrykk i nestede blokkkommandoer.



```
autoindent 77
Define autoindent()=
Prgm
For n,1,10
DrawLine n,n,n-10,n-10
While n<5
FillCircle n-25,n-5,n-10
n:=n+1
EndWhile
EndFor
EndPrgm
```

Kodefragmenter som kopieres og limes inn, beholder originalinnrykket.

Åpning av et program som er opprettet i en tidligere versjon av programvaren, beholder originalinnrykket.

Forbedrede feilmeldinger for TI-Basic

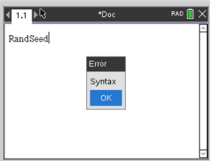
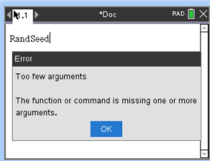
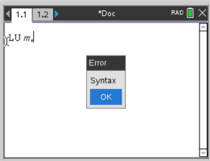
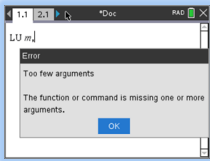
Feil

Feiltilstand	Ny melding
Feil i betinget utsagn (If/While)	Et betinget utsagn ble ikke løst til SANN eller USANN MERK: Etter endringen for å plassere markøren på linjen med feilen, trenger vi ikke lenger å angi om feilen er i et «If»-utsagn eller i et «While»-utsagn.
Mangler EndIf	Forventet EndIf , men fant en annen end-setning
Mangler EndFor	Forventet EndFor , men fant en annen end-setning
Mangler EndWhile	Forventet EndWhile , men fant en annen end-setning
Mangler EndLoop	Forventet EndLoop , men fant en annen end-setning

Feiltilstand	Ny melding
Mangler EndTry	Forventet EndTry , men fant en annen end-setning
« Then » utelatt etter If <condition>	Mangler If..Then
« Then » utelatt etter Elseif <condition>	Then mangler i blokken: Elseif .
Når « Then », « Else » og « Elseif » ble støtt på utenfor kontrollblokkene	Else er ugyldig utenfor blokkene: If..Then..EndIf eller Try..EndTry
« Elseif » vises utenfor « If..Then..EndIf »-blokken	Elseif er ugyldig utenfor blokk: If..Then..EndIf
« Then » vises utenfor « If....EndIf »-blokken	Then er ugyldig utenfor blokkene: If..EndIf

Syntaksfeil

Hvis kommandoer som forventer ett eller flere argumenter blir oppkalt med en ufullstendig liste over argumenter, vil en «**For få argumenter-feil**» vises istedenfor for «**syntaks**»-feil

Gjeldende atferd	Ny CX II-atferd
 <p>A screenshot of a TI-84 Plus CE calculator window titled '*Doc'. The cursor is on the command 'RasdSeed'. A small error dialog box is displayed with the text 'Error', 'Syntax', and an 'OK' button.</p>	 <p>A screenshot of a TI-84 Plus CE calculator window titled '*Doc'. The cursor is on the command 'RasdSeed'. A larger error dialog box is displayed with the text 'Error', 'Too few arguments', 'The function or command is missing one or more arguments.', and an 'OK' button.</p>
 <p>A screenshot of a TI-84 Plus CE calculator window titled '*Doc'. The cursor is on the command 'LU m'. A small error dialog box is displayed with the text 'Error', 'Syntax', and an 'OK' button.</p>	 <p>A screenshot of a TI-84 Plus CE calculator window titled '*Doc'. The cursor is on the command 'LU m'. A larger error dialog box is displayed with the text 'Error', 'Too few arguments', 'The function or command is missing one or more arguments.', and an 'OK' button.</p>

Konstanter og verdier

Den følgende tabellen inneholder konstanter og deres verdier, som er tilgjengelige når du utfører enhetsregninger. De kan skrives inn manuelt eller velges fra listen

Konstanter i Verktøy > Enhetsomregninger (Håndholdt enhet: trykk på  3).

Konstant	Navn	Verdi
_c	Lysets hastighet	299792458 _m/_s
_Cc	Coulomb-konstant	8987551792,261 _m/_F
_Fc	Faraday-konstant	96485,33212 _coul/_mol
_g	Gravitasjonens akselerasjon	9,80665 _m/_s ²
_Gc	Gravitasjonskonstant	6,6743E-11 _m ³ /_kg/_s ²
_h	Plancks konstant	6,62607015E-34 _J _s
_k	Boltzmanns konstant	1,380649E-23 _J/_°K
_μ0	Permeabilitet i vakuum	1,25663706212E-6 _N/_A ²
_μb	Bohr-magneton	9,274009994E-24 _J _m ² /_Wb
_Me	Elektronets hvilemasse	9,1093837015E-31 _kg
_Mμ	Myon-masse	1,883531627E-28 _kg
_Mn	Nøytronets hvilemasse	1,67492749804E-27 _kg
_Mp	Protonets hvilemasse	1,67262192369E-27 _kg
_Na	Avogadros tall	6,02214076E23 /_mol
_q	Elektronladning	1,602176634E-19 _coul
_Rb	Bohr-radius	5,29177210903E-11 _m
_Rc	Molar gasskonstant	8,314462618 _J/_mol/_°K
_Rdb	Rydbergs konstant	10973731,568160/_m
_Re	Elektron-radius	2,8179403262E-15 _m
_u	Atommasse	1,6605390666E-27 _kg
_Vm	Molarvolum	2,241396954E-2 _m ³ /_mol
_ε0	Permittivitet i vakuum	8,8541878128E-12 _F/_m
_σ	Stefan-Boltzmann-konstant	5,670367E-8 _W/_m ² /_°K ⁴
_φ0	Magnetisk flukskvantum	2,067833831E-15 _Wb

Feilkoder og feilmeldinger

Hvis det oppstår en feil, er koden knyttet til variabel *feilKode*. Egendefinerte programmer og funksjoner kan undersøke *feilKode* for å bestemme årsaken til feilen. For et eksempel på bruk av *feilKode*, se eksempel 2 under kommandoen **Prøv**, side 158.

