

TI-Nspire™ CX

Referanseguiden

Viktig Informasjon

Dersom ikke annet er uttrykkelig nevnt i Lisensen som finnes vedlagt programmet, gir ikke Texas Instruments noen garanti, verken uttrykt eller underforstått, herunder, men ikke begrenset til noen impliserte garantier for salgbarhet og egnethet for et bestemt formål, med hensyn til noen som helst programmer eller bokmaterialer som kun er tilgjengelig på et "som det er"-grunnlag. Ikke i noen tilfeller kan Texas Instruments bli holdt ansvarlig overfor noen for spesielle, indirekte, tilfeldige eller følgeskader i forbindelse med eller som et resultat av anskaffelsen eller bruken av disse materialene. Texas Instruments' eneste og eksklusive ansvar, uten hensyn til aksjonsformen, kan ikke overstige den summen som er blitt fremsatt i lisensen for programmet. I tillegg kan ikke Texas Instruments bli holdt ansvarlig for noen krav av noe slag mot bruken av disse materialene av en annen part.

© 2020 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	36
E	45
F	53
G	60
I	71
L	78
M	93
N	102
O	111
P	113
Q	120
R	123
S	137
T	157
U	169
V	170
W	171
X	173
Z	174
Symboler	180
TI-Nspire™ CX II – Tegnekommandoer	203
Grafikkprogrammering	203
Grafikkskjerm	203
Standardvisning og innstillinger	204
Feilmeldinger på grafikkskjerm	205
Ugyldige kommandoer i grafikkmodus	205
C	206
D	207
F	210
G	212
P	213
S	215
U	217

Tomme (åpne) elementer	218
Snarveier/hurtigtaster for å legge inn matematiske uttrykk	220
EOS™ (Ligningsoperativsystem)-hierarkiet	222
TI-Nspire CX II – TI-Basic programmeringsfunksjoner	224
Auto-innrykk i Programmeringseditor	224
Forbedrede feilmeldinger for TI-Basic	224
Konstanter og verdier	227
Feilkoder og feilmeldinger	228
Varselkoder og meldinger	236
Generell informasjon	238
Hjelp på nettet (online)	238
Kontakt TIs brukerstøtte	238
Service og garantiinformasjoner	238
Stikkordregister	239

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 182.

Eksempel:

12

8·2

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 183.

Eksempel:

2³

8

Kvadratrot-sjablon

ctrl

x²

taster

Merk: Se også √() (kvadratrot), side 192.

Eksempel:

√4

√{9,16,4}

2

{3,4,2}

N-te rot-sjablon

ctrl

^

taster

Merk: Se også rot(), side 134.

Eksempel:

Uttrykkssjabloner 1

N-te rot-sjablon

ctrl **^** **taster**

$$\sqrt[3]{8} \quad 2$$

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

e eksponent-sjablon

e^x **tast**

e

$$e^1 \quad 2.71828182846$$

Naturlig grunntall e opphøyd i en eksponent

Merk: Se også **e[^]()**, side 45.

Logaritme-sjablon

ctrl **10^x** **taster**

log ()

Eksempel:

$$\log_4(2) \quad 0.5$$

Beregner logaritme til et spesifisert grunntall. Hvis grunntallet er forhåndsinnstilt på 10, utelates grunntallet.

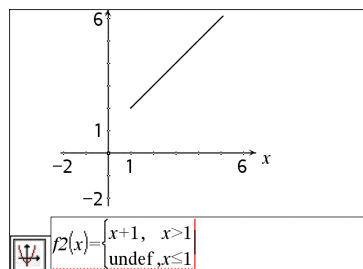
Merk: Se også **log()**, side 89.

Stykkevis sjablon (2-delers)

Katalog > **log**

{ }
{ }

Eksempel:



Lar deg opprette uttrykk og betingelser for en to-delers stykkevis definert funksjon. For å legge til en del, klikk på sjablonen og gjenta sjablonen.

Merk: Se også **stykkevis()**, side 115.

Stykkevis sjablon (N-delers)

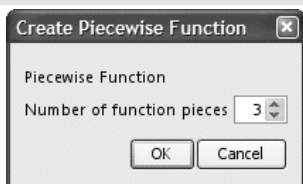
Katalog > **log**

Lar deg opprette uttrykk og betingelser for en N-delers stykkevis definert funksjon. Ber om N .

Eksempel:

Stykkevis sjablon (N-delers)

Katalog >



Se eksemplet for Stykkevis sjablon (2-delers).

Merk: Se også `stykkevis()`, side 115.

Sjablon for ligningssystemer med 2 ukjente

Katalog >



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

Merk: Se også `system()`, side 156.

Eksempel:

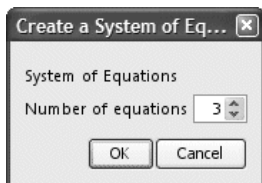
$$\begin{aligned} &\text{solve} \left(\begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y \right) && x=\frac{5}{2} \text{ and } y=-\frac{5}{2} \\ &\text{solve} \left(\begin{cases} y=x^2-2 \\ x+2 \cdot y=-1 \end{cases}, x, y \right) && x=-\frac{3}{2} \text{ and } y=\frac{1}{4} \text{ or } x=1 \text{ and }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

Sjablon for ligningssystemer med N ukjente

Katalog >



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



Merk: Se også `system()`, side 156.

Eksempel:

Se eksemplet for Sjabloner for ligningssystemer (2 ligninger).

Sjablon for absoluttverdi

Katalog >



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:

$$30^{\circ}15'10'' \quad 0.528011$$

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 > 

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

Eksempel:

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

Opretter en 2 x 2-matrise.

Matrise-sjablon (1 x 2)

Katalog > 

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

Eksempel:

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

Matrise-sjablon (2 x 1)

Katalog > 

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

Eksempel:

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

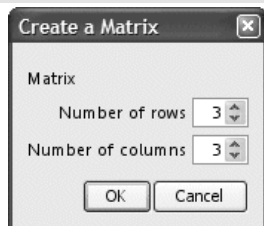
Matrise-sjablon (m x n)

Katalog > 

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

Eksempel:

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



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)

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

Eksempel:

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

Merk: Se også $\Sigma()$ (`sumSeq`), side 193.

Produkt-sjablon (Π)

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

Eksempel:

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

Merk: Se også $\Pi()$ (`prodSeq`), side 192.

Første derivert-sjablon

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

Eksempel:

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

Den første deriverte sjablonen kan brukes for å beregne førstederiverte i et punkt numerisk ved hjelp av automatiske derivasjonsmetoder.



Merk: Se også **d()** (derivert), side 191.

Andre derivert-sjablon



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

Eksempel:

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

Den andre deriverte sjablonen kan brukes for å beregne andrederiverte i et punkt numerisk ved hjelp av automatiske derivasjonsmetoder.

Merk: Se også **d()** (derivert), side 191.

Bestemt integral-sjablon



$$\int_a^b \square dx$$

Eksempel:

$$\int_0^{10} x^2 dx \quad 333.333$$

Den bestemte integral--sjablonen kan brukes for å beregne den bestemte integralet numerisk ved hjelp av den samme metoden som **nInt()**.

Merk: Se også **nInt()**, side 106.

Alfabetisk oversikt

Elementer med navn som ikke er alfabetiske (som f.eks. +, !, og >) er opplistet på slutten av dette avsnittet (side 180). 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(VerdiI)⇒verdi

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

{ 1.5708,1.0472 }

abs(ListeI)⇒liste

$|2-3 \cdot i|$

3.60555

abs(MatriseI)⇒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

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

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 167.

- 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.

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 193, og **bal()**, side 15.

and

BoolskUttr1 and *BoolskUttr2* $\Rightarrow$ *Boolsk uttrykk*

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
--------------	---

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

Merk: Et binært innlegg kan bestå av opptil 64 siffer (i tillegg til prefikset 0b). Et heksadesimalt innlegg kan bestå av opptil 16 siffer.

angle() vinkel

angle(*Verdi1*)⇒*verdi*

I Grader-vinkelmodus:

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

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

$\text{angle}(1+i)$	0.785398
$\text{angle}(\{1+2 \cdot i, 3+0 \cdot i, 0-4 \cdot i\})$	$\{1.10715, 0, -1.5708\}$

angle(*Liste1*)⇒*liste*

angle(*Matrise1*)⇒*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

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 151).

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

Utdata-variabel	Beskrivelse
stat.F	Verdi av F -statistikken
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 151).

LevRad=0 for Blokk

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

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 ⇒ <i>verdi</i>	56	56
Returnerer resultatet av det sist behandlede uttrykket.	56+4	60
	60+4	64

approx() (tilnærm)	Katalog > 
approx (<i>Verdi1</i>)⇒ <i>verdi</i>	approx($\frac{1}{3}$)0.333333 approx($\left\{\frac{1}{3}, \frac{1}{9}\right\}$)$\{0.333333, 0.111111\}$ approx($\{\sin(\pi), \cos(\pi)\}$)$\{0., -1.\}$ approx($\left[\sqrt{2} \quad \sqrt{3}\right]$)$[1.41421 \quad 1.73205]$ approx($\left[\frac{1}{3} \quad \frac{1}{9}\right]$)$[0.333333 \quad 0.111111]$
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↵ .	
approx (<i>Liste1</i>)⇒ <i>liste</i>	approx($\{\sin(\pi), \cos(\pi)\}$)$\{0., -1.\}$ approx($\left[\sqrt{2} \quad \sqrt{3}\right]$)$[1.41421 \quad 1.73205]$
approx (<i>Matrise1</i>)⇒ <i>matrise</i>	
Returnerer en liste eller <i>matrise</i> hvor hvert element er blitt behandlet til en desimalverdi, hvis mulig.	

►approxFraction()	Katalog > 
<i>Verdi</i> ► approxFraction ([<i>Tol</i>])⇒ <i>verdi</i>	$\frac{1}{2} + \frac{1}{3} + \tan(\pi)$ 0.833333
<i>Liste</i> ► approxFraction ([<i>Tol</i>])⇒ <i>liste</i>	0.8333333333333333 ► approxFraction (5.E-14)
<i>Matrise</i> ► approxFraction ([<i>Tol</i>])⇒ <i>matrise</i>	$\frac{5}{6}$
Returnerer argumentet som en brøk med en toleranse på <i>Tol</i> . Hvis <i>tol</i> utelates, brukes en toleranse på 5.E-14.	$\{\pi, 1.5\}$ ► approxFraction (5.E-14)
Merk: Du kan sette inn denne funksjonen fra datamaskintastaturet ved å skrive @> approxFraction (...).	$\left\{ \frac{5419351}{1725033}, \frac{3}{2} \right\}$

approxRational()	Katalog > 
approxRational (<i>Verdi</i> [, <i>Tol</i>])⇒ <i>verdi</i>	approxRational (0.333, 5·10 ⁻⁵) $\frac{333}{1000}$
approxRational (<i>Liste</i> [, <i>Tol</i>])⇒ <i>liste</i>	approxRational ({0.2, 0.33, 4.125}, 5.E-14)
approxRational (<i>Matrise</i> [, <i>Tol</i>])⇒ <i>matrise</i>	$\left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$
Returnerer argumentet som en brøk med en toleranse på <i>Tol</i> . Hvis <i>Tol</i> 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 30.
-----------------	---

arccoth()	Se coth⁻¹() , side 30.
------------------	--

arccsc()	Se csc⁻¹() , side 33.
-----------------	---

arccsch()	Se $\text{csch}^{-1}()$, side 34.
arcsec()	Se $\text{sec}^{-1}()$, side 137.
arcsech()	Se $\text{sech}^{-1}()$, side 138.
arcsin()	Se $\sin^{-1}()$, side 146.
arcsinh()	Se $\sinh^{-1}()$, side 147.
arctan()	Se $\tan^{-1}()$, side 158.
arctanh()	Se $\tanh^{-1}()$, side 159.

() (utvid/sett sammen)	Katalog > 						
augment(Liste1, Liste2)⇒liste Returnerer en ny liste som er <i>Liste2</i> lagt til på slutten av <i>Liste1</i> . augment(Matrise1, Matrise2)⇒matrise Returnerer en ny matrise som er <i>Matrise2</i> lagt til på <i>Matrise1</i> . Når tegnet “,” brukes, må matrisen ha like raddimensjoner, og <i>Matrise2</i> er lagt til på <i>Matrise1</i> som nye kolonner. Endrer ikke <i>Matrise1</i> eller <i>Matrise2</i> .	$\text{augment}(\{1, -3, 2\}, \{5, 4\}) \quad \{1, -3, 2, 5, 4\}$ <table> <tr> <td> $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$ </td> <td> $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ </td> </tr> <tr> <td> $\begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow m2$ </td> <td> $\begin{bmatrix} 5 \\ 6 \end{bmatrix}$ </td> </tr> <tr> <td> $\text{augment}(m1, m2)$ </td> <td> $\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$ </td> </tr> </table>	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 \\ 6 \end{bmatrix}$	$\text{augment}(m1, m2)$	$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$						
$\begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow m2$	$\begin{bmatrix} 5 \\ 6 \end{bmatrix}$						
$\text{augment}(m1, m2)$	$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$						

avgRC() (gjsnEH)

Katalog >

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

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

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

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

Returnerer differenskvotienten tatt i positiv retning (gjennomsnittlig endringshastighet).

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

Hvis *verdi* er spesifisert, opphever den eventuell forhåndsildelt 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()

Katalog >

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

bal(*NPmt*,*amortTabell*)⇒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 167.

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

$x:=2$	2
$avgRC(x^2-x+2,x)$	3.001
$avgRC(x^2-x+2,x,.1)$	3.1
$avgRC(x^2-x+2,x,3)$	6

Katalog 

$bal(5,6,5.75,5000,,12,12)$	833.11																												
$tbl:=amortTbl(6,6,5.75,5000,,12,12)$																													
<table><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																												

$N, I, PV, Pmt, FV, PpY, CpY$ og $PmtAt$ er beskrevet i tabellen med TVM-argumenter, side 167.

- Hvis du utelater Pmt , grunninnstilles den til $Pmt = \mathbf{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.
- 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 193.

►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

-2^{63} 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.

2^{63} blir -2^{63} og vises som

0h8000000000000000 i heksadesimal modus

0b100...000 (63 nuller) i binær modus

2^{64} blir 0 og vises som

0h0 i heksadesimal modus

0b0 i binær modus

$-2^{63} - 1$ blir $2^{63} - 1$ og vises som

0h7FFFFFFFFFFFFFFF i heksadesimal modus

0b111...111 (64 1-ere) i binær modus

Heltall1 ►Base10⇒*heltall*

0b10011►Base10 19

0h1F►Base10 31

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

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.

Heltall1 ►Base16⇒*heltall*

256►Base16 0h100

0b111100001111►Base16 0hF0F

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

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 verdidiområde. For mer informasjon, se ►Base2, side 16.

binomCdf()

binomCdf(n, p) ⇒ *liste*

binomCdf

($n, p, nedreGrense, øvreGrense$) ⇒ *tall* hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er lister

binomCdf($n, p, øvreGrense$) for $P(0 \leq X \leq øvreGrense)$ ⇒ *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 $P(X \leq øvreGrense)$, 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)

ceiling(*Verdi I*) ⇒ *verdi*

ceiling(.456)

1.

Returnerer det nærmeste heltallet som er $\geq$ argumentet.

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.

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

centralDiff()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 forhåndsstilt 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.

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

char()Katalog > 

char(*Heltall*) $\Rightarrow$ *tegn*

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

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

 χ^2 2way

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 151).

For informasjon om effekten av tomme elementer i en matrise, se “Tomme (åpne) elementer” (side 218).

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.UtrMat	Matrise av forventet element-telletebll ved antatt nullhypotese
stat.KompMat	Matrise av elementbidrag til chi kvadratstatistikk

 χ^2 Cdf()

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 218).

$\chi^2\text{GOF}$ *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 151).

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 218).

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.Complst	Elementbidrag til chi kvadratstatistikk

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

chi2Pdf(*XVerd,df*) $\Rightarrow \text{tall}$ 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 218).

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 169.

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

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 114, og **Try**, side 163.

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

For et eksempel på **ClrErr**, se eksempel 2 under **Try** -kommandoen, side 163.

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 summene 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**(*Verdi I*) $\Rightarrow$ *verdi***conj**(*Liste I*) $\Rightarrow$ *liste***conj**(*Matrise I*) $\Rightarrow$ *matrise*

Returnerer den komplekse konjugerte av argumentet.

Merk: Alle ubestemte variabler behandles som reelle variabler.

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

constructMat
(*Uttr*,*Var1*,*Var2*,*antRad*,*antKol*)⇒*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*.

$\text{constructMat}\left(\frac{1}{i+j}, i, j, 3, 4\right)$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$
	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$
	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$
	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$

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.

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.

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

katalog >

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

Katalog >

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

trig tast

I Grader-vinkelmodus:

$\cos\left(\left(\frac{\pi}{4}\right)^r\right)$	0.707107
---	----------

$\cos(45)$	0.707107
------------	----------

$$\overline{\cos(\{0,60,90\})} \qquad \{1,0.5,0.\}$$

I Gradian-vinkelmodus:

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

$\cos\left\{\frac{\pi}{4}\right\}$	0.707107
------------------------------------	----------

$\cos(45^\circ)$	0.707107
------------------	----------

I Radian-vinkelmodus:

cos()



Returnerer matrisens cosinus til *kvadratMatrise* *l*. Dette er ikke det samme som å beregne cosinus til hvert element.