Merk: Noen feilforhold gjelder kun for TI-Nspire™ CAS-produktene, og noen gjelder kun for TI-Nspire™-produktene.

Feilkode	Beskrivelse
10	En funksjon returnerte ingen verdi
20	En test ga ikke resultatet SANN eller USANN. Vanligvis kan udefinerte variabler ikke sammenliknes. Testen $If\ a < b$ vil for eksempel forårsake enten at a eller at b ikke er definert, dersom utsagnet If blir utført.
30	Argumentet kan ikke være et mappenavn.
40	Argumentfeil
50	Uoverensstemmelse i argument To eller flere argumenter må være av samme type.
60	Argumentet må være et Boolsk uttrykk eller et heltall
70	Argumentet må være et desimaltall
90	Argumentet må være en liste
100	Argumentet må være en matrise
130	Argumentet må være en streng
140	Argumentet må være et variabelnavn. Pass på at navnet: <ul style="list-style-type: none">• ikke begynner med et tall• ikke inneholder mellomrom eller spesialtegn• ikke bruker senket strek eller punktum på ugyldig måte• ikke overgår tillatt lengde Les mer om dette i Kalkulator-avsnittet dokumentasjonen.
160	Argumentet må være et uttrykk
165	For lite strøm i batteriene til å sende/motta Legg i nye batterier før sending eller mottak.
170	Grense Den nedre grensen må være mindre enn den øvre grensen for å definere søkeintervallet.

Feilkode	Beskrivelse
180	<p>Avbryt</p> <p>Det ble trykket på tasten <code>esc</code> eller <code>on</code> under en lang beregning eller mens et program ble utført.</p>
190	<p>Sirkulær definisjon</p> <p>Denne meldingen komme til syne for å unngå at du slipper opp for minne under uendelig erstating av variable verdier. For eksempel vil $a+1 \rightarrow a$, der a er en udefinert variabel, forårsake denne feilen.</p>
200	<p>Ugyldig begrensingsuttrykk</p> <p>For eksempel vil $\text{løs}(3x^2-4=0,x) \mid x < 0 \text{ eller } x > 5$ produsere denne feilmeldingen, fordi begrensningen er skilt med "eller" istedenfor "og".</p>
210	<p>Ugyldig datatype</p> <p>Et argument er av feil datatype.</p>
220	Avhengig grense
230	<p>Dimensjon</p> <p>En liste eller matriseindeks er ikke gyldig. Hvis for eksempel listen $\{1,2,3,4\}$ er lagret i $L1$, så er $L1[5]$ en dimensjonsfeil, fordi $L1$ kun inneholder fire elementer.</p>
235	Dimensjonsfeil. Ikke nok elementer i listene.
240	<p>Dimensjonsfeil</p> <p>To eller flere argumenter må være av samme dimensjon. For eksempel er $[1,2]+[1,2,3]$ en dimensjonsfeil, fordi matrisene inneholder ulikt antall elementer.</p>
250	Divisjon med null
260	<p>Grunnmengdefeil</p> <p>Et argument må være i en spesifisert grunnmengde. For eksempel er $\text{tilf}(0)$ ikke gyldig.</p>
270	Duplikatnavn på variabel
280	Else og Elseif ugyldig utenfor If...EndIf-blokk
290	EndTry uten tilhørende Else-uttrykk
295	For mange iterasjoner
300	Forventet 2- eller 3-elements liste eller matrise
310	Det første argumentet av $n\text{Solve}$ må være en ligning i én variabel. Det kan ikke inneholde noen annen variabel enn den variabelen som vi er interessert i.
320	<p>Det første argumentet til løs eller $k\text{Løs}$ må være en ligning eller ulikhet</p> <p>For eksempel er $\text{løs}(3x-4,x)$ ugyldig fordi det første argumentet ikke er en ligning.</p>

Feilkode	Beskrivelse
345	Inkonsistente enheter
350	Indeks utenfor gyldig område
360	Indireksjonsstrengen er ikke et gyldig variabelnavn
380	Udefinert Svar Enten opprettet ikke den forrige beregningen noe Svar, eller det ble ikke lagt inn noe forrige beregning.
390	Ugyldig tildeling
400	Ugyldig tildelingsverdi
410	Ugyldig kommando
430	Ugyldig for de gjeldende modusinnstillingene
435	Ugyldig gjetning (startverdi)
440	Ugyldig "skjult" multiplikasjon For eksempel er $x(x+1)$ ugyldig; derimot er $x*(x+1)$ korrekt syntaks. Dette skal forhindre forvirring mellom halvveis skjult multiplikasjon og oppkalling av funksjon.
450	Ugyldig i en funksjon eller gjeldende uttrykk Det er kun visse kommandoer som er gyldige i en egendefinert funksjon.
490	Ugyldig i Try..EndTry-blokk
510	Ugyldig liste eller matrise
550	Ugyldig utenfor funksjon eller program Et antall kommandoer er ikke gyldige utenfor en funksjon eller et program. For eksempel kan ikke Lokal brukes hvis den ikke er inne i en funksjon eller et program.
560	Ugyldig utenfor Loop..EndLoop-, For..EndFor- eller While..EndWhile-blokk For eksempel er Avslutt-kommandoen kun gyldig inne i disse loop-blokkene.
565	Ugyldig utenfor program
570	Ugyldig banenavn For eksempel er \var ugyldig.
575	Ugyldig polar kompleks verdi
580	Ugyldig programreferanse Det kan ikke refereres til programmer inne i funksjoner eller uttrykk, som f.eks. $1+p(x)$, der p er et program.