Når en skalarfunksjon $f(A)$ virker på *kvadratMatrise* *l* (*A*), beregnes resultatet av algoritmen:

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

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

kvadratMatrise *l* 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.

cos⁻¹()



cos⁻¹(Verdi) ⇒ *verdi*

I Grader-vinkelmodus:

cos⁻¹(Liste) ⇒ *liste*

0.

cos⁻¹(Verdi) returnerer vinkelen som har cosinus lik *Verdi*.

I Gradian-vinkelmodus:

cos⁻¹(Liste) returnerer en liste over invers cosinus for hvert element i *Liste*.

100.

I Radian-vinkelmodus:

$\cos^{-1}()$

 **tast**

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** (...).

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

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

kvadratMatrise I må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$$\cos^{-1}(\{0,0,2,0,5\}) \\ \{1.5708, 1.36944, 1.0472\}$$

I radian-vinkelmodus og rektangulært, kompleks format:

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

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

cosh()

Katalog > 

cosh(Verdi I) $\Rightarrow$ verdi

I Grader-vinkelmodus:

cosh(Liste I) $\Rightarrow$ liste

$$\cosh\left(\left(\frac{\pi}{4}\right)_r\right) \quad 1.74671 \text{E} 19$$

cosh(Verdi I) returnerer hyperbolsk cosinus til argumentet.

cosh(Liste I) returnerer en liste over hyperbolsk cosinus til hvert element i *Liste I*.

cosh(kvadratMatrise I) $\Rightarrow$ kvadratMatrise

I Radian-vinkelmodus:

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

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

kvadratMatrise I må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

cosh⁻¹()**Katalog >** **cosh⁻¹(Verdi I)** ⇒ verdi $\cosh^{-1}(1)$ 0**cosh⁻¹(Liste I)** ⇒ liste $\cosh^{-1}(\{1,2,1,3\})$ $\{0,1.37286,1.76275\}$

cosh⁻¹(Verdi I) returnerer invers hyperbolsk cosinus til argumentet.

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

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

cosh⁻¹(kvadratMatrise I) ⇒ kvadratMatrise

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

kvadratMatrise I må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I radian-vinkelmodus og rektangulært, kompleks format:

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

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

cot() **tast****cot(Verdi I)** ⇒ verdi

I Grader-vinkelmodus:

 $\cot(45)$ 1.**cot(Liste I)** ⇒ liste

Returnerer cotangens av *Verdi I* eller returnerer en liste med cotangens til alle elementene i *Liste I*.

I Gradian-vinkelmodus:

 $\cot(50)$ 1.

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 Radian-vinkelmodus:

 $\cot(\{1,2,1,3\})$ $\{0.642093, -0.584848, -7.01525\}$

cot⁻¹()**trig** **tast****cot⁻¹(Verdi I)** ⇒ *verdi*

I Grader-vinkelmodus:

cot⁻¹(Liste I) ⇒ *liste*cot⁻¹(1) 45.

Returnerer vinkelen som har cotangens lik *Verdi I* eller returnerer en liste som inneholder invers cotangens til hvert element i *Liste I*.

I Gradian-vinkelmodus:

cot⁻¹(1) 50.

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

I Radian-vinkelmodus:

cot⁻¹(1) 0.785398

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

coth()**Katalog** > **coth(Verdi I)** ⇒ *verdi*

coth(1.2) 1.19954

coth(Liste I) ⇒ *liste*

coth({ 1,3,2 }) { 1.31304,1.00333 }

Returnerer hyperbolsk cotangens til *uttrykk I*, eller returnerer en liste med hyperbolsk cotangens til alle elementene i *liste I*.

coth⁻¹()**Katalog** > **coth⁻¹(Verdi I)** ⇒ *verdi*coth⁻¹(3.5) 0.293893**coth⁻¹(Liste I)** ⇒ *liste*coth⁻¹({-2,2,1,6})
{ -0.549306,0.518046,0.168236 }

Returnerer invers hyperbolsk cotangens til *Verdi I* eller returnerer en liste som inneholder invers hyperbolsk cotangens til hvert element i *Liste I*.

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 218.

<code>count(2,4,6)</code>	3				
<code>count({2,4,6})</code>	3				
<code>count(2,{4,6},<table border="1"><tr><td>8</td><td>10</td></tr><tr><td>12</td><td>14</td></tr></table>)</code>	8	10	12	14	7
8	10				
12	14				

countIf() (tellIf)

Katalog > 

countIf(*Liste*, *Kriterium*) ⇒ *verdi*

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 218.

<code>countIf({1,3,"abc",undef,3,1},3)</code>	2
---	---

Teller alle elementer som er lik 3.

<code>countIf({"abc","def","abc",3},"def")</code>	1
---	---

Teller alle elementer som er lik "def."

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

Teller 1 og 3.

<code>countIf({1,3,5,7,9},2<?<8)</code>	3
---	---

Teller 3, 5 og 7.

countIf() (tellIf)

Merk: Se også **sumIf()**, side 155, og **frequency()**, side 58.

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

Teller 1, 3, 7 og 9.

cPolyRoots()

cPolyRoots(Poly,Var)⇒*liste*

<code>polyRoots(y³+1,y)</code>	{-1}
---	------

cPolyRoots(KoeffListe)⇒*liste*

<code>cPolyRoots(y³+1,y)</code>	{-1, 0.5-0.866025 <i>i</i> , 0.5+0.866025 <i>i</i> }
--	--

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

<code>polyRoots(x²+2*x+1,x)</code>	{-1,-1}
---	---------

Poly må være et polynom i utvidet form i én variabel. Ikke bruk utvidede former, som f.eks. $y^2 \cdot y + 1$ or $x \cdot x + 2 \cdot x + 1$

<code>cPolyRoots({1,2,1})</code>	{-1,-1}
----------------------------------	---------

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

Merk: Se også **polyRoots()**, side 117.

crossP() (kryssprodukt)

crossP(Liste1, Liste2)⇒*liste*

<code>crossP({0.1,2.2,-5},{1,-0.5,0})</code>	{-2.5,-5,-2.25}
--	-----------------

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

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

crossP(Vektor1, Vektor2)⇒*vektor*

<code>crossP([1 2 3],[4 5 6])</code>	[-3 6 -3]
<code>crossP([1 2],[3 4])</code>	[0 0 -2]

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.

csc()

 **tast**

csc(*Verdi1*) $\Rightarrow$ *verdi*

I Grader-vinkelmodus:

csc(*Liste1*) $\Rightarrow$ *liste*

$\text{csc}\{45\}$ 1.41421

Returnerer cosekans til *Verdi1* eller
returnerer en liste som inneholder cosekans
til alle elementene i *Liste1*.

I Gradian-vinkelmodus:

$\text{csc}\{50\}$ 1.41421

I Radian-vinkelmodus:

$\text{csc}\left\{\left\{1, \frac{\pi}{2}, \frac{\pi}{3}\right\}\right\}$ $\{1.1884, 1., 1.1547\}$

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

 **tast**

csc^{-1} (*Verdi1*) $\Rightarrow$ *verdi*

I Grader-vinkelmodus:

csc^{-1} (*Liste1*) $\Rightarrow$ *liste*

$\text{csc}^{-1}\{1\}$ 90.

Returnerer vinkelen som har cosekans lik
Verdi1 eller returnerer en liste som
inneholder invers cosekans til hvert element
i *Liste1*.

I Gradian-vinkelmodus:

$\text{csc}^{-1}\{1\}$ 100.

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

I Radian-vinkelmodus:

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

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

csch()

Katalog > 

csch(*Verdi1*) $\Rightarrow$ *verdi*

$\text{csch}\{3\}$ 0.099822

csch(*Liste1*) $\Rightarrow$ *liste*

$\text{csch}\{1, 2, 1, 4\}$
 $\{0.850918, 0.248641, 0.036644\}$

Returnerer hyperbolsk cosekans til *Verdi1*
eller returnerer en liste med hyperbolsk
cosekans til alle elementene i *Liste1*.

csch⁻¹(*Verdi*) ⇒ *verdi*csch⁻¹(1)

0.881374

csch⁻¹(*Liste1*) ⇒ *liste*csch⁻¹({1,2,1,3})

{0.881374,0.459815,0.32745}

Returnerer invers hyperbolsk cosekans til *Verdi1* eller returnerer en liste som inneholder invers hyperbolsk cosekans til hvert element i *Liste1*.

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 151).

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 numerisk eller streng 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 218).

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
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.

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

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 218.

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 flerlinjede 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
-------	------

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.

$[2 \ 2 \ 3]$ ►Cylind	$[2.82843 \ \angle 0.785398 \ 3.]$
-----------------------	------------------------------------

D

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

MM.DDÅÅ (format som vanligvis brukes i USA)

DDMM.ÅÅ (format som vanligvis brukes i Europa)

►DD

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

Verdi1 ►Decimal⇒verdi

Liste1 ►Decimal⇒verdi

Matrise1 ►Decimal⇒verdi

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)

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 parameterne. 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.

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

Merk: Se også **Define BiblPriv**, side 39 og **Define BiblOff**, side 39.

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 37 og **Define LibPub**, side 39.

Define LibPub (Definer BiblOff)

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 37 og **Define LibPriv**, side 39.

deltaList()

Se Δ List(), side 86.

DelVar

katalog > 

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

$2 \rightarrow a$	2
-------------------	---

DelVar *Var*.

$(a+2)^2$	16
-----------	----

Sletter den angitte variabelen eller variabelgruppen fra minnet.

DelVar <i>a</i>	Done
-----------------	------

$(a+2)^2$	"Error: Variable is not defined"
-----------	----------------------------------

Hvis en eller flere av variablene er låst, viser denne kommandoen en feilmelding og sletter kun de ulåste variablene. Se **unLock**, side 169.

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

<i>aa.a</i> :=45	45
------------------	----

<i>aa.b</i> :=5.67	5.67
--------------------	------

<i>aa.c</i> :=78.9	78.9
--------------------	------

getVarInfo()	<table><tr><td><i>aa.a</i></td><td>"NUM"</td><td>"</td></tr><tr><td><i>aa.b</i></td><td>"NUM"</td><td>"</td></tr><tr><td><i>aa.c</i></td><td>"NUM"</td><td>"</td></tr></table>	<i>aa.a</i>	"NUM"	" 	<i>aa.b</i>	"NUM"	" 	<i>aa.c</i>	"NUM"	" 
<i>aa.a</i>	"NUM"	" 								
<i>aa.b</i>	"NUM"	" 								
<i>aa.c</i>	"NUM"	" 								

DelVar <i>aa</i> .	Done
--------------------	------

getVarInfo()	"NONE"
--------------	--------

delVoid()

Katalog > 

delVoid(*Liste1*) $\Rightarrow$ *liste*

delVoid({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 218.

det()

det(*kvadratMatrise*[,
Toleranse])⇒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   eller stiller modusen **Auto eller Tilnærmet** på Tilnærmet, utføres beregningene med flyttallsaritmetikk.
- Hvis *Toleranse* utelates eller ikke blir brukt, beregnes standardtoleransen som:

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

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

$$\begin{bmatrix} 1.E20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \text{mat1} \quad \begin{bmatrix} 1.E20 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\det(\text{mat1}) \quad 0$$

$$\det(\text{mat1}, 1) \quad 1.E20$$

diag()

diag(*Liste*)⇒matrise

diag(*radMatrise*)⇒matrise

diag(*kolonneMatrise*)⇒matrise

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

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

diag()

Katalog > **diag**(*kvadratMatrise*)⇒*radMatrise*

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

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

kvadratMatrise må være kvadrat.

dim()

Katalog > **dim**(*Liste*)⇒*heltall*

Returnerer dimensjonen av *Liste*.

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

dim(*Matrise*)⇒*liste*

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

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

dim(*Streng*)⇒*heltall*

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

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

Disp (Vis)

Katalog > **Disp** *uttrElStreng1* [, *uttrElStreng2*] ...

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 produktboken.

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

Done

chars(240,243)	

	240 ø
	241 ñ
	242 ó
	243 ó

	<i>Done</i>

DispAt

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

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

DispAt**Eksempel**

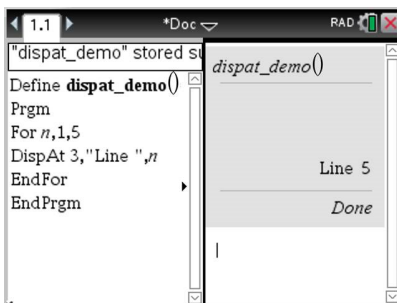
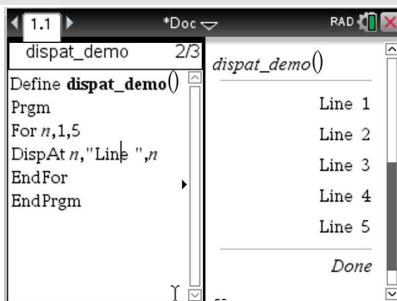
Linjenummeret kan angis som et uttrykk.

Merk: Linjenummeret gjelder ikke hele skjermbildet, 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.

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

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)*Verdi* ►DMS

I Grader-vinkelmodus:

Liste ►DMS

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

Matrise ►DMS

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

Tolker argumentet som en vinkel og viser ekvivalenten DMS (GGGGGG°MM' SS.ss")-tallet. Se °, ', " (side 197) 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)
Katalog >

dotP(Liste1, Liste2)⇒uttrykk

$\text{dotP}(\{1,2\},\{5,6\})$	17
--------------------------------	----

Returnerer "prikk"produktet av to lister.

dotP(Vektor1, Vektor2)⇒uttrykk

$\text{dotP}([1 \ 2 \ 3],[4 \ 5 \ 6])$	32
--	----

Returnerer "prikk"produktet av to vektorer.

Begge må være radvektorer, eller begge må være kolonnevektorer.

E
e^()
 tast

e^(Verdi1)⇒verdi

e^1	2.71828
-------	---------

Returnerer **e** opphøyd i *Verdi1*-potens.

e^{3^2}	8103.08
-----------	---------

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

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

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

e^(Liste1)⇒liste

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

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

$e^{\wedge}()$

e^x tast

$e^{\wedge}(kvadratMatrise1) \Rightarrow kvadratMatrise$

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.

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

eff()

Katalog >

$eff(nominellRente, CpY) \Rightarrow verdi$

$eff(5.75, 12)$

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 106.

eigVc() (egenvektor)

Katalog >

$eigVc(kvadratMatrise) \Rightarrow matrise$

I rektangulært, kompleks format:

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:

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

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

$eigVc(m1)$

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

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

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.

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:

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

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

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

Else

Se If, side 71.

ElseIf

If *BoolskUttr1* **Then**

Blokk1

ElseIf *BoolskUttr2* **Then**

Blokk2

 :

ElseIf *Boolsk UttrN* **Then**

BlokkN

EndIf

 :

 :

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

Define $g(x)$ = Func

 If $x \leq -5$ Then

 Return 5

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

 Return $-x$

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

 Return x

 ElseIf $x = 10$ Then

 Return 3

 EndIf

EndFunc

Done

EndFor	Se For, side 56.
EndFunc	Se Func, side 60.
EndIf	Se If, side 71.
EndLoop	Se Loop, side 92.
EndPrgm	Se Prgm, side 118.
EndTry	Se Try, side 163.
EndWhile	Se While, side 173.

euler ()

Katalog > 

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

Differensialligning:
 $y'=0,001 \cdot y \cdot (100-y)$ og $y(0)=10$

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

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

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

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

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

Bruker Eulers metode for å løse systemet

System av ligninger:

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

med $\text{avhVar}(\text{Var0}) = \text{avhVar0}$ på intervallet $[\text{Var0}, \text{VarMaks}]$. 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 (tilsvarer til rekkefølgen av avhengige variabler i *ListeMedAvhVarer*).

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

Var er den uavhengige variabelen.

ListeMedAvhVarer er en liste over avhengige variabler.

$\{\text{Var0}, \text{VarMaks}\}$ 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 $\text{sign}(\text{VarIntervall}) = \text{sign}(\text{VarMaks} - \text{Var0})$ og løsninger returneres ved $\text{Var0} + i \cdot \text{VarIntervall}$ for alle $i=0,1,2,\dots$ slik at $\text{Var0} + i \cdot \text{VarIntervall}$ er i $[\text{Var0}, \text{VarMaks}]$ (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 $\text{VarIntervall} / \text{eulersIntervall}$.

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

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

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

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

eval(*Uttr*) $\Rightarrow$ *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:=127</i>	127
Send "SET COLOR.BLUE eval(lum) "	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 61), **GetStr** (side 68) og **Send** (side 138).

Exit

Program:

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 flerinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

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

exp()



exp(Verdi1)⇒verdi

e^1	2.71828
-------	---------

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

e^{3^2}	8103.08
-----------	---------

Returnerer **e** opphøyd i *Verdi1*-potens.

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

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

exp(Liste1)⇒liste

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

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

exp(kvadratMatrise1)⇒kvadratMatrise

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

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.

expr (*String*) $\Rightarrow$ *Uttrykk*

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

"Define cube(x)=x^3" $\rightarrow$ *funcstr*

"Define cube(x)=x^3"

expr(*funcstr*) Done

cube(2) 8

ExpReg

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 151).