Feilkode	Beskrivelse
600	Ugyldig tabell
605	Ugyldig bruk av enheter
610	Ugyldig variabelnavn i Lokalt utsagn
620	Ugyldig variabel- eller funksjonsnavn
630	Ugyldig variabelreferanse
640	Ugyldig vektorsyntaks
650	Kommunikasjons-forbindelse En kommunikasjon mellom to enheter er ikke fullført. Kontroller at forbindelseskabelen er koplet godt til i begge ender.
665	Matrisen kan ikke diagonaliseres
670	Lite minne 1. Slett noen data i dette dokumentet 2. Lagre og lukk dette dokumentet Dersom 1 og 2 mislykkes, ta ut batteriene og sett dem inn igjen
672	Ressursbegrensning
673	Ressursbegrensning
680	Manglende (
690	Manglende)
700	Manglende “
710	Manglende]
720	Manglende }
730	Manglende start eller slutt på bloksyntaks
740	Manglende Then i If..EndIf-blokken
750	Navnet er ikke en funksjon eller et program
765	Ingen funksjoner er valgt
780	Fant ingen løsning
800	Ikke-reelt resultat Hvis for eksempel programvaren er i Reell innstilling, er $\sqrt{-1}$ ugyldig. For å tillate komplekse resultater, endre “Reell eller Kompleks” modusinnstilling til REKTANGULÆR eller POLAR.

Feilkode	Beskrivelse
830	Overflyt
850	Fant ikke programmet Det ble ikke funnet en programreferanse i et annet program i oppgitt bane under utføring.
855	Rand-funksjonstyper ikke tillatt i grafer
860	Rekursjonen for dyp
870	Reservert navn eller systemvariabel
900	Argumentfeil Median-median-modell kunne ikke brukes på datasettet.
910	Syntaksfeil
920	Fant ikke teksten
930	For få argumenter Funksjonen eller kommandoen mangler et eller flere argumenter.
940	For mange argumenter Uttrykket eller ligningen inneholder for mange argumenter og kan ikke behandles.
950	For mange indekser
955	For mange udefinerte variabler
960	Variabelen er ikke definert Variabelen er ikke tildelt noen verdi. Bruk en av følgende kommandoer: <ul style="list-style-type: none"> • sto → • := • Define for å tildele variablene verdi.
965	Ulisensiert OS
970	Variabel er i bruk, så referanser eller endringer er ikke tillatt
980	Variabel er beskyttet
990	Ugyldig variabelnavn Pass på at navnet ikke overgår tillatt lengde
1000	Grunnmengde for vindusvariabel
1010	Zoom

Feilkode	Beskrivelse
1020	Intern feil
1030	Overtredelse av beskyttet minne
1040	Ustøttet funksjon. Denne funksjonen krever Computer Algebra System. Prøv TI-Nspire CAS.
1045	Ustøttet operator. Denne operatoren krever Computer Algebra System. Prøv TI-Nspire CAS.
1050	Ustøttet egenskap. Denne operatoren krever Computer Algebra System. Prøv TI-Nspire CAS.
1060	Innlagt argument må være numerisk. Bare innlegg som inneholder numeriske verdier er tillatt.
1070	Trig-funksjonsargument for stort for nøyaktig reduksjon
1080	Ustøttet bruk av Svar. Denne applikasjonen støtter ikke Svar.
1090	Funksjonen er ikke definert. Bruk en av følgende kommandoer: <ul style="list-style-type: none"> • Define • := • sto → for å definere en funksjon.
1100	Ikke-reell beregning Hvis for eksempel programvaren er i Reell innstilling, er $\sqrt{-1}$ ugyldig. For å tillate komplekse resultater, endre "Reell eller Kompleks" modusinnstilling til REKTANGULÆR eller POLAR.
1110	Ugyldige grenser
1120	Tegn ikke endret
1130	Argumentet kan ikke være en liste eller matrise
1140	Argumentfeil Det første argumentet må være et polynomisk uttrykk i det andre argumentet. Dersom det andre argumentet utelates, prøver programvaren å velge en grunninnstilling.
1150	Argumentfeil De første to argumentene må være polynomiske uttrykk i det tredje argumentet. Dersom det tredje argumentet utelates, prøver programvaren å velge en grunninnstilling.
1160	Ugyldig banenavn for bibliotek Et banenavn må være av formen xxx\yyy, der: <ul style="list-style-type: none"> • Delen xxx kan bestå av mellom 1 og 16 tegn.

Feilkode	Beskrivelse
	<ul style="list-style-type: none"> Delen <code>yyy</code> kan ha 1 til 15 tegn. <p>Les mer om dette i Bibliotek-avsnittet dokumentasjonen.</p>
1170	<p>Ugyldig bruk av banenavn for bibliotek</p> <ul style="list-style-type: none"> En verdi kan ikke tildeles et banenavn som bruker Define, <code>:=</code> eller <code>sto</code> →. Et banenavn kan ikke erklæres som en lokal variabel eller brukes som parameter i en funksjonsdefinisjon eller programdefinisjon.
1180	<p>Ugyldig variabelnavn på bibliotek</p> <p>Pass på at navnet:</p> <ul style="list-style-type: none"> Ikke inneholder punktum Ikke begynner med senket strek Ikke består av mer enn 15 tegn <p>Les mer om dette i Bibliotek-avsnittet dokumentasjonen.</p>
1190	<p>Bibliotek-dokumentet ble ikke funnet:</p> <ul style="list-style-type: none"> Kontroller om biblioteket er i mappen <code>MittBibl</code>. Oppdater biblioteker. <p>Les mer om dette i Bibliotek-avsnittet dokumentasjonen.</p>
1200	<p>Bibliotek-variabler ble ikke funnet:</p> <ul style="list-style-type: none"> Kontroller om bibliotek-variablene eksisterer i den første oppgaven i biblioteket. Forsikre deg om at bibliotek-variabelen er blitt definert som <code>BiblOff</code> eller <code>BiblPriv</code>. Oppdater biblioteker. <p>Les mer om dette i Bibliotek-avsnittet dokumentasjonen.</p>
1210	<p>Ugyldig navn på snarvei til bibliotek.</p> <p>Pass på at navnet:</p> <ul style="list-style-type: none"> Ikke inneholder punktum Ikke begynner med senket strek Ikke består av mer enn 16 tegn Ikke er et reservert navn <p>Les mer om dette i Bibliotek-avsnittet i dokumentasjonen.</p>
1220	<p>Grunnmengdefeil:</p> <p>Funksjonene <code>tangentLinje</code> og <code>normalLinje</code> støtter kun funksjoner med reelle verdier.</p>
1230	<p>Grunnmengdefeil.</p>

Feilkode	Beskrivelse
	Trigonometriske omregningsoperatorer støttes ikke i Grader- eller Gradianvinkelmodus.
1250	Argumentfeil Bruk et system av lineære ligninger. Eksempel på et system av to lineære ligninger med variablene x og y: $3x+7y=5$ $2y-5x=-1$
1260	Argumentfeil: Det første argumentet til nfMin eller nfMax må være et uttrykk i én variabel. Det kan ikke inneholde noen annen variabel enn den variabelen som vi er interessert i.
1270	Argumentfeil Den deriverte må være av orden 1 eller 2.
1280	Argumentfeil Bruk et polynom på utvidet (ekspandert) form i én variabel.
1290	Argumentfeil Bruk et polynom i én variabel.
1300	Argumentfeil Koeffisientene i polynomet må være numeriske verdier.
1310	Argumentfeil: En funksjon kan ikke behandles for ett eller flere av dens argumenter.
1380	Argumentfeil: Nestede oppringinger til område() funksjon er ikke tillatt.

Advarselskoder og -meldinger

Du kan bruke funksjonen **warnCodes()** for å lagre advarselskodene som ble generert da et uttrykk ble behandlet. Denne tabellen opplyster hver numeriske varselkode og dens assosierte melding. Se **warnCodes()** for et eksempel på lagring av advarselskoder, side 166.

VARSELKODE	Melding
10000	Kommandoen kan gi falske løsninger. Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10001	Derivasjon av en ligning kan gi en ugyldig ligning.
10002	Tvilsom løsning Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10003	Tvilsom nøyaktighet Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10004	Kommandoen kan utelate løsninger. Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10005	KLøs kan spesifisere flere nullpunkter.
10006	Løs kan spesifisere flere nullpunkter. Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10007	Flere løsninger kan eksistere. Prøv å angi passende øvre og nedre grenser, og/eller en gjetning. Eksempler ved bruk av solve(): <ul style="list-style-type: none">• solve(Ligning, Var=Forslag) nedGrens<Var<øvreGrens• solve(Ligning, Var) nedGrens<Var<øvreGrens• solve(Ligning,Var=Forslag) Når det er aktuelt, prøv å bruke grafiske metoder for å verifisere resultatene.
10008	Grunnmengden til resultatet kan være mindre enn grunnmengden til innlegget (inndata).
10009	Grunnmengden til resultatet kan være større enn grunnmengden til innlegget (inndata).
10012	Ikke-reell beregning
10013	∞^0 eller undef^0 erstattet av 1
10014	undef^0 erstattet av 1

vARSELKODE	Melding
10015	1^∞ eller 1^{undef} erstattet av 1
10016	1^{undef} erstattet av 1
10017	Overløp erstattet av ∞ eller $-\infty$
10018	Kommando krever og returnerer 64-biters verdi.
10019	Ressursbegrensning, forenkling kanskje ufullstendig.
10020	Trig-funksjonsargument for stort for nøyaktig reduksjon.
10021	Inndataene inneholder en udefinert parameter. Resultatet kanskje ikke gyldig for alle mulige parameterverdier.
10022	Å spesifisere riktig nedre og øvre grense kan gi en løsning.
10023	Skalar har blitt multiplisert med identitetsmatrisen.
10024	Resultat oppnådd med tilnærmet aritmetikk.
10025	Ekvivalens kan ikke verifiseres i EXACT-modus.
10026	Begrensningen kan bli ignorert. Spesifiser begrensning i skjemaet «Variable MathTestSymbol Constant» eller et konjunkt (en kombinasjon) av disse formene, f.eks. 'x<3 and x>-12'

Generell informasjon

Hjelp på nettet (online)

education.ti.com/eguide

Velg ditt land for mer produktinformasjon.

Kontakt TIs brukerstøtte

education.ti.com/ti-cares

Velg ditt land for tekniske og andre støtteressurser.

Service og garantiinformasjoner

education.ti.com/warranty

Velg landet ditt for informasjon om lengden og vilkårene for garantien eller om produkttjenester.

Begrenset garanti. Denne garantien påvirker ikke dine lovmessige rettigheter.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Stikkordregister

		, begrensingsoperator	191
		+	
' , fremstilling minutter	189	+ , addere	175
		/	
- , subtrahere[*]	175	/ , dividere[*]	176
		=	
! , faktultet	183	≠ , ulik[*]	180
		= , er lik	179
" , fremstilling sekunder	189	>	
		> , større enn	181
#		∏	
# , Indir.ref	187	∏ , produkt, sjablon for	5
# , omregnings-operator	214	∏ , produkt[*]	184
%		∑	
% , prosent	179	∑ () , sum[*]	185
		∑Int ()	185
		∑Prn ()	186
&		√	
& , legg til	183	√ , kvadratrott[*]	184
*		≤	
* , multiplisere	176	≤ , mindre enn eller lik	181
		≥	
.- , prikk subtraksjon	178	≥ , større enn eller lik med	182
.* , prikk multiplikasjon	178		
./ , prikk divisjon	178	►	
.^ , prikk potens	178	►Cylind, vise som sylindrisk vektor [Sylind]	35
.+ , prikk addisjon	177	►Polar, vise som polar vektor[Polar]	112
		► , omregner til gradian vinkel[Grad]	69
:		►approxFraction ()	13
:= , tildele	192	►DD, vises som desimalvinkel[DD] ..	36
^			
^ , potens	176		