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 numerisk eller streng 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 218).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $a \cdot (b)^x$
stat.a, stat.b	Regresjonskoeffisienter
stat.r ²	Lineær determinasjonskoeffisient for transformerte data

Utdata-variabel	Beskrivelse
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>

F

factor() (faktor)

Katalog > 

factor(rasjonaltTall) 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 *rasjonaltTall* ikke er et primtall og hvis den nest største faktoren består av mer enn fem siffer.

FCdf

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

FCdf

(
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 *øvGrense* for spesifisert *dfTeller* (frihetsgrader) og *dfNevner*.

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

Fill (Fyll)

Fill *Verdi*, *matriseVar* $\Rightarrow$ *matrise*

Erstatter hvert element i variabel *matriseVar* med *Verdi*.

matriseVar må eksistere allerede.

1 2	$\rightarrow$ <i>amatrix</i>	1 2
3 4		3 4

Fill 1.01,*amatrix* Done

<i>amatrix</i>	1.01 1.01
	1.01 1.01

Fill *Verdi*, *listeVar* $\Rightarrow$ *liste*

Erstatter hvert element i variabel *listeVar* med *Verdi*.

listeVar må eksistere allerede.

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

Fill 1.01,*alist* 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 151.)

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 numerisk eller streng 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 218.

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

floor() (nedre)

floor(Verdi) ⇒ *heltall*

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

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

Argumentet kan være et reelt eller et komplekst tall.

floor(Liste) ⇒ *liste*

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

floor(Matrise) ⇒ *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	Katalog > 	
For <i>Var</i> , <i>Lav</i> , <i>Høy</i> [, <i>Intervall</i>]	Define <i>g()</i> =Func	<i>Done</i>
<i>Blokk</i>	Local <i>tempsum</i> , <i>step</i> , <i>i</i>	
	0 → <i>tempsum</i>	
	1 → <i>step</i>	
	For <i>i</i> ,1,100, <i>step</i>	
	<i>tempsum</i> + <i>i</i> → <i>tempsum</i>	
EndFor	EndFor	
	EndFunc	
<i>Var</i> må ikke være en systemvariabel.	<i>g()</i>	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 flerinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

format()	Katalog > 	
format (<i>Verdi</i> [, <i>formatStreng</i>])⇒ <i>streng</i>	format(1.234567, "f3")	"1.235"
Returnerer <i>Verdi</i> som en tegnstreng basert på formatsjablonen.	format(1.234567, "s2")	"1.23E0"
	format(1.234567, "e3")	"1.235E0"
<i>formatStreng</i> er en streng og må være av formen: “F[n]”, “S[n]”, “E[n]”, “G[n][c]”, hvor [] viser alternative muligheter.	format(1.234567, "g3")	"1.235"
	format(1234.567, "g3")	"1,234.567"
	format(1.234567, "g3,r:")	"1:235"

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.

fPart() (funksjonsdel)

fPart (<i>Uttal</i>) $\Rightarrow$ <i>uttrykk</i>	$fPart(-1.234)$	-0.234
fPart (<i>Liste</i>) $\Rightarrow$ <i>liste</i>	$fPart(\{1,-2,3,7.003\})$	$\{0,-0.3,0.003\}$

fPart(*Matrise*) $\Rightarrow$ *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*) $\Rightarrow$ *tall* hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste

FPdf(*XVerdi*,*dfTeller*,*dfNevner*) $\Rightarrow$ *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 (<i>Liste</i> <i>frekvHeltall</i> <i>Liste</i>) $\Rightarrow$ <i>liste</i>	$freqTable \blacktriangleright list(\{1,2,3,4\}, \{1,4,3,1\})$ $\{1,2,2,2,2,3,3,3,4\}$
	$freqTable \blacktriangleright list(\{1,2,3,4\}, \{1,4,0,1\})$ $\{1,2,2,2,2,4\}$

Returnerer en liste som inneholder elementene fra *Liste1* utvidet i henhold til frekvensene i *frekvHeltallListe*. Denne funksjonen kan brukes til å generere en frekvenstabell for applikasjonen Data og statistikk.

Liste1 kan være enhver gyldig liste.

frekvHeltallListe må ha samme dimensjon som *Liste1* og kun inneholde ikke-negative heltallselementer. Hvert element angir hvor mange ganger det korresponderende *Liste1*-elementet skal gjentas i resultatlisten. En verdi lik null utelater det korresponderende *Liste1*-elementet.

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

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 218.

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 $\{b(1), b(2), \dots, b(n)\}$, er de spesifiserte områdene $\{? \leq b(1), b(1) < ? \leq b(2), \dots, b(n-1) < ? \leq b(n), b(n) > ?\}$. 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 $\{\text{countIf}(\text{liste}, ? \leq b(1)), \text{countIf}(\text{liste}, b(1) < ? \leq b(2)), \dots, \text{countIf}(\text{liste}, b(n-1) < ? \leq b(n)), \text{countIf}(\text{liste}, b(n) > ?)\}$.

datalist = $\{1, 2, e, 3, \pi, 4, 5, 6, \text{"hello"}, 7\}$

$\{1, 2, 2.71828, 3, 3.14159, 4, 5, 6, \text{"hello"}, 7\}$

$\text{frequency}(\text{datalist}, \{2.5, 4.5\}) \quad \{2, 4, 3\}$

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 noen av de definerte stolpene.

Elementer fra *Liste1* som ikke kan “plasseres i en stolpe” ignoreres. Tomme (åpne) elementer ignoreres også. For mer informasjon om tomme elementer, se side 218.

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

Merk: Se også `countif()`, side 31.

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 151).

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 218.

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 = $n1-1$

Utdata-variabel	Beskrivelse
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)

Katalog > 

Func

Blokk

EndFunc

Sjablon for oppretting av brukerdefinert funksjon.

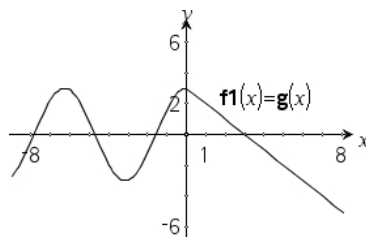
Blokk kan være ett enkelt utsagn, en rekke utsagn adskilt med ":"-tegnet, eller en rekke med utsagn på separate linjer. Funksjonen kan bruke **Returner**-kommandoen for å returnere et spesifikt resultat.

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

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 $g(x)$



G

gcd() (største felles divisor)

Katalog > 

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

$\text{gcd}(18,33)$ 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

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

gcd() (største felles divisor)

Katalog > 

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

gcd(*Matrise1*, *Matrise2*)⇒*matrise*

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

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

geomCdf()

Katalog > 

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

geomCdf(*p*, *øvreGrense*) for $P(1 \leq X \leq \textit{øvreGrense}) \Rightarrow \textit{tall}$ 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 $P(X \leq \textit{øvreGrense})$, sett *nedreGrense* = 1.

geomPdf()

Katalog > 

geomPdf(*p*, *XVerdi*)⇒*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:

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

- 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**.

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.

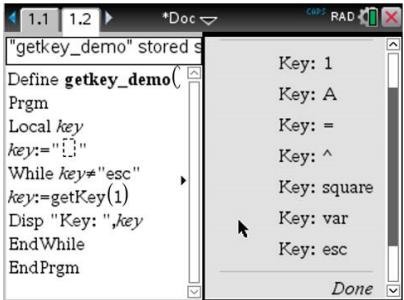
Merk: Se også **GetStr**, side 68 og **Send**, side 138.

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

getDenom() (lesNevner)	Katalog > 
getDenom(<i>Brøk1</i>)⇒ <i>verdi</i>	
Omformer argumentet inn til et uttrykk som har en redusert felles nevner og returnerer så uttrykkets nevner.	
$x:=5; y:=6$	6
$\text{getDenom}\left(\frac{x+2}{y-3}\right)$	3
$\text{getDenom}\left(\frac{2}{7}\right)$	7
$\text{getDenom}\left(\frac{1}{x} + \frac{y^2+y}{y^2}\right)$	30

getKey()	Katalog > 
getKey([0 1]) ⇒ returnString	getKey()
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 oppkallet setter utførelsen av programmet på pause til en tast trykkes ned.	Eksempel 1: 

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"

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
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"
/ (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)

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
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 skrivebordskontrolltaster	Ingen returnerte tegn
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

Hendelse	Enhet	Stasjonær PC – TI-Nspire™ Student Software
		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()

getLockInfo(*Var*)⇒*verdi*

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

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

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

Se **Lock**, side 89, og **unLock**, side 169.

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

GetMode() (lesModus)

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 141.

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

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)

Katalog > 

getNum(*Brøk1*)⇒*verdi*

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

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

GetStr

Hub-meny

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 61 og **Send**, side 138.

getType()

Katalog > 

getType(var)⇒streng

Returnerer en streng som angir dataens typevariabel *var*.

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

$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$
<code>getType(temp)</code>	"LIST"
$3 \cdot i \rightarrow temp$	$3 \cdot i$
<code>getType(temp)</code>	"EXPR"
<code>DelVar temp</code>	<i>Done</i>
<code>getType(temp)</code>	"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(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.

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><tr><td>x</td><td>"NUM"</td><td>"{"</td><td>1</td></tr><tr><td>y</td><td>"LIST"</td><td>"LibPriv"</td><td>0</td></tr><tr><td>z</td><td>"FUNC"</td><td>"LibPub"</td><td>0</td></tr></table>	x	"NUM"	"{"	1	y	"LIST"	"LibPriv"	0	z	"FUNC"	"LibPub"	0
x	"NUM"	"{"	1										
y	"LIST"	"LibPriv"	0										
z	"FUNC"	"LibPub"	0										
getVarInfo(tmp3)	"Error: Argument must be a string"												
getVarInfo("tmp3")	<table><tr><td>volcy12</td><td>"NONE"</td><td>"LibPub"</td><td>0</td></tr></table>	volcy12	"NONE"	"LibPub"	0								
volcy12	"NONE"	"LibPub"	0										

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.

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

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 produkthåndboken.

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

► Grad

Uttr1 ► **Grad** ⇒ *Uttrykk*

Omregner *Uttr1* til gradian vinkelmåling.

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

I Grader-vinkelmodus:

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

I Radian-vinkelmodus:

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

identity()		Katalog > 																
identity(<i>Heltall</i>) $\Rightarrow$ <i>matrise</i>	identity(4)	<table><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
1	0	0	0															
0	1	0	0															
0	0	1	0															
0	0	0	1															
Returnerer identitetsmatrisen med dimensjonen <i>Heltall</i> .																		
<i>Heltallet</i> må være et positivt heltall.																		

Hvis		Katalog > 
Hvis <i>BooleanExpr</i> <i>Utsagn</i>	Define $g(x)$ =Func	Done
	If $x < 0$ Then	
	Return x^2	
	EndIf	
	EndFunc	
Hvis <i>BooleanExpr</i> , så <i>Blokk</i>		
OgHvis	$g(-2)$	4
Hvis <i>BooleanExpr</i> behandles som sann, utføres det enkle utsagnet <i>Utsagn</i> eller blokken av utsagn <i>Blokk</i> før utførelsen fortsetter.		
Hvis <i>BooleanExpr</i> behandles som usann, fortsettes utførelsen uten å utføre utsagnet eller blokken av utsagn.		
<i>Blokk</i> 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 flerinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

Hvis <i>BooleanExpr</i> , så <i>Blokk1</i>	Define $g(x)$ =Func	Done
	If $x < 0$ Then	
	Return $-x$	
	Else	
	Return x	
	EndIf	
	EndFunc	
hvis ikke <i>Blokk2</i>	$g(12)$	12
OgHvis	$g(-12)$	12
Hvis <i>BooleanExpr</i> behandles som sann, utføres <i>Blokk1</i> og utelater så <i>Blokk2</i> .		
Hvis <i>BooleanExpr</i> behandles som usann, utelates <i>Blokk1</i> , men <i>Blokk2</i> 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

If $x < -5$ Then

Return 5

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

Return $\neg x$

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

Return x

ElseIf $x = 10$ Then

Return 3

EndIf

EndFunc

Done

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

ifFn()

Katalog >

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* 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

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.

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

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

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

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.

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

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()

imag(ValueI) ⇒ verdi

Returnerer den imaginære delen av argumentet.

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

imag(ListI) ⇒ liste

Returnerer en liste over de imaginære delene av elementene.

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

imag(MatrixI) ⇒ matrise

Returnerer en matrise over de imaginære delene av elementene.

$$\text{imag} \left(\begin{bmatrix} 1 & 2 \\ i \cdot 3 & i \cdot 4 \end{bmatrix} \right) \quad \begin{bmatrix} 0 & 0 \\ 3 & 4 \end{bmatrix}$$

Indireksjon

Se #(), side 195.

inString()

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

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

$$\text{inString}("Hello there", "the") \quad 7$$

$$\text{inString}("ABCEFG", "D") \quad 0$$

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

inString()

Katalog > 

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

int()

Katalog > 

int(*Verdi*) $\Rightarrow$ *heltall*

int(*List1*) $\Rightarrow$ *liste*

int(*Matrix1*) $\Rightarrow$ *matrise*

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.

$\text{int}(-2.5)$	-3.
$\text{int}\left[\begin{bmatrix} -1.234 & 0 & 0.37 \end{bmatrix}\right]$	$\begin{bmatrix} -2. & 0 & 0. \end{bmatrix}$

intDiv()

Katalog > 

intDiv(*Number1*, *Number2*) $\Rightarrow$ *heltall*

intDiv(*List1*, *List2*) $\Rightarrow$ *liste*

intDiv(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrise*

Returnerer heltallsdelen med fortegn av (*Number1* $\div$ *Number2*).

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

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

interpoler ()

Katalog > 

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

Denne funksjonen gjør følgende:

Differensialligning:

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

$rk := rk23(-3 \cdot y + 6 \cdot t + 5, t, y, \{0, 10\}, 5, 1)$					
0.	1.	2.	3.	4.	$\rightarrow$
5.	3.19499	5.00394	6.99957	9.00593	

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

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.

$xValue$ kan være et tall eller en liste med tall.