►Desimal, vise resultat som desimal [Decimal]	36
►DMS, vise som grader/minutter/sekunder [DMS (GMS)]	43
►Grunntall10, vise som desimalt heltall[Grunntall10]	18
►Grunntall16, vise som heksadesimal[Grunntall16]	18
►Grunntall2, vise som binær [Grunntall2]	16
►Rad, omregne til radian vinkel	120
►Rect, vise som rektangulær vektor	123
►Sphere, vise som sfærisk vektor [Sfære (kule)]	146

→

→, lagre	192
----------------	-----

⇒

⇒, logisk implikasjon[*]	182, 211
--------------------------------	----------

↔

↔, logisk dobbel implikasjon[*]	183
---------------------------------------	-----

©

©, kommentar	193
--------------------	-----

°

°, grader fremstilling[*]	189
°, grader/minutter/sekunder[*]	189

0

Ob, binær indikator	193
Oh, heksadesimal indikator	193

1

10^(), tier-potens	190
---------------------------	-----

2

2-delers stykkevis funksjon sjablon for	2
2-utvalg F test	57

A

abs(), absoluttverdi	7
Absoluttverdi sjablon for	3-4
addere, +	175
amortiseringstabell, amortTbl()	7, 15
amortTbl(), amortiseringstabell	7, 15
and, Boolsk operator	8
andrederivert sjablon for	6
angle(), vinkel	9
ANOVA, enveis varians-analyse	9
ANOVA2-veis, toveis varians-analyse	10
Ans, siste svar	12
antall betingede elementer i en liste , tellIf()	30
antall elementer i en liste, antall() .	30
antall(), antall elementer i en liste .	30
åpne (tomme) elementer	209
åpne elementer, fjern	39
approx(), tilnærmet	12
approxRational()	13
arccos()	13
arccosh()	13
arccot()	13
arcoth()	13
arccsc()	13
arccsch()	14
arcsec()	14
arcsech()	14
arcsin()	14
arcsinh()	14
arctan()	14
arctanh()	14
argumenter i TVM-funksjoner	162
augment(), utvid/sett sammen	14
avgRC(), gjennomsnittlig endringshastighet	15
avrund, round()	132
avslutt funksjon, EndFunc	58
avslutt, Exit	49

B

begrensningsoperator " "	191
begrensningsoperator, rekkefølge av beregning	213

e i en potens, $e^{\wedge}()$	44, 49	Exit, avslutt	49
E, eksponent	187	$\exp()$, e i en potens	49
$e^{\wedge}()$, e i en potens	44	$\text{expr}()$, streng til uttrykk	50
eff), omregn nominell til effektiv rente	44	ExpReg, eksponensiell regresjon	50
effektiv rente, eff()	44		
egendefinerte funksjoner	36	F	
egendefinerte funksjoner og programmer	37-38	factor(), faktor	52
egenvektor, eigVc()	45	faktor, factor()	52
egenverdi, eigVl()	45	fakultet, !	183
eigVc(), egenvektor	45	feil og problemløsning	
eigVl(), egenverdi	45	send feil, SendFeil	111
ekskludering med " " operator	191	slett feil, SlettFeil	23
eksponensiell regresjon, ExpReg	50	feilkoder og meldinger	227
eksponent, E	187	Fill, matrise fyller	53
eksponenter		finansielle funksjoner, tvnFV()	160
sjablon for	1	finansielle funksjoner, tvml()	161
ekte brøk, propFrac	115	finansielle funksjoner, tvnN()	161
eliminasjonsform, ref()	123	finansielle funksjoner, tvnPmt()	161
eller (Boolsk), eller	108	finansielle funksjoner, tvnPv()	162
eller, Boolsk operator	108	FiveNumSummary	53
else if, Elseif	46	fjerdegrads regresjon, QuartReg	118
else, Else	69	fjern	
Elseif, else if	46	åpne elementer fra liste	39
en-variabel-statistikk, OneVar	107	floor(), nedre	54
end		For	54
For...EndFor	54	for, For	54
if, Endif	69	For, for	54
stigningstall, EndLoop	90	fordelingsfunksjoner	
while, EndWhile	168	binomCdf()	19, 74
end if, Endif	69	binomPdf()	19
end stigningstall, EndLoop	90	c 2 Pdf()	22
end while, EndWhile	168	c22-veis()	21
endfunksjon, EndFunc	58	invNorm()	74
Endret internrente av retur, mirr(),	96	invt()	75
EndTry, avslutt prøv	158	normCdf()	104
EndWhile, end while	168	normPdf()	105
enhetsvektor, unitV()	164	poissCdf()	111
enten ... eller ..., Boolsk eksklusiv		poissPdf()	112
eller	168	tCdf()	154
EOS (Equation Operating System) ..	213	tPdf()	157
Equation Operating System		χ^2 Cdf()	21
(Ligningsoperativsystem)		χ^2 GOF()	22
(EOS)	213	fordelingsfunksjoner	
er lik, =	179	Inv χ^2 ()	73
erstatning med " " operator	191	Forespør	126
etikettNavn, Lbl	76	ForespørStr	128
euler(), Euler function	47	format(), formatstreng	55
		formatstreng, format()	55