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

```
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.,6.5,7.,7.5,8.,8.5,9.,9.5,10.}
xlist:=mat►list(rk[1])
{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.}
ylist:=mat►list(rk[2])
{5.,3.19499,5.00394,6.99957,9.00593,10.9978,12.0021,13.995,15.001,16.999,18.001,19.999,21.001,22.999,24.001,25.001,26.999,28.001,29.999,31.001,32.999,34.001,35.001,36.999,38.001,39.999,41.001,42.999,44.001,45.001,46.999,48.001,49.999,51.001,52.999,54.001,55.001,56.999,58.001,59.999,61.001,62.999,64.001,65.001,66.999,68.001,69.999,71.001,72.999,74.001,75.001,76.999,78.001,79.999,81.001,82.999,84.001,85.001,86.999,88.001,89.999,91.001,92.999,94.001,95.001,96.999,98.001,99.999,100.001}
yprimelist:=-3*y+6*t+5|y=ylist and t=xlist
{-10.,1.41503,1.98819,2.00129,1.98221,2.00679,2.01379,2.02121,2.02879,2.03621,2.04379,2.05121,2.05879,2.06621,2.07379,2.08121,2.08879,2.09621,2.10379,2.11121,2.11879,2.12621,2.13379,2.14121,2.14879,2.15621,2.16379,2.17121,2.17879,2.18621,2.19379,2.20121,2.20879,2.21621,2.22379,2.23121,2.23879,2.24621,2.25379,2.26121,2.26879,2.27621,2.28379,2.29121,2.29879,2.30621,2.31379,2.32121,2.32879,2.33621,2.34379,2.35121,2.35879,2.36621,2.37379,2.38121,2.38879,2.39621,2.40379,2.41121,2.41879,2.42621,2.43379,2.44121,2.44879,2.45621,2.46379,2.47121,2.47879,2.48621,2.49379,2.50121,2.50879,2.51621,2.52379,2.53121,2.53879,2.54621,2.55379,2.56121,2.56879,2.57621,2.58379,2.59121,2.59879,2.60621,2.61379,2.62121,2.62879,2.63621,2.64379,2.65121,2.65879,2.66621,2.67379,2.68121,2.68879,2.69621,2.70379,2.71121,2.71879,2.72621,2.73379,2.74121,2.74879,2.75621,2.76379,2.77121,2.77879,2.78621,2.79379,2.80121,2.80879,2.81621,2.82379,2.83121,2.83879,2.84621,2.85379,2.86121,2.86879,2.87621,2.88379,2.89121,2.89879,2.90621,2.91379,2.92121,2.92879,2.93621,2.94379,2.95121,2.95879,2.96621,2.97379,2.98121,2.98879,2.99621,3.00379,3.01121,3.01879,3.02621,3.03379,3.04121,3.04879,3.05621,3.06379,3.07121,3.07879,3.08621,3.09379,3.10121,3.10879,3.11621,3.12379,3.13121,3.13879,3.14621,3.15379,3.16121,3.16879,3.17621,3.18379,3.19121,3.19879,3.20621,3.21379,3.22121,3.22879,3.23621,3.24379,3.25121,3.25879,3.26621,3.27379,3.28121,3.28879,3.29621,3.30379,3.31121,3.31879,3.32621,3.33379,3.34121,3.34879,3.35621,3.36379,3.37121,3.37879,3.38621,3.39379,3.40121,3.40879,3.41621,3.42379,3.43121,3.43879,3.44621,3.45379,3.46121,3.46879,3.47621,3.48379,3.49121,3.49879,3.50621,3.51379,3.52121,3.52879,3.53621,3.54379,3.55121,3.55879,3.56621,3.57379,3.58121,3.58879,3.59621,3.60379,3.61121,3.61879,3.62621,3.63379,3.64121,3.64879,3.65621,3.66379,3.67121,3.67879,3.68621,3.69379,3.70121,3.70879,3.71621,3.72379,3.73121,3.73879,3.74621,3.75379,3.76121,3.76879,3.77621,3.78379,3.79121,3.79879,3.80621,3.81379,3.82121,3.82879,3.83621,3.84379,3.85121,3.85879,3.86621,3.87379,3.88121,3.88879,3.89621,3.90379,3.91121,3.91879,3.92621,3.93379,3.94121,3.94879,3.95621,3.96379,3.97121,3.97879,3.98621,3.99379,4.00121,4.00879,4.01621,4.02379,4.03121,4.03879,4.04621,4.05379,4.06121,4.06879,4.07621,4.08379,4.09121,4.09879,4.10621,4.11379,4.12121,4.12879,4.13621,4.14379,4.15121,4.15879,4.16621,4.17379,4.18121,4.18879,4.19621,4.20379,4.21121,4.21879,4.22621,4.23379,4.24121,4.24879,4.25621,4.26379,4.27121,4.27879,4.28621,4.29379,4.30121,4.30879,4.31621,4.32379,4.33121,4.33879,4.34621,4.35379,4.36121,4.36879,4.37621,4.38379,4.39121,4.39879,4.40621,4.41379,4.42121,4.42879,4.43621,4.44379,4.45121,4.45879,4.46621,4.47379,4.48121,4.48879,4.49621,4.50379,4.51121,4.51879,4.52621,4.53379,4.54121,4.54879,4.55621,4.56379,4.57121,4.57879,4.58621,4.59379,4.60121,4.60879,4.61621,4.62379,4.63121,4.63879,4.64621,4.65379,4.66121,4.66879,4.67621,4.68379,4.69121,4.69879,4.70621,4.71379,4.72121,4.72879,4.73621,4.74379,4.75121,4.75879,4.76621,4.77379,4.78121,4.78879,4.79621,4.80379,4.81121,4.81879,4.82621,4.83379,4.84121,4.84879,4.85621,4.86379,4.87121,4.87879,4.88621,4.89379,4.90121,4.90879,4.91621,4.92379,4.93121,4.93879,4.94621,4.95379,4.96121,4.96879,4.97621,4.98379,4.99121,4.99879,5.00621,5.01379,5.02121,5.02879,5.03621,5.04379,5.05121,5.05879,5.06621,5.07379,5.08121,5.08879,5.09621,5.10379,5.11121,5.11879,5.12621,5.13379,5.14121,5.14879,5.15621,5.16379,5.17121,5.17879,5.18621,5.19379,5.20121,5.20879,5.21621,5.22379,5.23121,5.23879,5.24621,5.25379,5.26121,5.26879,5.27621,5.28379,5.29121,5.29879,5.30621,5.31379,5.32121,5.32879,5.33621,5.34379,5.35121,5.35879,5.36621,5.37379,5.38121,5.38879,5.39621,5.40379,5.41121,5.41879,5.42621,5.43379,5.44121,5.44879,5.45621,5.46379,5.47121,5.47879,5.48621,5.49379,5.50121,5.50879,5.51621,5.52379,5.53121,5.53879,5.54621,5.55379,5.56121,5.56879,5.57621,5.58379,5.59121,5.59879,5.60621,5.61379,5.62121,5.62879,5.63621,5.64379,5.65121,5.65879,5.66621,5.67379,5.68121,5.68879,5.69621,5.70379,5.71121,5.71879,5.72621,5.73379,5.74121,5.74879,5.75621,5.76379,5.77121,5.77879,5.78621,5.79379,5.80121,5.80879,5.81621,5.82379,5.83121,5.83879,5.84621,5.85379,5.86121,5.86879,5.87621,5.88379,5.89121,5.89879,5.90621,5.91379,5.92121,5.92879,5.93621,5.94379,5.95121,5.95879,5.96621,5.97379,5.98121,5.98879,5.99621,6.00379,6.01121,6.01879,6.02621,6.03379,6.04121,6.04879,6.05621,6.06379,6.07121,6.07879,6.08621,6.09379,6.10121,6.10879,6.11621,6.12379,6.13121,6.13879,6.14621,6.15379,6.16121,6.16879,6.17621,6.18379,6.19121,6.19879,6.20621,6.21379,6.22121,6.22879,6.23621,6.24379,6.25121,6.25879,6.26621,6.27379,6.28121,6.28879,6.29621,6.30379,6.31121,6.31879,6.32621,6.33379,6.34121,6.34879,6.35621,6.36379,6.37121,6.37879,6.38621,6.39379,6.40121,6.40879,6.41621,6.42379,6.43121,6.43879,6.44621,6.45379,6.46121,6.46879,6.47621,6.48379,6.49121,6.49879,6.50621,6.51379,6.52121,6.52879,6.53621,6.54379,6.55121,6.55879,6.56621,6.57379,6.58121,6.58879,6.59621,6.60379,6.61121,6.61879,6.62621,6.63379,6.64121,6.64879,6.65621,6.66379,6.67121,6.67879,6.68621,6.69379,6.70121,6.70879,6.71621,6.72379,6.73121,6.73879,6.74621,6.75379,6.76121,6.76879,6.77621,6.78379,6.79121,6.79879,6.80621,6.81379,6.82121,6.82879,6.83621,6.84379,6.85121,6.85879,6.86621,6.87379,6.88121,6.88879,6.89621,6.90379,6.91121,6.91879,6.92621,6.93379,6.94121,6.94879,6.95621,6.96379,6.97121,6.97879,6.98621,6.99379,7.00121,7.00879,7.01621,7.02379,7.03121,7.03879,7.04621,7.05379,7.06121,7.06879,7.07621,7.08379,7.09121,7.09879,7.10621,7.11379,7.12121,7.12879,7.13621,7.14379,7.15121,7.15879,7.16621,7.17379,7.18121,7.18879,7.19621,7.20379,7.21121,7.21879,7.22621,7.23379,7.24121,7.24879,7.25621,7.26379,7.27121,7.27879,7.28621,7.29379,7.30121,7.30879,7.31621,7.32379,7.33121,7.33879,7.34621,7.35379,7.36121,7.36879,7.37621,7.38379,7.39121,7.39879,7.40621,7.41379,7.42121,7.42879,7.43621,7.44379,7.45121,7.45879,7.46621,7.47379,7.48121,7.48879,7.49621,7.50379,7.51121,7.51879,7.52621,7.53379,7.54121,7.54879,7.55621,7.56379,7.57121,7.57879,7.58621,7.59379,7.60121,7.60879,7.61621,7.62379,7.63121,7.63879,7.64621,7.65379,7.66121,7.66879,7.67621,7.68379,7.69121,7.69879,7.70621,7.71379,7.72121,7.72879,7.73621,7.74379,7.75121,7.75879,7.76621,7.77379,7.78121,7.78879,7.79621,7.80379,7.81121,7.81879,7.82621,7.83379,7.84121,7.84879,7.85621,7.86379,7.87121,7.87879,7.88621,7.89379,7.90121,7.90879,7.91621,7.92379,7.93121,7.93879,7.94621,7.95379,7.96121,7.96879,7.97621,7.98379,7.99121,7.99879,8.00621,8.01379,8.02121,8.02879,8.03621,8.04379,8.05121,8.05879,8.06621,8.07379,8.08121,8.08879,8.09621,8.10379,8.11121,8.11879,8.12621,8.13379,8.14121,8.14879,8.15621,8.16379,8.17121,8.17879,8.18621,8.19379,8.20121,8.20879,8.21621,8.22379,8.23121,8.23879,8.24621,8.25379,8.26121,8.26879,8.27621,8.28379,8.29121,8.29879,8.30621,8.31379,8.32121,8.32879,8.33621,8.34379,8.35121,8.35879,8.36621,8.37379,8.38121,8.38879,8.39621,8.40379,8.41121,8.41879,8.42621,8.43379,8.44121,8.44879,8.45621,8.46379,8.47121,8.47879,8.48621,8.49379,8.50121,8.50879,8.51621,8.52379,8.53121,8.53879,8.54621,8.55379,8.56121,8.56879,8.57621,8.58379,8.59121,8.59879,8.60621,8.61379,8.62121,8.62879,8.63621,8.64379,8.65121,8.65879,8.66621,8.67379,8.68121,8.68879,8.69621,8.70379,8.71121,8.71879,8.72621,8.73379,8.74121,8.74879,8.75621,8.76379,8.77121,8.77879,8.78621,8.79379,8.80121,8.80879,8.81621,8.82379,8.83121,8.83879,8.84621,8.85379,8.86121,8.86879,8.87621,8.88379,8.89121,8.89879,8.90621,8.91379,8.92121,8.92879,8.93621,8.94379,8.95121,8.95879,8.96621,8.97379,8.98121,8.98879,8.99621,9.00379,9.01121,9.01879,9.02621,9.03379,9.04121,9.04879,9.05621,9.06379,9.07121,9.07879,9.08621,9.09379,9.10121,9.10879,9.11621,9.12379,9.13121,9.13879,9.14621,9.15379,9.16121,9.16879,9.17621,9.18379,9.19121,9.19879,9.20621,9.21379,9.22121,9.22879,9.23621,9.24379,9.25121,9.25879,9.26621,9.27379,9.28121,9.28879,9.29621,9.30379,9.31121,9.31879,9.32621,9.33379,9.34121,9.34879,9.35621,9.36379,9.37121,9.37879,9.38621,9.39379,9.40121,9.40879,9.41621,9.42379,9.43121,9.43879,9.44621,9.45379,9.46121,9.46879,9.47621,9.48379,9.49121,9.49879,9.50621,9.51379,9.52121,9.52879,9.53621,9.54379,9.55121,9.55879,9.56621,9.57379,9.58121,9.58879,9.59621,9.60379,9.61121,9.61879,9.62621,9.63379,9.64121,9.64879,9.65621,9.66379,9.67121,9.67879,9.68621,9.69379,9.70121,9.70879,9.71621,9.72379,9.73121,9.73879,9.74621,9.75379,9.76121,9.76879,9.77621,9.78379,9.79121,9.79879,9.80621,9.81379,9.82121,9.82879,9.83621,9.84379,9.85121,9.85879,9.86621,9.87379,9.88121,9.88879,9.89621,9.90379,9.91121,9.91879,9.92621,9.93379,9.94121,9.94879,9.95621,9.96379,9.97121,9.97879,9.98621,9.99379,10.00121,10.00879,10.01621,10.02379,10.03121,10.03879,10.04621,10.05379,10.06121,10.06879,10.07621,10.08379,10.09121,10.09879,10.10621,10.11379,10.12121,10.12879,10.13621,10.14379,10.15121,10.15879,10.16621,10.17379,10.18121,10.18879,10.19621,10.20379,10.21121,10.21879,10.22621,10.23379,10.24121,10.24879,10.25621,10.26379,10.27121,10.27879,10.28621,10.29379,10.30121,10.30879,10.31621,10.32379,10.33121,10.33879,10.34621,10.35379,10.36121,10.36879,10.37621,10.38379,10.39121,10.39879,10.40621,10.41379,10.42121,10.42879,10.43621,10.44379,10.45121,10.45879,10.46621,10.47379,10.48121,10.48879,10.49621,10.50379,10.51121,10.51879,10.52621,10.53379,10.54121,10.54879,10.55621,10.56379,10.57121,10.57879,10.58621,10.59379,10.60121,10.60879,10.61621,10.62379,10.63121,10.63879,10.64621,10.65379,10.66121,10.66879,10.67621,10.68379,10.69121,10.69879,10.70621,10.71379,10.72121,10.72879,10.73621,10.74379,10.75121,10.75879,10.76621,10.77379,10.78121,10.78879,10.79621,10.80379,10.81121,10.81879,10.82621,10.83379,10.84121,10.84879,10.85621,10.86379,10.87121,10.87879,10.88621,10.89379,10.90121,10.90879,10.91621,10.92379,10.93121,10.93879,10.94621,10.95379,10.96121,10.96879,10.97621,10.98379,10.99121,10.99879,11.00621,11.01379,11.02121,11.02879,11.03621,11.04379,11.05121,11.05879,11.06621,11.07379,11.08121,11.08879,11.09621,11.10379,11.11121,11.11879,11.12621,11.13379,11.14121,11.14879,11.15621,11.16379,11.17121,11.17879,11.18621,11.19379,11.20121,11.20879,11.21621,11.22379,11.23121,11.23879,11.24621,11.25379,11.26121,11.26879,11.27621,11.28379,11.29121,11.29879,11.30621,11.31379,11.32121,11.32879,11.33621,11.34379,11.35121,11.35879,11.36621,11.37379,11.38121,11.38879,11.39621,11.40379,11.41121,11.41879,11.42621,11.43379,11.44121,11.44879,11.45621,11.46379,11.47121,11.47879,11.48621,11.49379,11.50121,11.50879,11.51621,11.52379,11.53121,11.53879,11.54621,11.55379,11.56121,11.56879,11.57621,11.58379,11.59121,11.59879,11.60621,11.61379,11.62121,11.62879,11.63621,11.64379,11.65121,11.65879,11.66621,11.67379,11.68121,11.68879,11.69621,11.70379,11.71121,11.71879,11.72621,11.73379,11.74121,11.74879,11.75621,11.76379,11.77121,11.77879,11.78621,11.79379,11.80121,11.80879,11.81621,11.82379,11.83121,11.83879,11.84621,11.85379,11.86121,11.86879,11.87621,11.88379,11.89121,11.89879,11.90621,11.91379,11.92121,11.92879,11.93621,11.94379,11.95121,11.95879,11.96621,11.97379,11.98121,11.98879,11.99621,12.00379,12.01121,12.01879,12.02621,12.03379,12.04121,12.04879,12.05621,12.06379,12.07121,12.07879,12.08621,12.09379,12.10121,12.10879,12.11621,12.12379,12.13121,12.13879,12.14621,12.15379,12.16121,12.16879,12.17621
```

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 .

invt(*Area,df*)

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

iPart()**iPart(*Number*)** $\Rightarrow$ *heltall***iPart(*List1*)** $\Rightarrow$ *liste***iPart(*Matrix1*)** $\Rightarrow$ *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*])** $\Rightarrow$ *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 98.

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

isPrime()

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 **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 produkt håndboken.

isPrime(5)	true
isPrime(6)	false

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

Define nextprim(<i>n</i>)=Func	Done
Loop	
$n+1 \rightarrow n$	
If isPrime(<i>n</i>)	
Return <i>n</i>	
EndLoop	
EndFunc	
nextprim(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 218.

<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	Done
Local <i>temp,i</i>	
$0 \rightarrow temp$	
$1 \rightarrow i$	
Lbl <i>top</i>	
$temp+i \rightarrow temp$	
If $i < 10$ Then	
$i+1 \rightarrow i$	
Goto <i>top</i>	
EndIf	
Return <i>temp</i>	
EndFunc	
<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*) $\Rightarrow$ uttrykk

$\text{lcm}(6,9)$	18
-------------------	----

lcm(*Liste1*, *Liste2*) $\Rightarrow$ liste

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

lcm(*Matrise1*, *Matrise2*) $\Rightarrow$ 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 desimalpunkttall er produktet av teller og nevner.

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

left() (venstre)

left(*kildeStreng*[, *Num*]) $\Rightarrow$ streng

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

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

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

left(*Liste1*[, *Num*]) $\Rightarrow$ liste

$\text{left}(\{1, 3, -2, 4\}, 3)$	$\{1, 3, -2\}$
-----------------------------------	----------------

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

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

left(*Sammenlikning*) $\Rightarrow$ Uttrykk

Returnerer venstre side av en ligning eller ulikhet.

libShortcut(*BibliotekNavnStreng*,
HurtigtastNavnStreng[,
BiblPrivMerke])⇒*liste av variabler*

Oppretter en variabelgruppe i den gjeldende oppgaven som inneholder referanser til alle objektene i det angitte bibliotekdokumentet *bibliotekNavnStreng*. Legger også gruppe-medlemmene til i Variabler-menyen. 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 40).

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 151.)

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 numerisk eller streng 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 218).

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 *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 151.)

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 numerisk eller streng 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 218).

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 (lineær regresjon)

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 151.)

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 218).

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-varibel	Beskrivelse
[stat.LowerPred, stat.UpperPred]	Prediksjonsintervall for én enkeltobservasjon
stat.MEPred	Prediksjonsintervallets feilmargin
stat.SEPred	Standardfeil for prediksjonen
stat. $\hat{y}$	$a + b \cdot X_{Verdi}$

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 151).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 218).

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

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

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

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

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

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

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

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

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.

$\Delta\text{list}()$ (liste)

Katalog > 

$\Delta\text{List}(\text{Liste1}) \Rightarrow \text{liste}$

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

Merk: Du kan sette inn denne funksjonen fra tastaturet ved å skrive **$\Delta\text{List}(\dots)$** .

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*.

$\text{list}\blacktriangleright\text{mat}()$

Katalog > 

$\text{list}\blacktriangleright\text{mat}(\text{Liste [, elementerRerRad]}) \Rightarrow \text{matrise}$

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

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 **$\text{list@}\blacktriangleright\text{mat}(\dots)$** .

$\ln()$

  **taster**

$\ln(\text{Verdi1}) \Rightarrow \text{verdi}$

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

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

Returnerer argumentets naturlige logaritme.

Hvis kompleks formatmodus er reell:

For en liste, returneres elementenes naturlige logaritmer.

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

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

$$\{1.09861+3.14159 \cdot i, 0.182322, 1.60944\}$$

I radian-vinkelmodus og rektangulært, kompleks format:

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

$$\begin{bmatrix} 1.83145+1.73485 \cdot i & 0.009193-1.49086 \\ 0.448761-0.725533 \cdot i & 1.06491+0.623491 \cdot i \\ -0.266891-2.08316 \cdot i & 1.12436+1.79018 \cdot i \end{bmatrix}$$

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

LnReg**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 151).

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 numerisk eller streng 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 218).

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

88
Alfabetisk oversikt

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

Lock*Var*.

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 169 og **getLockInfo()**, side 67.

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

log()

  **taster**

log(*Verdi1* [, *Verdi2*])⇒*verdi*

$$\log_{10} (2.) = 0.30103$$

log(*Liste1* [, *Verdi2*])⇒*liste*

$$\log_4 (2.) = 0.5$$

Returnerer grunntallet -*Verdi2* argumentets logaritme.

$$\log_3 (10) - \log_3 (5) = 0.63093$$

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

For en liste, returneres grunntall -*Verdi2* for elementenes logaritme.

Hvis kompleks formatmodus er reell:

$$\log_{10} (\{-3,1.2,5\})$$

"Error: Non-real calculation"

Hvis *Uttr2* utelates, brukes 10 som grunntall.

Hvis kompleks formatmodus er rektangulær:

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

log(*kvadratMatrise1* [, *Verdi2*])⇒*kvadratMatrise*

I radian-vinkelmodus og rektangulært, kompleks format:

Returnerer matrisens grunn- *Verdi2* logaritme av *kvadratMatrise1*. Dette er ikke det samme som å beregne grunntallet- *Verdi2* logaritme av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

Hvis grunntall-argumentet utelates, brukes 10 som grunntall.

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

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

Logistic

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 151).

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 numerisk eller streng 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 218).

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 <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>

LogisticD

katalog > 

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 151).

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 numerisk eller streng 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 218).

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
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 produkhåndboken.

```

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

```

Done

rollcount()	16
rollcount()	3

LU (= nedre/øvre)

Katalog > 

LU *Matrise*, *lMatrise*, *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 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:
 $5E-14 \cdot \max(\dim(Matrise)) \cdot \text{radNorm}(Matrise)$

Faktoreriserende 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 m1,lower,upper,perm	Done
lower	$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
upper	$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
perm	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

M

mat►list() (matrise til liste)

Katalog > 

mat►list(*Matrise*) $\Rightarrow$ *liste*

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

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

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

max()		Katalog > 
max (Verdi1, Verdi2)⇒Uttrykk	$\max(2.3, 1.4)$	2.3
max (Liste1, Liste2)⇒liste	$\max(\{1, 2\}, \{-4, 3\})$	$\{1, 3\}$
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 <i>liste</i> .		
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 <i>Matrise1</i> .		
Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 218.		
Merk: Se også min() .		

mean() (gjennomsnitt)		Katalog > 
mean (Liste[, frekvListe])⇒uttrykk	$\text{mean}(\{0.2, 0.1, -0.3, 0.4\})$	0.26
Returnerer gjennomsnittet av elementene i Liste.	$\text{mean}(\{1, 2, 3\}, \{3, 2, 1\})$	$\frac{5}{3}$
		3
Hvert frekvListe element teller antallet forekomster av det tilsvarende elementet i Liste.		
mean (Matrise1[, FrekvMatrise]) ⇒matrise	I rektangulært vektorformat:	
Returnerer en radvektor av gjennomsnittet for alle kolonnene i Matrise1.		
Hvert frekvMatrise element teller antallet forekomster av det tilsvarende elementet i Matrise1.		
Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 218.		

mean() (gjennomsnitt)Katalog > 

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

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

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

median()Katalog > **median**(*Liste*[, *frekvListe*]) $\Rightarrow$ uttrykk

$$\text{median}(\{0.2, 0.1, -0.3, 0.4\}) \quad 0.2$$

Returnerer medianen av elementene i *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

median(*MatriseI*[, *frekvMatrise*]) $\Rightarrow$ *matrise*

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

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 ignorerer. For mer informasjon om tomme elementer, se side 218.

MedMedkatalog > **MedMed** *X*, *Y* [, *Frekv*] [, *Kategori*, *Inkluder*]]

Beregner median-median-linjeny = $(m \cdot x + b)$ for listene X og Y med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 151.)

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 numerisk eller streng 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 218).

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)

Katalog > 

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(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*.

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

mid({9,8,7,6},3)	{7,6}
mid({9,8,7,6},2,2)	{8,7}
mid({9,8,7,6},1,2)	{9,8}
mid({9,8,7,6},1,0)	{ }

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

min() (minimum)

Katalog > 

min(Verdi1, Verdi2)⇒Uttrykk

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

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

min() (minimum)Katalog > **min**(*Matrise!*) $\Rightarrow$ *matrise*

Returnerer en radvektor som inneholder minimumselementet av hver kolonne i *Matrise1*.

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

Merk: Se også **max()**.

mirr()Katalog > **mirr**

(*finansRente*, *reinvestRente*, *CF0*, *CFListe* [, *CFFrekv*])

Finansiell funksjon som returnerer modifisert rente av en investering.

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

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 77.

mod()Katalog > **mod**(*Verdi1*, *Verdi2*) $\Rightarrow$ *Uttrykk*

$$\text{mod}(7, 0) \quad 7$$

mod(*Liste1*, *Liste2*) $\Rightarrow$ *liste*

$$\text{mod}(7, 3) \quad 1$$

mod(*Matrise1*, *Matrise2*) $\Rightarrow$ *matrise*

$$\text{mod}(-7, 3) \quad 2$$

$$\text{mod}(7, -3) \quad -2$$

$$\text{mod}(-7, -3) \quad -1$$

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

Returnerer det første argumentet modulo det andre argumentet slik som definert ved identitetene:

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

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

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.

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 129

mRow() (mRad)

mRow(*Verdi*, *Matrise1*, *Indeks*) $\Rightarrow$ *matrise*

Returnerer en kopi av *Matrise1* med hvert element i rad *Indeks* av *Matrise1* multiplisert med *Verdi*.

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

mRowAdd() (mRadAdd)

mRowAdd(*Verdi*, *Matrise1*, *Indeks1*, *Indeks2*) $\Rightarrow$ *matrise*

Returnerer en kopi av *Matrise1* med hvert element i rad *Indeks2* i *Matrise1* erstattet med:

$$\text{Verdi} \times \text{rad Indeks1} + \text{rad Indeks2}$$

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

MultReg

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 151).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 218).

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.ŷ List	$\hat{y}$ List = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualene fra regresjonen

MultRegIntervals

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 151).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 218).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.ŷ	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

Utdata-variabel	Beskrivelse
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

[katalog](#) > 

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 lages i *stat.resultater*-variabelen (side 151).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 218).

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

Utdata-variabel	Beskrivelse
stat.MSError	Feilens kvadratgjennomsnitt (gjennomsnittlig kvadratavvik)
stat.bList	{b0,b1,...} 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
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)

  -taster

BoolskUttr1 **nand** *BoolskUttr2* returnerer
Boolsk uttrykk

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* $\Rightarrow$ *heltall*

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

Sammenlikner to reelle heltall bit-for-bit med en nand-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.

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(Verdi1, Verdi2)⇒Uttrykk

$$nCr(z, 3)_{z=5} \quad 10$$

For heltall *Verdi1* og *Verdi2* der $Verdi2 \geq 0$, er **nCr()** antall kombinasjoner av *Verdi1* som tar *Verdi2* om gangen. (Dette er også kjent som en binomisk koeffisient.)

$$nCr(z, 3)_{z=6} \quad 20$$
nCr(Verdi, 0)⇒1
nCr(Verdi, negHeltall)⇒0
nCr(Verdi, posHeltall)⇒Verdi · (Verdi-1) ... (Verdi-posHeltall+1) / posHeltall!
nCr(Verdi, ikkeHeltall)⇒Uttrykk! / ((Verdi-ikkeHeltall)! · ikkeHeltall!)
nCr(Liste1, Liste2)⇒liste

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

Returnerer en liste over kombinasjoner basert på samsvarende elementpar i de to listene. Argumentene må ha samme listestørrelse.

nCr(Matrise1, Matrise2)⇒matrise

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

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*]) $\Rightarrow$ *verdi*

nDerivative(*Uttr1*, *Var*[,*Orden*]) |
Var=*Verdi* $\Rightarrow$ *verdi*

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.

Hvis variabelen *Var* ikke inneholder en numerisk verdi, må du oppgi en *Verdi*.

Den deriverte må være av orden 1 eller 2.

Merk: Den **nDerivative()** algoritmen har en begrensning: den arbeider rekursivt gjennom det ikke-forenklede uttrykket og beregner den numeriske verdien av den første deriverte (og eventuelt den andre) og behandlingen av hvert deluttrykk, som kan føre til et uventet resultat.

Studer eksemplet til høyre. Den første deriverte av $x \cdot (x^2 + x)^{1/3}$ i $x=0$ er lik 0. Men siden den første deriverte av deluttrykket $(x^2 + x)^{1/3}$ er udefinert i $x=0$ og denne verdien blir brukt til å beregne den deriverte av hele uttrykket, rapporterer **nDerivative()** resultatet som udefinert og viser en varselmelding.

Hvis du støter på denne begrensningen, verifiserer du løsningen grafisk. Du kan også prøve å bruke **centralDiff()**.

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

$\text{nDerivative}\left(x \cdot (x^2 + x)^{\frac{1}{3}}, x, 1\right) _{x=0}$	undef
$\text{centralDiff}\left(x \cdot (x^2 + x)^{\frac{1}{3}}, x\right) _{x=0}$	0.000033

newList() (nyListe)

Katalog > 

newList(*numElementer*) $\Rightarrow$ *liste*

Returnerer en liste med en dimensjon lik *numElementer*. Hvert element er null.

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

newMat() (nyMat)**Katalog** > **newMat**(*numRader*,
numKolonner) $\Rightarrow$ matrise

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

Returnerer en matrise med bare nuller med dimensjonen *numRader* og *numKolonner*.

nfMax()**Katalog** > **nfMax**(*Uttr1*, *Var*) $\Rightarrow$ verdi

nfMax $\left(-x^2-2\cdot x-1,x\right)$	-1.
---	-----

nfMax(*Uttr1*, *Var*, *nedGrense*) $\Rightarrow$ verdi

nfMax $\left(0.5\cdot x^3-x-2,x,-5,5\right)$	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

nfMin $\left(x^2+2\cdot x+5,x\right)$	-1.
--	-----

nfMin(*Uttr1*, *Var*, *nedGrense*) $\Rightarrow$ verdi

nfMin $\left(0.5\cdot x^3-x-2,x,-5,5\right)$	-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

Hvis integranden *Uttr1* ikke inneholder andre verdier enn *Var*, og hvis *Nedre* og *Øvre* er konstanter, positiv ∞ , eller negativ ∞ , så returnerer **nInt()** en tilnærmet av $\int$ (*Uttr1*, *Var*, *Nedre*, *Øvre*). Denne tilnærmede er et veiet gjennomsnitt av noen utvalgsverdier av integranden i intervallen *Nedre*<*Var*<*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 ("Tvil som nøyaktighet") når det ser ut til at målet ikke er nådd.

Nest **nInt()** å utføre multiplert numerisk integrasjon. Integrasjonsgrensene kan avhenge av integrasjonsvariabler utenfor dem.

$$\text{nInt}(\cos(x), x, -\pi, \pi + 1. \text{E-}12) \quad -1.04144 \text{E-}12$$

$$\text{nInt}\left(\text{nInt}\left(\frac{e^{-x \cdot y}}{\sqrt{x^2 - y^2}}, y, -x, x\right), x, 0, 1\right) \quad 3.30423$$

nom()

Katalog > 

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 46.

nor (verken ... eller)

  -taster

BoolskUttr1 **nor** *BoolskUttr2* returnerer
Boolsk uttrykk

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-bits 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*

$\text{norm}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	5.47723
--	---------

norm(*Vektor*) ⇒ *uttrykk*

$\text{norm}\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right)$	2.23607
---	---------

Returnerer Frobenius-normen.

$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	2.23607
--	---------

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).

For $P(X \leq \text{øvreGrense})$, sett *nedreGrense* = -9E999.

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 *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.

not (2 $\geq$ 3)	true
not 0hB0►Base16	0hFFFFFFFFFFFFFF4F
not not 2	2

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)	Katalog > 	
nPr (<i>Verdi1</i> , <i>Verdi2</i>) $\Rightarrow$ <i>Uttr1</i>	$nPr(z,3) z=5$	60
For heltall <i>Verdi1</i> og <i>Verdi2</i> der <i>Verdi1</i> $\geq$ <i>Verdi2</i> $\geq$ 0, er nPr() antall permutasjoner av <i>Verdi1</i> som tar <i>Verdi2</i> om gangen.	$nPr(z,3) z=6$	120
nPr (<i>Verdi</i> , 0) $\Rightarrow$ 1	$nPr(\{5,4,3\},\{2,4,2\})$	$\{20,24,6\}$
nPr (<i>Verdi</i> , <i>negHeltall</i>)	$nPr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$
$\Rightarrow 1/((Verdi+1) \cdot (Verdi+2)) \dots$ (<i>Verdi-negHeltall</i>)		
nPr (<i>Verdi</i> , <i>posHeltall</i>)		
$\Rightarrow Verdi \cdot (Verdi-1) \dots$ (<i>Verdi-posHeltall+1</i>)		
nPr (<i>Verdi</i> , <i>ikkeHeltall</i>)		
$\Rightarrow Verdi! / (Verdi-ikkeHeltall)!$		
nPr (<i>Liste1</i> , <i>Liste2</i>) $\Rightarrow$ <i>liste</i>	$nPr(\{5,4,3\},\{2,4,2\})$	$\{20,24,6\}$
Returnerer en liste over permutasjoner basert på samsvarende elementpar i de to listene. Argumentene må ha samme listestørrelse.		
nPr (<i>Matrise1</i> , <i>Matrise2</i>) $\Rightarrow$ <i>matrise</i>	$nPr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$
Returnerer en matrise av permutasjoner basert på tilsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.		

npv()	Katalog > 	
npv (<i>Rentefot</i> , <i>CFO</i> , <i>CFListe</i> [, <i>CFFrekv</i>])	$list1:=\{6000,-8000,2000,-3000\}$	$\{6000,-8000,2000,-3000\}$
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.	$list2:=\{2,2,2,1\}$	$\{2,2,2,1\}$
<i>Rentefot</i> er den renten som trekkes fra kontantstrømmene (pengekostnadene) over en periode.	$npv(10,5000,list1,list2)$	4769.91
<i>CFO</i> er kontantstrømmen ved start kl. 0; den må være et reelt tall.		

CFListe er en liste med kontantstrømbeløp etter den innledende kontantstrømmen *CF0*.

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)

nSolve(*Ligning*, *Var*[=*Forslag*]) $\Rightarrow$ *tall* eller *feil_streng*

nSolve(*Ligning*, *Var*[=*Forslag*], *nedGrense*) $\Rightarrow$ *tall* eller *feil_streng*

nSolve(*Ligning*, *Var* [=*Forslag*], *nedGrense*, *øvGrense*) $\Rightarrow$ *tall* eller *feil_streng*

nSolve(*Ligning*, *Var*[=*Forslag*]) | *nedGrense* $\leq$ *Var* $\leq$ *øvGrense* $\Rightarrow$ *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."

$\text{nSolve}(x^2 + 5 \cdot x - 25 = 9, x)$	3.84429
$\text{nSolve}(x^2 = 4, x = -1)$	-2.
$\text{nSolve}(x^2 = 4, x = 1)$	2.

Merk: Hvis det foreligger flere løsninger, kan du bruke et forslag som hjelp for å finne en spesiell 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"

OneVar (EnVar)Katalog > 

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 151).

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 218.

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

Utdata-varibel	Beskrivelse
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)

Katalog >

BoolskUtr1 **or** *BoolskUtr2* returnerer *Boolsk uttrykk*

BoolskListe1 **or** *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrise1 **or** *BoolskMatrise2* returnerer *Boolsk matrise*

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 flerinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

Heltall1 **or** *Heltall2* **or** *heltall*

Define $g(x)$	=Func	Done
	If $x \leq 0$ or $x \geq 5$	
	Goto end	
	Return $x \cdot 3$	
	Lbl end	
	EndFunc	
$g(3)$		9
$g(0)$		A function did not return a value

I heksades grunntall-modus:

0h7AC36 or 0h3D5F	0h7BD7F
-------------------	---------

Viktig: Null, ikke bokstaven O.