første deriverte		grupper, teste låsestatus	65
sjablon for	5		
fortegn, sign()	140	H	
fpart(), funksjonsdel	55	heksadesimal	
frekvens()	57	indikator, 0h	193
fremstilling i		vise, 4Grunntall16	18
grader/minutter/sekunder	189	heltall, int()	72
fremstilling minutter,	189	heltallsdel, iPart()	75
fremstilling sekunder, "	189	Hent	60, 203
freqTable()	56	hent/returner	
Frobenius-norm, norm()	104	variabelinformasjon, getVarInfo()	65, 67
Func, funksjon	58	høyre(), høyre	129
Func, programfunksjon	58	høyre, høyre()	129
funksjoner		høyre, right()	73
del, fpart()	55	hurtigtaster	211
egendefinere	36	hvis, Hvis	69
programfunksjon, Func	58	hyperbolic (hyperbolsk)	
funksjoner og variabler		invers tangens, tanh ⁻¹ ()	154
kopiere	25	hyperbolsk	
fyll	201-202	cosinus, cosh()	28
		invers cosinus, cosh ⁻¹ ()	29
G		invers sinus, sinh ⁻¹ ()	143
G, gradianer	187	sinus, sinh()	142
gå til, Goto	68	tangens, tanh()	154
gcd(), største felles divisor	59	I	
geomCdf()	59	identitet(), identitetsmatrise	69
geomPdf()	60	identitetsmatrise, identitet()	69
getDenom(), lesNevner	61	If, if	69
getKey()	61	ifFn()	70
getLangInfo(), hent/returner		ikke både...og, Boolsk operator	100
språkinformasjon	65	ikke, Boolsk operator	105
getLockInfo(), tester låsestatus av		imag(), imaginær del	71
variabel eller		imaginær del, imag()	71
variabelgruppe	65	lndir.ref, #	187
getNum(), les/returner teller	66	innenfor streng, inString()	72
GetStr	67	innstillinger, les aktuell	65
getType(), get type of variable	67	inString(), innenfor streng	72
getVarInfo(), les/returner		int(), heltall	72
variabelinformasjon	67	intDiv(), dividere heltall	72
gjennomsnitt, mean()	92	interpolere(), interpolere	73
gjennomsnittlig endringshastighet,		invers cosinus, cos ⁻¹ ()	27
avgRC()	15	invers kumulativ normalfordeling	
Goto, gå til	68	(invNorm()	74
grader fremstilling, °	189	invers sinus, sin / ()	142
grader/minutter/sekunder-visning,		invers tangens, tan ⁻¹ ()	153
DMS	43	invers, x ⁻¹	191
gradian fremstilling, G	187		
grupper, låse og låse opp	87, 164		

invF()	73	lesTeller, getNum()	66
invNorm(), invers kumulativ normalfordeling)	74	lesModus(), les modus	137
invnt()	75	lesModus(), les modus-innstillinger	65
Invx ² ()	73	libShortcut(), lage snarveier til bibliotekobjekter	78
iPart(), heltallsdel	75	ligningssystemer (2-ligning) sjablon for	3
irr(), internrente		ligningssystemer (N-ligning) sjablon for	3
internrente, irr()	75	lineær regresjon, LinRegAx	79
isPrime(), primtest	76	lineær regresjon, LinRegBx	78, 80
isVoid(), test for tomrom	76	LinRegBx, lineær regresjon	78
		LinRegMx, lineær regresjon	79
		LinRegtIntervals, lineær regresjon	80
		LinRegtTest	82
		linSolve()	83
		list►mat(), liste til matrise	84
		liste til matrise, list►mat()	84
		liste, antall betingede elementer i	30
		liste, antall elementer i	30
		lister	
		differens, @liste()	84
		differenser i en liste, @list()	84
		kryssprodukt, crossP()	31
		kumulativ sum, cumulativeSum()	34
		liste til matrise, list►mat()	84
		maksimum, max()	92
		matrise til liste, mat Δliste()	91
		midtstreng, mid()	94
		minimum, min()	95
		nye, newList()	101
		prikk produkt, dotP()	43
		produkt, product()	115
		sorter fallende, SortD	145
		sorter stigende, SortA	145
		summering, sum()	150-151
		tomme elementer i	209
		utvid/sett sammen, utvid()	14
		ln(), naturlig logaritme	84
		LnReg, logaritmisk regresjon	85
		Local, lokal variabel	86
		Logaritme	
		sjablon for	2
		logaritmer	84
		logaritmisk regresjon, LnReg	85
		logisk dobbel implikasjon, ⇔	183
		logisk implikasjon, ⇒	182, 211
		Logistic, logistisk regresjon	87
K			
kolUtvid	24		
kombinasjoner, nCr()	101		
Kommandoen Wait	166		
kommentar, ©	193		
kompleks			
konjugert, conj()	24		
konstruer matrise, constructMat()	25		
kopiere variabel eller funksjoner, CopyVar	25		
korrelasjonsmatrise, corrMat()	26		
korrMat(), korrelasjonsmatrise	26		
kryssprodukt, crossP()	31		
kubisk regresjon, CubicReg	33		
kumulativ sum, cumulativeSum()	34		
kvadratisk regresjon, QuadReg	117		
kvadratro			
sjablon for	1		
kvadratro, √()	184		
kvadratro, ‡()	146		
L			
lagre			
symbol, &	192		
Lås, lås variabel eller variabelgruppe	87		
låse opp variabler og variabelgrupper	164		
låse variabler og variabelgrupper	87		
Lbl, etikettNavn	76		
lcm, minste felles multiplum	77		
left(), venstre	77		
legg til, &	183		
lengde på streng	40		
les			
modus, lesModus()	137		
les/returner			
lesNevner, getDenom()	61		