I binær grunntall-modus:

Sammenlikner to reelle heltall bit-for-bit med en or-handling. Internt er begge heltallene omregnet til 64-bits 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**.

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)

ord(*Streng*) $\Rightarrow$ *heltall*

ord("hello") 104

ord(*Liste1*) $\Rightarrow$ *liste*

char{104} "h "

Returnerer numerisk kode for de første tegnene i tegnstreng *Streng*, eller en liste over de første tegnene i hvert listeelement.

ord(char{24}) 24

ord({"alpha", "beta"}) { 97, 98 }

P

P►Rx()

P►Rx(*rUtt*, θ *Utt*) $\Rightarrow$ *uttrykk*

I Radian-vinkelmodus:

P►Rx(*rListe*, θ *Liste*) $\Rightarrow$ *liste*

P►Rx(4,60°) 2.

P►Rx(*rMatrise*, θ *Matrise*) $\Rightarrow$ *matrise*

$$P \blacktriangleright Rx \left\{ \left\{ -3, 10, 1.3 \right\}, \left\{ \frac{\pi}{3}, \frac{\pi}{4}, 0 \right\} \right\}$$

$$\{ -1.5, 7.07107, 1.3 \}$$

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(*rVerdi*, θ *Verdi*) $\Rightarrow$ *verdi*

I Radian-vinkelmodus:

P►Ry(*rListe*, θ *Liste*) $\Rightarrow$ *liste*

P►Ry(4,60°) 3.4641

P►Ry(*rMatrise*, θ *Matrise*) $\Rightarrow$ *matrise*

P►Ry $\left\{\left\{-3,10,1.3\right\},\left\{\frac{\pi}{3},\frac{\pi}{4},0\right\}\right\}$
 $\{-2.59808,-7.07107,0\}$

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 163.

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 163.

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* [, ...]]]])

Returnerer definisjoner for en stykkevis definert funksjon i form av en liste. Du kan også opprette stykkevis definisjoner med en sjablon.

Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$	Done
$p(1)$	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 *nedreGrense*=0

poissPdf()

poissPdf(λ , *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 $[1 \ 3.] \blacktriangleright \text{Polar}$ $[3.16228 \ \angle 71.5651]$

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 126.

kompleksVerdi ► Polar

Viser *kompleksVektor* i polar form.

- Grader-vinkelmodus returnerer $(r \angle \theta)$.
- Radian-vinkelmodus returnerer $re^{i\theta}$.

kompleksVerdi kan ha en 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:

$(3+4 \cdot i) \blacktriangleright \text{Polar}$	$e^{0.927295 \cdot i} \cdot 5$
$\left(\left(4 \angle \frac{\pi}{3} \right) \right) \blacktriangleright \text{Polar}$	$e^{1.0472 \cdot i} \cdot 4$

I Gradian-vinkelmodus:

$(4 \cdot i) \blacktriangleright \text{Polar}$	$(4 \angle 100.)$
--	-------------------

I Grader-vinkelmodus:

$(3+4 \cdot i) \blacktriangleright \text{Polar}$	$(5 \angle 53.1301)$
--	----------------------

polyEval()**polyEval(Liste1, Uttr1) ⇒ uttrykk**

$\text{polyEval}(\{1, 2, 3, 4\}, 2)$	26
--------------------------------------	----

polyEval(Liste1, Liste2) ⇒ uttrykk

$\text{polyEval}(\{1, 2, 3, 4\}, \{2, -7\})$	$\{26, -262\}$
--	----------------

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: { }.

Poly må være et polynom i utvidet form i én variabel. Ikke bruk utvidede former, som f.eks. $y^2 \cdot y + 1$ or $x \cdot x + 2 \cdot x + 1$

Den andre syntaksen, **polyRoots**(*KoeffListe*), returnerer en liste over sanne røtter for koeffisientene i *KoeffListe*.

Merk: Se også **cPolyRoots()**, side 32.

$\text{polyRoots}(y^3+1, y)$	$\{-1\}$
$\text{cPolyRoots}(y^3+1, y)$	$\{-1, 0.5-0.866025i, 0.5+0.866025i\}$
$\text{polyRoots}(x^2+2 \cdot x+1, x)$	$\{-1, -1\}$
$\text{polyRoots}(\{1, 2, 1\})$	$\{-1, -1\}$

PowerReg (PotensReg)

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 151).

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 numerisk eller streng 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 218).

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 <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>

Prgm

Katalog >

Prgm

Blokke

EndPrgm

Sjablon for å opprette et egendefinert program. Må brukes med kommandoen **Define**, **Define LibPub** eller **Define LibPriv**.

Blokke kan være ett enkelt utsagn, en rekke utsagn adskilt med “:”-tegnet eller en rekke med utsagn på separate linjer.

Blokke 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.

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

```
proggcd(4560,450)
```

450 60

60 30

30 0

GCD=30

Done

Product()

Katalog > **product(Liste[, Start[, slutt]])** $\Rightarrow$ uttrykk

Returnerer produktet av elementene i *Liste*. *Start* og *Slutt* er valgfrie. De spesifiserer et elementområde.

$\text{product}(\{1,2,3,4\})$	24
$\text{product}(\{4,5,8,9\},2,3)$	40

product(MatriseI[, Start[, slutt]]) $\Rightarrow$ matrise

Returnerer en radvektor som inneholder produktet av elementene i kolonnene i *MatriseI*. *Start* og *slutt* er alternativer. De spesifiserer et radområde.

$\text{product}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	$\begin{bmatrix} 28 & 80 & 162 \end{bmatrix}$
$\text{product}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},1,2\right)$	$\begin{bmatrix} 4 & 10 & 18 \end{bmatrix}$

Tomme (åpne) elementer ignorerer. For mer informasjon om tomme elementer, se side 218.

propFrac() (ekteBrøk)

Katalog > **propFrac(VerdiI[, Var])** $\Rightarrow$ verdi

propFrac(rasjonal_tall) returnerer *rasjonalt_tall* som summen av et heltall og en brøk som har samme fortegn og større nevner enn teller.

$\text{propFrac}\left(\frac{4}{3}\right)$	$1+\frac{1}{3}$
$\text{propFrac}\left(-\frac{4}{3}\right)$	$-1-\frac{1}{3}$

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.

QR

Katalog > **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 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:
 $5E-14 \cdot \max(\dim(\text{Matrise})) \cdot \text{radNorm}(\text{Matrise})$

Faktoriseringen 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*.

Det flytende desimalpunktallet (9.) i m1 gjør at resultatene må beregnes i flytende desimalpunkt-form.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix} \rightarrow m1 \qquad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix}$$

QR m1,qm,rm		Done		
qm	0.123091	0.904534	0.408248	
	0.492366	0.301511	-0.816497	
	0.86164	-0.301511	0.408248	
rm	8.12404	9.60114	11.0782	
	0.	0.904534	1.80907	
	0.	0.	0.	

QuadReg (KvadReg)

katalog > **QuadReg** *X*, *Y* [, *Frekv*] [, *Kategori*, *Inkluder*]]

Finner 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 151.)

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 numerisk eller streng 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 218).

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 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>

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 151.)

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 numerisk eller streng 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 218).

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 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>

R►Pθ()**Katalog >** 

R►Pθ (*xValue*, *yValue*) ⇒ *verdi*
R►Pθ (*xList*, *yList*) ⇒ *liste*
R►Pθ (*xMatrix*, *yMatrix*) ⇒ *matrise*

Returnerer tilsvarende θ-koordinat for (x,y) par-argumentene.

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 Grader-vinkelmodus:

R►Pθ(2,2)	45.
-----------	-----

I Gradian-vinkelmodus:

R►Pθ(2,2)	50.
-----------	-----

I Radian-vinkelmodus:

R►Pθ(3,2)	0.588003
R►Pθ $\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right)$	$\begin{bmatrix} 0. & 2.94771 & 0.643501 \end{bmatrix}$

R►Pr()**Katalog >** 

R►Pr (*xValue*, *yValue*) ⇒ *verdi*
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 (...)**.

I Radian-vinkelmodus:

R►Pr(3,2)	3.60555
R►Pr $\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right)$	$\begin{bmatrix} 3 & 4.07638 & \frac{5}{2} \end{bmatrix}$

►Rad**Katalog >** 

Value►Rad ⇒ *verdi*

Omformer argumentet til radian vinkelmåling.

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

I Grader-vinkelmodus:

(1.5)►Rad	(0.02618) ^r
-----------	------------------------

I Gradian-vinkelmodus:

(1.5)►Rad	(0.023562) ^r
-----------	-------------------------

tilf()Katalog > **rand()** ⇒ *uttrykk*

Angir det tilfeldige tallet for start.

rand(#Trials) ⇒ *liste***rand()** returnerer en tilfeldig verdi mellom 0 og 1.**rand(#Trials)** returnerer en liste som inneholder *#Trials* tilfeldige verdier mellom 0 og 1.

RandSeed 1147	Done
rand(2)	{ 0.158206, 0.717917 }

tilfBin()Katalog > **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.

randBin(80,0.5)	46.
randBin(80,0.5,3)	{ 43., 39., 41. }

tilfInt()Katalog > **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.

randInt(3,10)	3.
randInt(3,10,4)	{ 9., 3., 4., 7. }

randMat()

Katalog > 

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)	$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$

Merk: Verdien i denne matrisen endres hver gang du trykker på **enter**.

randNorm()

Katalog > 

randNorm(μ , σ) ⇒ *uttrykk*
randNorm(μ , σ , #Trials) ⇒ *liste*

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.

RandSeed 1147	Done
randNorm(0,1)	0.492541
randNorm(3,4.5)	-3.54356

randPoly()

Katalog > 

randPoly(Var, Order) ⇒ *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]) ⇒ *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.

Define list3={1,2,3,4,5}	Done
Define list4=randSamp(list3,6)	Done
list4	{1.,3.,3.,1.,3.,1.}

RandSeed

Katalog > 

RandSeed Tall

Hvis *Tall* = 0, settes startverdien for tilfeldig tall-generatoren til fabrikkinnstilling. Hvis *Tall* ≠ 0, brukes det for å opprette to startverdier, som lagres i systemvariablene seed1 og seed2.

RandSeed 1147	Done
rand()	0.158206

real()

Katalog > 

real(Value1) ⇒ verdi

$\text{real}(2+3 \cdot i)$	2
----------------------------	---

Returnerer den reelle delen av argumentet.

real(List1) ⇒ liste

$\text{real}(\{1+3 \cdot i, 3, i\})$	$\{1, 3, 0\}$
--------------------------------------	---------------

Returnerer den reelle delen av alle elementer.

real(Matrix1) ⇒ matrise

$\text{real}\left(\begin{bmatrix} 1+3 \cdot i & 3 \\ 2 & i \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix}$
--	--

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 116.

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.

$\left(\begin{bmatrix} 3 & \angle \frac{\pi}{4} & \angle \frac{\pi}{6} \end{bmatrix}\right) \text{► Rect}$	$[1.06066 \quad 1.06066 \quad 2.59808]$
--	---

I Radian-vinkelmodus:

$\left(\begin{bmatrix} \frac{\pi}{4} \\ 4 \cdot e^{\frac{\pi}{3}} \end{bmatrix}\right) \text{► Rect}$	11.3986
$\left(\begin{bmatrix} 4 & \angle \frac{\pi}{3} \end{bmatrix}\right) \text{► Rect}$	$2.+3.4641 \cdot i$

Merk: Du må bruke parentes for å legge inn polar ($r \angle \theta$).

I Gradian-vinkelmodus:

$$\left((1 \angle 100) \right) \blacktriangleright \text{Rect} \quad i$$

I Grader-vinkelmodus:

$$\left((4 \angle 60) \right) \blacktriangleright \text{Rect} \quad 2.+3.4641 \cdot i$$

Merk: For å skrive $\angle$, velg den fra symbollisten i Katalogen.

ref()

ref(MatrixI[, Tol]) $\Rightarrow$ *matrise*

Returnerer eliminasjonsformen av *MatrixI*.

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** til Tilnærmet, utføres beregningene med flyttallsregning.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
 $5E-14 \cdot \max(\dim(\text{MatrixI})) \cdot \text{rowNorm}(\text{MatrixI})$

Unngå udefinerte elementer i *MatrixI*. De kan føre til uventede resultater.

Hvis for eksempel *a* er udefinert i følgende uttrykk, vises en varselmelding, og resultatet vises som:

$$\text{ref} \left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \quad \begin{bmatrix} 1 & \frac{1}{a} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{ref} \left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix} \right) \quad \begin{bmatrix} 1 & \frac{-2}{5} & \frac{-4}{5} & \frac{4}{5} \\ 0 & 1 & \frac{4}{7} & \frac{11}{7} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

Varselet 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.

$$\text{ref} \left(\begin{bmatrix} a & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \middle| a=0 \right) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Merk: Se også **rref()**, page 136.

RefreshProbeVars

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()

remain(*Value1*, *Value2*) ⇒ *verdi*
remain(*List1*, *List2*) ⇒ *liste*
remain(*Matrix1*, *Matrix2*) ⇒ *matrise*

Returnerer resten av det første argumentet med hensyn på det andre argumentet som definert av identitetene:

remain(*x*,0) *x*
remain(*x*,*y*) $x - y \cdot \text{iPart}(x/y)$

Som en konsekvens, merk at **remain**(-*x*,*y*) – **remain**(*x*,*y*). Resultatet er enten null eller det har samme fortegn som det første argumentet.

Merk: Se også **mod()**, side 98.

remain (7,0)	7
remain (7,3)	1
remain (-7,3)	-1
remain (7,-3)	1
remain (-7,-3)	-1
remain ($\{12, -14, 16\}, \{9, 7, -5\}$)	$\{3, 0, 1\}$

remain ($\begin{pmatrix} 9 & -7 \\ 6 & 4 \end{pmatrix}, \begin{pmatrix} 4 & 3 \\ 4 & -3 \end{pmatrix}$)	$\begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix}$
--	---

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*.

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()

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.

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:

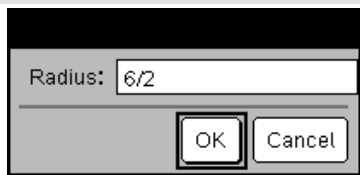
Definer *func(arg1, ...argn)* = *brukerens svar*

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



Resultat etter trykk på **OK**:

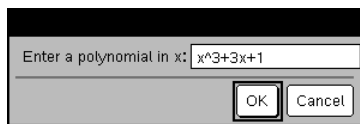
Stråle: 6/2
Area = 28,2743

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: $\{-0,322185\}$

- **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 131.

ForespørStr

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 129.

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 [*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 flerinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkt håndboken.

```
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*

Returnerer *Num*-elementene som ligger lengst til høyre i *List1*.

Hvis du utelater *Num*, returneres alle elementer i *List1*.

right(*sourceString* [, *Num*]) ⇒ *streng*

Returnerer *Num*-tegnene som ligger lengst til høyre i tegnstreng *sourceString*.

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

right(*Comparison*) ⇒ *uttrykk*

Returnerer høyre side av en ligning eller ulikhet.

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

right("Hello", 2) "lo"

rk23 ()**rk23**(*Expr*, *Var*, *depVar*, {*Var0*, *VarMax*}, *depVar0*, *VarStep* [, *dif_tol*]) ⇒ *matrise***rk23**(*SystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep* [, *dif_tol*]) ⇒ *matrise***rk23**(*ListOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep* [, *dif_tol*]) ⇒ *matrise*

Differensialligning:

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

rk23 (0.001 · y · { 100 - y }, t, y, { 0, 100 }, 10, 1)					
	0.	1.	2.	3.	4.
	10.	10.9367	11.9493	13.042	14.2

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

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.

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

SystemOfExpr er et system på høyre side som definerer systemet av ODE-er (tilsvarer rekkefølgen av avhengige variabler i ListOfDepVars).

ListOfExpr er en liste over høyresider som definerer systemet av ODE-er (tilsvarer rekkefølgen av avhengige variabler i ListOfDepVars).

Var er den uavhengige variabelen.

ListOfDepVars er en liste over avhengige variabler.

$\{\text{Var}0, \text{VarMax}\}$ er en liste med to elementer som forteller funksjonen at den skal integrere fra $\text{Var}0$ til VarMax .

$\text{ListOfDepVars}0$ er en liste over opprinnelige verdier for avhengige variabler.

Hvis VarStep beregnes til et tall som ikke er null: $\text{tegn}(\text{VarStep}) = \text{tegn}(\text{VarMax} - \text{Var}0)$ og løsninger returneres ved $\text{Var}0 + i * \text{VarStep}$ for alle $i = 0, 1, 2, \dots$ slik at $\text{Var}0 + i * \text{VarStep}$ er i $[\text{var}0, \text{VarMax}]$ (får kanskje ikke en løsningsverdi ved VarMax).

Hvis VarStep beregnes til null, returneres løsningene ved "Runge-Kutta" Var -verdiene.

Samme ligning med *difto* / satt til $1.E-6$

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

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 $y2(0) = 5$

rk23($\left\{ \begin{cases} -y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1 \right\}$)				
0.	1.	2.	3.	4.
2.	1.94103	4.78694	3.25253	1.82848
5.	16.8311	12.3133	3.51112	6.27245

diftol er feiltoleransen (grunnverdi på 0,001).

root()

root(*Value*) $\Rightarrow$ *rot*

root(*Value1*, *Value2*) $\Rightarrow$ *rot*

root(*Value*) returnerer kvadratroten av *Value*.

root(*Value1*, *Value2*) returnerer *Value2*-roten av *Value1*. *Value1* kan være en reell eller kompleks flytende desimalpunkt-konstant, et heltall eller en kompleks rasjonal konstant.

Merk: Se også **mal for N.-rot**, side 1.

$\sqrt[3]{8}$	2
$\sqrt[3]{3}$	1.44225

rotate()

rotate(*Integer1* [, *#ofRotations*]) $\Rightarrow$ *heltall*

I binær grunntall-modus:

Roterer bitene i et binært heltall. Du kan angi *Integer1* i enhver basis. Den konverteres automatisk til en 64-biters binærform med fortegn. Hvis *Integer1* er for stort for denne formen, brukes en symmetrisk modulusoperasjon til å konvertere tallet inn i gyldig verdiområde. (For mer informasjon, se **►Base2**, side 16.

Hvis *#ofRotations* er positiv, skjer rotasjonen til venstre. Hvis *#ofRotations* er negativ, skjer rotasjonen til høyre. Grunninnstilling er -1 (roteres én bit mot høyre).

For eksempel i en høyre-rotasjon:

Hver bit roterer mot høyre.

0b000000000000001111010110000110101

Bit helt til høyre roterer mot venstre.

produserer:

0b100000000000000111101011000011010

rotate (0b11111111111111111111111111111111)	
0b100000000000000000000000000000000001	►
rotate (256,1)	0b1000000000

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

I heksadesimal grunntall-modus:

rotate (0h78E)	0h3C7
rotate (0h78E, -2)	0h8000000000000001E3
rotate (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).

Resultatene vises i forhold til grunntall-modusen.

rotate(ListI[,#ofRotations]) $\Rightarrow$ *liste*

Returnerer en kopi av *ListI* som er rotert mot høyre eller mot venstre av *#ofRotations*-elementer. Endrer ikke *ListI*.

Hvis *#ofRotations* er positiv, skjer rotasjonen til venstre. Hvis *#ofRotations* er negativ, skjer rotasjonen til høyre. Grunninnstilling er -1 (roterer ett element mot høyre).

rotate(StringI[,#ofRotations]) $\Rightarrow$ *streng*

Returnerer en kopi av *StringI* som er rotert mot høyre eller mot venstre av *#ofRotations*-tegnene. Endrer ikke *StringI*.

Hvis *#ofRotations* er positiv, skjer rotasjonen til venstre. Hvis *#ofRotations* er negativ, skjer rotasjonen til høyre. Grunninnstilling er -1 (roterer ett tegn mot høyre).

I desimalgrunntall-modus:

$\text{rotate}(\{1,2,3,4\})$	$\{4,1,2,3\}$
$\text{rotate}(\{1,2,3,4\},2)$	$\{3,4,1,2\}$
$\text{rotate}(\{1,2,3,4\},1)$	$\{2,3,4,1\}$

$\text{rotate}(\text{"abcd"})$	"dabc"
$\text{rotate}(\text{"abcd"},2)$	"cdab"
$\text{rotate}(\text{"abcd"},1)$	"bcda"

round()

round(ValueI[,digits]) $\Rightarrow$ *verdi*

Returnerer argumentet avrundet til spesifisert antall sifre etter desimalpunktet.

digits må være et heltall i området $1-12$. Hvis *digits* ikke er inkludert, returneres argumentet avrundet til 12 signifikante siffer.

Merk: Visning av siffermodus kan ha innvirkning på hvordan dette vises.

round(ListI[,digits]) $\Rightarrow$ *liste*

Returnerer en liste over elementer avrundet til spesifisert antall sifre.

round(MatrixI[,digits]) $\Rightarrow$ *matrise*

Returnerer en matrise over elementer avrundet til spesifisert antall sifre.

$\text{round}(1.234567,3)$	1.235
----------------------------	-------

$\text{round}(\{\pi, \sqrt{2}, \ln(2)\},4)$	$\{3.1416, 1.4142, 0.6931\}$
---	------------------------------

$\text{round}\left(\begin{bmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{bmatrix},1\right)$	$\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$
---	--

rowAdd()Katalog > 

rowAdd(*Matrix1*, *rIndex1*, *rIndex2*) ⇒ *matrise*

Returnerer en kopi av *Matrix1* med rad *rIndex2* erstattet med summen av rader *rIndex1* og *rIndex2*.

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

rowDim()Katalog > 

rowDim(*Matrix*) ⇒ *uttrykk*

Returnerer antallet rader i *Matrix*.

Merk: Se også **colDim()**, side 24.

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

$$\text{rowDim}(m1) \quad 3$$

rowNorm()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.

$$\text{rowNorm}\left(\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix}\right) \quad 25$$

rowSwap()Katalog > 

rowSwap(*Matrix1*, *rIndex1*, *rIndex2*) ⇒ *matrise*

Returnerer *Matrix1* med rader *rIndex1* og *rIndex2* ombyttet.

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

$$\text{rowSwap}(mat, 1, 3) \quad \begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$$

rref()Katalog > 

rref(*Matrix1*[, *Tol*]) ⇒ *matrise*

Returnerer den reduserte eliminasjonsformen av *Matrix1*.

$$\text{rref}\left(\begin{bmatrix} -2 & -2 & 0 & -6 \\ 1 & -1 & 9 & -9 \\ -5 & 2 & 4 & -4 \end{bmatrix}\right) \quad \begin{bmatrix} 1 & 0 & 0 & \frac{66}{71} \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & \frac{-62}{71} \end{bmatrix}$$

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   eller stiller modusen **Auto eller Tilnærmet** til Tilnærmet, utføres beregningene med flyttallsregning.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
 $5E-14 \cdot \max(\dim(\text{Matrix } I)) \cdot \text{rowNorm}(\text{Matrix } I)$

Merk: Se også **ref()**, page 127.

S

sec()



sec(*Verdi I*) $\Rightarrow$ *verdi*

I Grader-vinkelmodus:

sec(*Liste I*) $\Rightarrow$ *liste*

$\sec(45)$	1.41421
$\sec(\{1, 2, 3, 4\})$	$\{1.00015, 1.00081, 1.00244\}$

Returnerer sekans til *Verdi I* eller returnerer en liste med sekans til hvert element i *Liste I*.

Merk: Argumentet tolkes som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus-innstilling. Du kan bruke $^{\circ}$, G eller r for å hoppe over vinkelmodusen midlertidig.

sec⁻¹()



sec⁻¹(*Verdi I*) $\Rightarrow$ *verdi*

I Grader-vinkelmodus:

sec⁻¹(*Liste I*) $\Rightarrow$ *liste*

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

Returnerer vinkelen som har sekans lik *Verdi I* eller returnerer en liste med inverse sekanser til hvert element i *Liste I*.

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

$\sec^{-1}()$

 **tast**

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

I Gradian-vinkelmodus:

$$\sec^{-1}(\sqrt{2}) \quad 50$$

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

I Radian-vinkelmodus:

$$\sec^{-1}(\{1,2,5\}) \quad \{0,1.0472,1.36944\}$$

sech()

Katalog > 

sech(Verdi1) $\Rightarrow$ verdi

$$\text{sech}(3) \quad 0.099328$$

sech(Liste1) $\Rightarrow$ liste

$$\text{sech}(\{1,2,3,4\}) \\ \{0.648054, 0.198522, 0.036619\}$$

Returnerer hyperbolsk sekans av *Verdi1* eller returnerer en liste med hyperbolske sekanser av *Liste1*-elementene.

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

Katalog > 

sech⁻¹(Verdi1) $\Rightarrow$ verdi

I Radian-vinkelmodus og Rectangular-kompleksmodus:

$$\text{sech}^{-1}(1) \quad 0$$

sech⁻¹(Liste1) $\Rightarrow$ liste

$$\text{sech}^{-1}(\{1, -2, 2, 1\}) \\ \{0, 2.0944 \cdot i, 8 \cdot 10^{-15} + 1.07448 \cdot i\}$$

Returnerer invers hyperbolsk sekans til *Verdi1* eller returnerer en liste med inverse hyperbolske sekanser til hvert element i *Liste1*.

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.

Send "SET COLOR.BLUE ON TIME .5"

Done

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 61), **GetStr** (side 68), og **eval()** (side 50).

Eksempel: Etterspør nåværende verdi fra hubbens innebygde lysnivåsensor. En **Get**-kommando henter verdien og tildeler den til variabelen *lysver*.

Send "READ BRIGHTNESS"	Done
Get <i>lightval</i>	Done
<i>lightval</i>	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($m \cdot n$)"	Done
<i>iostr.SendAns</i>	"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.

$\text{seq}\left(n^2, n, 1, 6\right)$	$\{1, 4, 9, 16, 25, 36\}$
$\text{seq}\left(\frac{1}{n}, n, 1, 10, 2\right)$	$\left\{1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \frac{1}{9}\right\}$
$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	$\frac{1968329}{1270080}$

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 .

$\text{sum}\left(\text{seq}\left(\frac{1}{n^2}, n, 1, 10, 1\right)\right)$	1.54977
--	---------

seqGen(*Uttr*, *Var*, *avhVar*, {*Var0*, *VarMaks*}[, *ListeAvInnlLedd* [, *VarIntervall* [, *Loftrverdi*]]]) $\Rightarrow$ *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* [, *Loftrverdi*]]]) $\Rightarrow$ *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:

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u_2(n-1)}{2} + u_1(n-1)\right\}, n, \{u_1, u_2\}, \{1, 5\}, \left[\begin{array}{c} - \\ 2 \end{array}\right]\right) \\ \left[\begin{array}{ccccc} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{array}\right]$$

Merk: Tomrommet () i matrisen med innledende ledd over brukes for å angi at det innledende leddet for $u_1(n)$ er beregnet ved hjelp av den eksplisitte tallfølgeformelen $u_1(n)=1/n$.

seqn(*Uttr*(*u*, *n* [, *ListeMedInnlLedd*[, *nMaks* [, *Loftrverdi*]]]) $\Rightarrow$ *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 $\text{Uttr}(u, n)$ og *ListeMedInnlLedd*, og returnerer resultatene som en liste.

seqn(*uttr*(*n* [, *nMaks* [, *Loftrverdi*]])) $\Rightarrow$ *liste*

Genererer de første 6 leddene i tallfølgen $u(n) = u(n-1)/2$, med $u(1)=2$.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right) \\ \left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

Genererer en liste over ledd for en ikke-rekursiv tallfølge $u(n)=Utr(n)$ som følger: Øker n fra 1 til $nMaks$ med 1, beregner $u(n)$ for tilsvarende verdier av n ved hjelp av formelen $Utr(n)$, og returnerer resultatene som en liste.

$$\text{seqn}\left(\frac{1}{n^2}, 6\right) \quad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

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) $\Rightarrow$ heltall

SetMode(liste) $\Rightarrow$ 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.

Define <i>prog1()</i> =Prgm	Done
Disp π	
setMode(1,16)	
Disp π	
EndPrgm	
<i>prog1()</i>	
	3.14159
	3.14
	Done

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 67.

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	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
Ekspensielt 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:

0b000000000000000111101011000011010

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 "undef".

shift(*StrengI* [, #*avSkift*]) $\Rightarrow$ *streng*

Returnerer en kopi av *StrengI* skiftet mot høyre eller mot venstre av #*ofShifts*-tegn. Endrer ikke *StrengI*.

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(*VerdiI*) $\Rightarrow$ *verdi*

sign(*ListeI*) $\Rightarrow$ *list e*

sign(*MatriseI*) $\Rightarrow$ *matrise*

For reell og komplekst *VerdiI*, returnerer *VerdiI* / **abs**(*VerdiI*) når *VerdiI* $\neq$ 0.

Returnerer 1 hvis *VerdiI* er positiv.

Returnerer -1 hvis *VerdiI* er negativ.

sign(0) returnerer ± 1 hvis kompleks formatmodus er Reell; ellers returnerer den seg selv.

sign(0) representerer enhetssirkelen i den komplekse grunnmengden.

For en liste eller matrise returneres fortegnene for alle elementene.

sign(-3.2)	-1
sign({2,3,4,-5})	{1,1,1,-1}

Hvis kompleks formatmodus er reell:

sign([-3 0 3])	[-1 undef 1]
----------------	--------------

simult()

Katalog > 

simult(koeff*Matrise*, konst*Vektor*(, 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.

Merk: Se også **linSolve()**, side 85.

koeffMatrise må være en kvadratmatrise som inneholder ligningskoeffisientene.

konstVektor må ha samme antall rader (samme dimension) som *koeffMatrise* og inneholde konstantene.

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:
 $5E-14 \cdot \max(\dim(koeffMatrise)) \cdot \text{radNavn}koeffMatrise)$

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.

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

Løsningen er $x=-3$ og $y=2$.

Løs:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \text{matx1} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\text{simult}\left(\text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \quad \begin{bmatrix} 0 \\ \frac{1}{2} \\ 2 \end{bmatrix}$$

Løs:

$$x + 2y = 1$$

$$3x + 4y = -1$$

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right) \quad \begin{bmatrix} -3 & -7 \\ 2 & \frac{9}{2} \end{bmatrix}$$

For det første systemet er $x=-3$ og $y=2$. For det andre systemet er $x=-7$ og $y=9/2$.

sin()

 **tast**

sin(*Verdi I*) $\Rightarrow$ *verdi*

sin(*Liste I*) $\Rightarrow$ *liste*

sin(*Verdi I*) returnerer sinus til argumentet.

sin(*Liste I*) returnerer en liste over sinus til alle elementer i *Liste I*.

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.

I Grader-vinkelmodus:

$\sin\left(\left(\frac{\pi}{4}\right)r\right)$	0.707107
$\sin(45)$	0.707107
$\sin(\{0,60,90\})$	$\{0.,0.866025,1.\}$

I Gradian-vinkelmodus:

$\sin(50)$	0.707107
------------	----------

I Radian-vinkelmodus:

$\sin\left(\frac{\pi}{4}\right)$	0.707107
$\sin(45^{\circ})$	0.707107

sin(*kvadratMatrise I*) $\Rightarrow$ *kvadratMatrise*

Returnerer matrisens sinus til *kvadratMatrise I*. Dette er ikke det samme som å beregne sinus til hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise I må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

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(*Liste I*) $\Rightarrow$ *liste*

sin⁻¹(*Verdi I*) returnerer vinkelen med sinus lik *Verdi I*.

sin⁻¹(*Liste I*) returnerer en liste over invers sinus til hvert element i *Liste I*.

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(...)**.

I Grader-vinkelmodus:

$\sin^{-1}(1)$	90.
----------------	-----

I Gradian-vinkelmodus:

$\sin^{-1}(1)$	100.
----------------	------

I Radian-vinkelmodus:

$\sin^{-1}(\text{kvadratMatrise } I) \Rightarrow \text{kvadratMatrise}$

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

kvadratMatrise I må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$\sin^{-1}(\{0,0.2,0.5\}) \quad \{0.,0.201358,0.523599\}$

I radian-vinkelmodus og rektangulær, kompleks modus:

$\sin^{-1}\left(\begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}\right)$
 $\begin{bmatrix} -0.174533-0.12198 \cdot i & 1.74533-2.35591 \cdot i \\ 1.39626-1.88473 \cdot i & 0.174533-0.593162 \cdot i \end{bmatrix}$

sinh()

Katalog > 

$\sinh(\text{Verdi } I) \Rightarrow \text{verdi}$

$\sinh(1.2) \quad 1.50946$

$\sinh(\text{Liste } I) \Rightarrow \text{liste}$

$\sinh(\{0,1,2,3\}) \quad \{0,1.50946,10.0179\}$

$\sinh(\text{Verdi } I)$ returnerer hyperbolsk sinus til argumentet.

$\sinh(\text{Liste } I)$ returnerer en liste over hyperbolsk sinus til hvert element i *Liste I*.

$\sinh(\text{kvadratMatrise } I) \Rightarrow \text{kvadratMatrise}$

Returnerer matrisens hyperbolske sinus for *kvadratMatrise I*. Dette er ikke det samme som å beregne hyperbolsk sinus for hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise I må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

$\sinh\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$
 $\begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$

$\sinh^{-1}()$

Katalog > 

$\sinh^{-1}(\text{Verdi } I) \Rightarrow \text{verdi}$

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

$\sinh^{-1}(\text{Liste } I) \Rightarrow \text{liste}$

$\sinh^{-1}(\{0,2,1,3\}) \quad \{0,1.48748,1.81845\}$

$\sinh^{-1}(\text{Verdi } I)$ returnerer invers hyperbolsk sinus til argumentet.

$\sinh^{-1}(\text{Liste } I)$ returnerer en liste over invers hyperbolsk sinus til hvert element i *Liste I*.

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

sinh⁻¹(kvadratMatrise1) ⇒ 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 151.)

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 numerisk eller streng 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 218).

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 <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>

SortA (SorterSt)

Katalog >

SortAListe1[, Liste2] [, Liste3]...

$\{2,1,4,3\} \rightarrow list1$ $\{2,1,4,3\}$

SortAVektor1[, Vektor2] [, Vektor3] ...

SortA list1 Done

Sorterer elementene i det første argument i stigende rekkefølge.

list1 $\{1,2,3,4\}$

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.

$\{4,3,2,1\} \rightarrow list2$ $\{4,3,2,1\}$

SortA list2,list1 Done

list2 $\{1,2,3,4\}$

list1 $\{4,3,2,1\}$

Alle argumentene må være navn på lister eller vektorer. Alle argumentene må ha like dimensjoner.

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 218.

SortD (SorterSy)

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</i> , <i>list2</i>	Done
-----------------------------------	------

<i>list1</i>	$\{4,3,2,1\}$
--------------	---------------

<i>list2</i>	$\{3,4,1,2\}$
--------------	---------------

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 218.

►Sphere (sfærisk)

Vektor ►Sphere

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

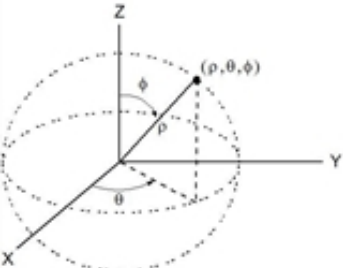
Viser rad- eller kolonnevektor i sfærisk form [ρ $\angle \theta$ $\angle \phi$].

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.

$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \text{►Sphere}$
$\begin{bmatrix} 3.74166 & \angle 1.10715 & \angle 0.640522 \end{bmatrix}$

$\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix} \text{►Sphere}$
$\begin{bmatrix} 3.60555 & \angle 0.785398 & \angle 0.588003 \end{bmatrix}$



sqrt() (kvdrt)	Katalog > 	
sqrt (<i>Verdi</i>)⇒ <i>verdi</i>	$\sqrt{4}$	2
sqrt (<i>Liste</i>)⇒ <i>liste</i>	$\sqrt{\{9,2,4\}}$	$\{3,1.41421,2\}$

Returnerer kvadratroten til argumentet.

For en liste, returneres kvadratroten til alle elementene i *Liste*.

Merk: Se også **Kvadratrot-sjablon**, side 1.

stat.results (stat.resultsats)

Katalog > 

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

xlist={1,2,3,4,5}

{1,2,3,4,5}

ylist={4,8,11,14,17}

{4,8,11,14,17}

LinRegMx xlist,ylist,1: stat.results

"Title"

"Linear Regression (mx+b)"

"RegEqn"

"m*x+b"

"m"

3.2

"b"

1.2

"r²"

0.996109

"r"

0.998053

"Resid"

"{...}"

stat.values

"Linear Regression (mx+b)"

"m*x+b"

3.2

1.2

0.996109

0.998053

"{-0.4,0.4,0.2,0,-0.2}"

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBlock
stat.AdjR ²	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r ²	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.σx	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.σy	stat.tlist
stat.b6	stat.e	stat.MSReg	stat.σx1	stat.UpperPred

stat.b7	stat.ExpMatrix	stat.MSRow	stat.σx2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.Χ
stat.b9	stat.FBlock	stat. $\hat{p}$	stat.Σx ²	stat.Χ1
stat.b10	stat.Fcol	stat. $\hat{p}$ 1	stat.Σxy	stat.Χ2
stat.bList	stat.FInteract	stat. $\hat{p}$ 2	stat.Σy	stat.ΧDiff
stat.χ ²	stat.FreqReg	stat. $\hat{p}$ Diff	stat.Σy ²	stat.ΧList
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg
stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.Complist	stat.LowerVal	stat.PValCol	stat.SEPred	stat.ȳ
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat.ŷ
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat.ŷList
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

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:

stDevPop() (stAvvPop)**Katalog >** 

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 218

**stDevPop(*MatriseI*[,
FrekvMatrise])**⇒*matrise*

Returnerer en radvektor av populasjonens standardavvik i kolonnene i *MatriseI*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

Merk: *MatriseI* må ha minst to rader. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 218.

stDevPop($\{\{1,2,5,-6,3,-2\}\}$)	3.59398
stDevPop($\{\{1.3,2.5,-6.4\},\{3,2,5\}\}$)	4.11107

stDevPop($\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}$)	$[3.26599 \quad 2.94392 \quad 1.63299]$
stDevPop($\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}$)	$[2.52608 \quad 5.21506]$

stDevSamp() (UtvstdAvv)**Katalog >** 

stDevSamp(*Liste*[,*frekvListe*])⇒*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 218

stDevSamp($\{\{1,2,5,-6,3,-2\}\}$)	3.937
stDevSamp($\{\{1.3,2.5,-6.4\},\{3,2,5\}\}$)	4.33345

stDevSamp() (UtvstdAvv)**Katalog** > **stDevSamp**(*Matrise1*[,
frekvMatrise])⇒*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 218.

$\text{stDevSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right)$	$\begin{bmatrix} 4. & 3.60555 & 2. \end{bmatrix}$
$\text{stDevSamp}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right)$	$\begin{bmatrix} 2.7005 & 5.44695 \end{bmatrix}$

Stop (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 produkhåndboken.

<i>i</i> :=0	0
Define <i>prog1</i> ()=Prgm	Done
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1</i> ()	Done
<i>i</i>	5

Lagre**Se → (lagre), side 201.****String() (Streng)****Katalog** > **string**(*Utttr*)⇒*string*

Forenkler *Utttr* og returnerer resultatet som en tegnstreng.

string(1.2345)	"1.2345"
string(1+2)	"3"

subMat() (undermatrise)**Katalog >** 

subMat(*MatriseI* [, *startRad*] [, *startKol*] [, *endRad*] [, *endKol*]) ⇒ *matrise*

Returnerer den spesifiserte undermatrisen til *MatriseI*.

Grunninnstillinger: *startRad*=1, *startKol*=1, *endRad*=siste rad, *endKol*=siste kolonne.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
<code>subMat(m1,2,1,3,2)</code>	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
<code>subMat(m1,2,2)</code>	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)**Se G(), side 193.****sum()****Katalog >** 

sum(*Liste* [, *Start*] [, *Slutt*]) ⇒ *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 218.

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 218.

<code>sum({1,2,3,4,5})</code>	15
<code>sum({a,2·a,3·a})</code>	"Error: Variable is not defined"
<code>sum(seq(n,n,1,10))</code>	55
<code>sum({1,3,5,7,9},3)</code>	21

<code>sum($\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix})$</code>	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
<code>sum($\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix})$</code>	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
<code>sum($\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},2,3)$</code>	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

sumIf()**Katalog >** 

sumIf(*Liste*,*Kriterium* [, *SumListe*]) ⇒ *verdi*

<code>sumIf({1,2,e,3,π,4,5,6},2.5<?<4.5)</code>	12.859874482
<code>sumIf({1,2,3,4},2<?<5,{10,20,30,40})</code>	70

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.

Hvis et *Liste*-element møter *Kriteriene*, legges dette elementet til den samlende summen. Hvis du inkluderer *sumListe*, legges tilsvarende element fra *sumListe* til summen istedenfor.

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

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 218.

Merk: Se også **countIf()**, side 31.

system(Verdi1 [, Verdi2 [, Verdi3 [, ...]])

Returnerer et ligningssystem, formatert som en liste. Du kan også opprette et system med en sjablon.

T(transponert)

katalog > 

Matrise $I^T \Rightarrow$ *matrise*

Returnerer den komplekse konjugerte transponerte av *Matrise* I .

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

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

tan()

 **tast**

tan(*Verdi* I) $\Rightarrow$ *verdi*

I Grader-vinkelmodus:

tan(*Liste* I) $\Rightarrow$ *liste*

tan(*Verdi* I) returnerer tangens til argumentet.

tan(*Liste* I) returnerer en liste over tangensene til alle elementene i *Liste* I .

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.

I Gradian-vinkelmodus:

$\tan\left(\left(\frac{\pi}{4}\right)r\right)$	1.
$\tan(45)$	1.
$\tan(\{0,60,90\})$	$\{0.,1.73205,\text{undef}\}$

I Radian-vinkelmodus:

$\tan\left(\left(\frac{\pi}{4}\right)r\right)$	1.
$\tan(50)$	1.
$\tan(\{0,50,100\})$	$\{0.,1.,\text{undef}\}$

tan(*kvadratMatrise* I) $\Rightarrow$ *kvadratMatrise*

Returnerer matrisetangensen av *kvadratMatrise* I . Dette er ikke det samme som å beregne tangens for hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

I Radian-vinkelmodus:

$\tan\left(\frac{\pi}{4}\right)$	1.
$\tan(45^{\circ})$	1.
$\tan\left(\left\{\pi,\frac{\pi}{3},\pi,\frac{\pi}{4}\right\}\right)$	$\{0.,1.73205,0.,1.\}$

tan()

 **tast**

kvadratMatrise I må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$$\tan \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

tan⁻¹()

 **tast**

tan⁻¹(*Verdi I*) ⇒ *verdi*

I Grader-vinkelmodus:

tan⁻¹(*Liste I*) ⇒ *liste*

$$\tan^{-1}(1) \quad 45$$

tan⁻¹(*Verdi I*) returnerer vinkelen med tangens til *Verdi I*.

I Gradian-vinkelmodus:

tan⁻¹(*Liste I*) returnerer en liste over de inverse tangenser til hvert element i *Liste I*.

$$\tan^{-1}(1) \quad 50$$

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

I Radian-vinkelmodus:

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

$$\tan^{-1}(\{0,0.2,0.5\}) \quad \{0,0.197396,0.463648\}$$

tan⁻¹(*kvadratMatrise I*) ⇒ *kvadratMatrise*

I Radian-vinkelmodus:

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

$$\tan^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

kvadratMatrise I må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

tanh()

Katalog > 

tanh(*Verdi I*) ⇒ *verdi*

$$\tanh(1.2) \quad 0.833655$$

tanh(*Liste I*) ⇒ *liste*

$$\tanh(\{0,1\}) \quad \{0,0.761594\}$$

tanh(*Verdi I*) returnerer hyperbolsk tangens til argumentet.

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(Verdi1)⇒verdi

tanh⁻¹(Liste1)⇒liste

tanh⁻¹(Verdi1) returnerer invers hyperbolsk tangens til argumentet.

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:

$$\begin{array}{ll} \tanh^{-1}(0) & 0. \\ \tanh^{-1}(\{1, 2, 1, 3\}) & \\ \{ \text{undef}, 0.518046-1.5708 \cdot i, 0.346574-1.5708 \cdot i \} & \end{array}$$

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

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 *nedGrense* og *øvreGrense* for spesifisert grader av frihet *df*.

For $P(X \leq \text{øvreGrense})$, sett *nedreGrense* = -9E999.

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 129 eller **RequestStr**, side 131.

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 tilfeldige tallene i en dialogboks.

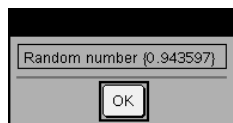
Innenfor malen *Prgm...EndPrgm* fullfører du hver linje ved å trykke på  istedenfor på **Enter**. På tastaturet på datamaskinen, hold nede **Alt** og trykk på **Enter**.