LogisticD, logistisk regresjon	89	produkt, product()	115
logistisk regresjon, Logistic	87	QR faktorisering, QR	116
logistisk regresjon, LogisticD	89	radaddisjon, rowAdd()	132
lokal variabel, Local	86	raddimensjon, rowDim()	132
lokal, Local	86	radhandling, mRow()	97
Loop, stigningstall	90	radmultiplikasjon og addisjon, mRowAdd()	97
LU (lower-upper), matrisens nedre- øvre dekomposisjon	91	radnorm, rowNorm()	133
		radskift, rowSwap()	133
		redusert eliminasjonsform, rref()	133
		summering, sum()	150-151
M		tilfeldig, tilfMat()	121
maksimum, max()	92	transponert, T	152
mat \blacktriangleright list(), matrise til liste	91	undermatrise, subMat()	150
matrices (matriser)		utvid/sett sammen, augment()	14
undermatrise, subMat()	152	max(), maksimum	92
matrise (1 \times 2)		mean(), gjennomsnitt	92
sjablon for	4	med (gitt at),	191
matrise (2 \times 1)		median-median linjeregresjon, MedMed	93
sjablon for	4	median(), median	93
matrise (2 \times 2)		median, median()	93
sjablon for	4	MedMed, median-median linjeregresjon	93
matrise (m \times n)		mid(), midtstreng	94
sjablon for	4	midtstreng, mid()	94
matrise til liste, mat Δ liste()	91	min(), minimum	95
matriser		mindre enn eller lik, {	181
determinant, det()	39	minimum, min()	95
diagonal, diag()	40	minste felles multiplum, lcm	77
dimensjon, dim()	40	mirr(), endret internrente av retur	96
egenvektor, eigVc()	45	mod(), modul	96
egenverdi, eigVl()	45	modul, mod()	96
eliminerasjonsform, ref()	123	modus-innstillinger, lesModus()	65
fill, Fylle	53	moduser	
identitet, identitet()	69	lesing, lesModus()	137
kolonnedimensjoner, colDim()	24	mRow(), matrise radhandling	97
kolonnenorm, colNorm()	24	mRowAdd(), matrise radmultiplikasjon og addisjon	97
kumulativ sum, cumulativeSum()	34	multiplisere, *	176
liste til matrise, list \blacktriangleright mat()	84	Multipel lineær regresjon ttest	98
maksimum, max()	92	MultReg	97
matrise til liste, mat Δ liste()	91	MultRegIntervals() (MultRegIntervaller)	98
minimum, min()	95	MultRegTests() (MultRegTester)	98
nedre-øvre dekomposisjon, LU (lower-upper)	91		
nye, newMat()	101		
prikk addisjon, .+	177		
prikk divisjon, .P	178		
prikk multiplikasjon, .*	178		
prikk potens, .^	178		
prikk subtraksjon, .N	178		

N

n-te rot	
sjablon for	1
når, when()	167
naturlig logaritme, ln()	84
nCr(), kombinasjoner	101
nDerivative(), numerisk derivert	101
nedre, floor()	54
negasjon, legge inn negative tall	214
netto nåverdi, npv()	106
newList(), ny liste	101
newMat(), ny matrise	101
nfMax(), numerisk	
funksjonsmaksimum	102
nfMin(), numerisk	
funksjonsminimum	102
nInt(), numerisk integral	102
nom(), omregn effektiv til nominell	
rente	103
nominell rente, nom()	103
norm(), Frobenius-norm	104
normCdf()	104
normPdf()	105
nPr(), permutasjoner	105
npv(), netto nåverdi	106
nSolve(), numerisk løsning	106
numerisk	
derivert, nDeriv()	102
derivert, nDerivative()	101
integral, nInt()	102
løsning, nSolve()	106
ny	
liste, newList()	101
matrise, newMat()	101

O

objekter	
lage snarveier til bibliotek	78
omregne	
►Rad	120
4Grad	69
omregnings-operator (#)	214
OneVar, en-variabel-statistikk	107
operatører	
rekkefølge av behandling	213
ord(), numerisk tegnkode	110
Overfører øyeblikkelig kontroll til	35

den neste iterasjonen i	
aktuell løkke (For, While,	
eller Loop).	19-20, 31
øvre, ceiling()	19-20, 31

P

P►Rx(), rektangulær x-koordinat	110
P►Ry(), rektangulær y-koordinat	110
Pdf()	56
permutasjoner, nPr()	105
poissCdf()	111
poissPdf()	112
polar	
koordinat, R►Pr()	120
koordinat, R►Pθ()	119
vektor-visning, ►Polar	112
polyEval(), behandle polynom	113
polynomer	
behandle, polyEval()	113
tilfeldig, tilfPoly()	122
PolyRoots()	113
potens, ^	176
potensregresjon,	
PowerReg	113, 126, 128, 155
PowerReg, potensregresjon	113
Prgm, definer program	114
prikkP	
addere, +	177
divisjon, ./	178
multiplikasjon, .*	178
potens, ^	178
produkt, dotP()	43
subtraksjon, -	178
primtallstest, isPrime()	76
prodSeq()	115
product(), produkt	115
produkt, P()	184
produkt, product()	115
programmer	
definere felles (offentlig)	
bibliotek	38
definere privat bibliotek	37
programmer og programmering	
slett feil, SlettFeil	23
try, Try	158
vis I/O-skjerm, Vis	134
vis I/O skjerm, Vis	41

sjabloner		standardavvik, stdDev() ...	148-149, 165
([]), produkt	5	tilfeldig startverdi, RandSeed	122
2-delers stykkevis funksjon	2	tilfeldig norm, tilfNorm()	121
Absoluttverdi	3-4	to-variable resultater, TwoVar	162
andrederivert	6	varians, variance()	165
bestemt integral	6	stdDevPop(), populasjonens	
brøk	1	standardavvik	148
e eksponent	2	stdDevSamp(), utvalgets	
eksponent	1	standardavvik	149
første deriverte	5	stigningstall, Loop	90
kvadratrot	1	Stoppkommando	150
ligningssystemer (2-ligning)	3	større enn eller lik med,	182
ligningssystemer (N-ligning)	3	større enn, >	181
Logaritme	2	største felles divisor, gcd()	59
matrise (1 × 2)	4	streng	
matrise (2 × 1)	4	dimensjon, dim()	40
matrise (2 × 2)	4	lengde	40
matrise (m × n)	4	strenger	
n-te rot	1	brukes for å opprette	
stykkevis funksjon (N-delers)	2	variabelnavn	214
sum (G)	5	format, format()	55
skift, shift()	139	formatering	55
slett		høyre, høyre()	73, 129
åpne elementer fra liste	39	Indir.ref, #	187
feil, SlettFeil	23	innenfor, inString	72
Slett	197	legg til, &	183
slettAZ	23	midtstreng, mid()	94
slette		rotete, rotate()	130
variabel, DelVar	39	skift, shift()	139
SlettFeil, slett feil	23	streng til uttrykk, expr()	50
snarveier, tastatur	211	tegnkode, ord()	110
SortA, sorter stigende	145	tegnstreng, char()	20
SortD, sorter fallende	145	uttrykk til streng, string()	150
sorterer		venstre, left()	77
fallende, SortD	145	string(), uttrykk til streng	150
stigende, SortA	145	strings	
språk		right, right()	47, 166
hente språkinformasjon	65	stykkevis funksjon (N-delers)	
sqrt(), kvadratrot	146	sjablon for	2
standardavvik, stdDev()	148-149, 165	stykkevis()	111
stat.resultater	147	subMat(), undermatrise	150, 152
stat.verdier	148	subtrahere, -	175
statistikk		sum (G)	
en-variabel-statistikk, OneVar	107	sjablon for	5
fakultet, !	183	sum av hovedbetalinger	186
gjennomsnitt, mean()	92	sum av rentebetalinger	185
kombinasjoner, nCr()	101	sum(), summering	150
median, median()	93	sum, Σ()	185
permutasjoner, nPr()	105	sumIf()	151

summering, sum()	150	to-variable resultater, TwoVar	162
sumSeq()	152	tomme (åpne) elementer	209
svar (siste), Ans	12	tomrom, test for	76
T			
T, transponert	152	trace()	157
t test, tTest	159	transponert, T	152
$\tan^{-1}()$, invers tangens	153	Try, feil håndteringskommando	158
tan(), tangens	152	try, Try	158
tangens, tan()	152	Try, try	158
$\tanh^{-1}()$, hyperbolsk invers tangens	154	tTest, t test	159
tanh(), hyperbolsk tangens	154	tTest_2Samp, to-utvalgs t-test	160
tCdf(), sannsynlig student t-fordeling	154	TVM-argumenter	162
tegn		tvmFV()	160
streng, char()	20	tvmI()	161
tegnkode, ord()	110	tvmN()	161
tegne	198-200	tvmPmt()	161
tegnstreng, char()	20	tvmPV()	162
Tekstkommando	155	TwoVar, to-variable resultater	162
teller dager mellom datoer, dbd()	35	U	
tellIf(), antall betingede elementer i en liste	30	ulik, ≠	180
test for tomrom, isVoid()	76	undermatrise, subMat()	150, 152
Test_2S, 2_utvalg F test	57	unitV(), enhetsvektor	164
tidsverdi for penger, antall betalinger	161	unLock, lås opp variabel eller variabelgruppe	164
tidsverdi for penger, betalingsbeløp	161	uttrykk	
tidsverdi for penger, Fremtidig verdi	160	streng til uttrykk, expr()	50
tidsverdi for penger, nåverdi	162	utvid/sett sammen, augment()	14
tidsverdi for penger, Rente	161	V	
tier-potens, $10^{\wedge}()$	190	variabel	
tilf(), tilfeldig nummer	120	opprette navn fra en tegnstreng	214
tilfBin, tilfeldig tall	120	variabler	
tilfeldig		lokal, Local	86
matrise, tilfMat()	121	slette alle enkelttegn	23
norm, tilfNorm()	121	slette, DelVar	39
polynom, tilfPoly()	122	variabler og funksjoner	
startverdi, RandSeed	122	kopiere	25
tilfeldig utvalg	122	variabler, låse og låse opp	65, 87, 164
tilfInt(), tilfeldig heltall	121	varians, variance()	165
tilfNorm(), tilfeldig norm	121	varPop()	165
tilfPoly(), tilfeldig polynom	122	varSamp(), utvalgets varians	165
tilfSamp()	122	varselkoder og meldinger	227
tilMat(), tilfeldig matrise	121	vektorer	
tilnærmet, approx()	12	enhet, unitV()	164
tInterval, t konfidensintervall	156	kryssprodukt, crossP()	31
tInterval_2Samp, toutvalg t konfidensintervall	156	Overfører øyeblikkelig kontroll til den neste	35

iterasjonen i aktuell løkke (For, While, eller Loop).		Δ	
prikk produkt, dotP()	43	Δlist(), differensliste	84
venstre, left()	77		
verken ... eller, Boolsk operator	103	X	
vinkel, angle()	9	χ ² Cdf()	21
vis data, Vis	41, 134	χ ² GOF	22
Vis, vis data	41, 134		
vise som			
binær, 4Grunntall2	16		
desimalt heltall, 4Grunntall10	18		
desimalvinkel, ►DD	36		
grader/minutter/sekunder, ►DMS	43		
heksadesimal, 4Grunntall16	18		
polar vektor, ►Polar	112		
rektangulær vektor, ►Rect	123		
sfærisk vektor, ►Sphere	146		
sylindrisk vektor, ►Cylind	35		
		W	
warnCodes(), Warning codes	166		
when(), når	167		
while, While	168		
While, while	168		
		X	
x ⁻¹ , resiprok	191		
x ² , kvadrat	177		
XNOR	183		
		Z	
zInterval, z konfidensintervall	169		
zInterval_1Prop, en-proporsjons z konfidensintervall	170		
zInterval_2Prop, to-proporsjons z konfidensintervall	170		
zInterval_2Samp, to-utvalgs z konfidensintervall	171		
zTest	172		
zTest_1Prop, en-proporsjons z-test	172		
zTest_2Prop, to-proporsjons z-test	173		
zTest_2Samp, to-utvalgs z-test	174		