```
Define tekst_demo()=Prgm
  For i,1,5
    strinfo:="Random number " &
      string(rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Kjør program:

tekst_demo()

Eksempel på dialogboks:



Then (Så)

tInterval

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 151.)

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 218).

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 151).

Felles=1 fellesvarianser; *Felles=0* gjør ikke fellesvarianser.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 218).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling
stat. $\bar{x}1$ - $\bar{x}2$	Uvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.df	Grader-av-frihet
stat. $\bar{x}1$, stat. $\bar{x}2$	Uvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat. $\sigma x1$, stat. $\sigma x2$	Utalgets 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 > 

trace(*kvadratMatrise*) $\Rightarrow$ *verdi*

Returnerer diagonalsummen (summen av alle elementene på hoveddiagonalen) til *kvadratMatrise*.

trace $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	15
<i>a</i> :=12	12
trace $\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}$	24

```
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 228.

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 flerlinjede 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.

$$\text{egenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

Merk: Se også **ClrErr**, side 23, og **PassErr**, side 114.

```
Define prog1()=Prgm
  Try
    z:=z+1
  Disp "z incremented."
  Else
    Disp "Sorry, z undefined."
  EndTry
EndPrgm
```

Done

```
z:=1:prog1()
z incremented.
```

Done

```
DelVar z:prog1()
Sorry, z undefined.
```

Done

Define 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

EndIf

EndTry

EndPrgm

tTest**tTest** μ_0 ,*Liste*[,*Frekv*[,*Hypot*]]

(Dataliste inndata)

tTest μ_0 , $\bar{x}$,*sx*,*n*, [*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 151).

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 218).

Utdata-variabel	Beskrivelse
stat.t	$(\bar{x} - \mu_0) / (\text{stdev} / \sqrt{n})$
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**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 151).

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 218).

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

Finansiell funksjon som beregner fremtidig verdi for penger.

tvmFV()Katalog > 

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 167. 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 167. 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 167. 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 167. 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 167. 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 finansiøser. Men finansielle funksjoner lagrer ikke argumentverdiene eller resultatene til TVM-variablene.

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

Beregner 2-variabels statistiske observatorer. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 151.)

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 numerisk eller streng 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 218.

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 $x \cdot y$ -verdier
stat. r	Korrelasjonskoeffisient

Utdata-variabel	Beskrivelse
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.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(<i>Vektor1</i>)⇒vektor Returnerer enten en rad- eller kolonne-enhetsvektor, avhengig av formen på <i>Vektor1</i> . <i>Vektor1</i> må være enten en enkel-rad-matrise eller en enkel-kolonne-matrise.	unitV([1 2 1]) [0.408248 0.816497 0.408248] unitV([1; 2; 3]) [0.267261; 0.534522; 0.801784]

unLock	Katalog > 
unLockVar1 [, Var2] [, Var3] ... unLockVar. Låser opp spesifisert variabel eller variabelgruppe. Låste variabler kan ikke modifiseres eller slettes. Se Lock , side 89, og getLockInfo() , side 67.	a:=65 65 Lock a Done getLockInfo(a) 1 a:=75 "Error: Variable is locked." DelVar a "Error: Variable is locked." Unlock a Done a:=75 75 DelVar a Done

varPop()

Katalog >

varPop(Liste[,frekvListe])⇒uttrykk

varPop({5,10,15,20,25,30})72.9167

Returnerer 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), ignorerer dette elementet, og det tilsvarende elementet i den andre listen ignorerer også. For mer informasjon om tomme elementer, se side 218.

varSamp() (utvalgets varians)

Katalog >

varSamp(Liste[,frekvListe])⇒uttrykk

varSamp({1,2,5,-6,3,-2})312

Returnerer utvalgets varians for *Liste*.

Hvert *frekvListe* element teller antallet forekomster av det tilsvarende elementet i *Liste*.

varSamp({1,3,5},{4,6,2})6833

Merk: *Liste* må inneholde minst to elementer.

Hvis et element i en av listene er tomt (åpent), ignorerer dette elementet, og det tilsvarende elementet i den andre listen ignorerer også. For mer informasjon om tomme elementer, se side 218.

varSamp(Matrise1[,frekvMatrise])⇒matrise

Returnerer en radvektor som inneholder utvalgets varians for hver kolonne i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

varSamp($\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{pmatrix}$) $\begin{bmatrix} 4.75 & 1.03 & 4 \end{bmatrix}$

varSamp($\begin{pmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{pmatrix}, \begin{pmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{pmatrix}$) $\begin{bmatrix} 3.91731 & 2.08411 \end{bmatrix}$

Hvis et element i en av matrisene er tomt (åpent), ignorerer dette elementet, og det tilsvarende elementet i den andre matrisen ignorerer også. For mer informasjon om tomme elementer, se side 218.

Merk: *Matrise I* 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 variablen *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(*Uttr1*, *StatusVar*) $\Rightarrow$ 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 236 for en liste over varselkoder og assosierte meldinger.

 warnCodes(det([1.23456E-999]),warn)	
	1.23456E-999
warn	{ 10029 }

when() (når)

when(*Betingelse*, *santResultat* [, *usantResultat*][, *ukjentResultat*])
 $\Rightarrow$ 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.

when($x < 0, x + 3$) $x = 5$	undef
----------------------------------	-------

when($n > 0, n \cdot \text{factorial}(n - 1), 1$) $\rightarrow$ factorial(n)	Done
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 produkthåndboken.

Define <i>sum_of_recip</i> (<i>n</i>)=Func	
Local <i>i,tempsum</i>	
<i>1</i> → <i>i</i>	
<i>0</i> → <i>tempsum</i>	
While <i>i</i> ≤ <i>n</i>	
<i>tempsum</i> + $\frac{1}{i}$ → <i>tempsum</i>	
<i>i</i> + 1 → <i>i</i>	
EndWhile	
Return <i>tempsum</i>	
EndFunc	
Done	
<i>sum_of_recip</i> (3)	$\frac{11}{6}$

X

BoolskUttr1 **xor** *BoolskUttr2* returnerer *Boolsk uttrykk*

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 112.

Heltall1 **xor** *Heltall2* ⇒ *heltall*

true xor true	false
5>3 xor 3>5	true

I heksades grunntall-modus:	
Viktig: Null, ikke bokstaven O.	
0h7AC36 xor 0h3D5F	0h79169

I binær grunntall-modus:	
0b100101 xor 0b100	0b100001

Sammenlikner to reelle heltall bit-for-bit med en **xor**-handling. Internt er begge heltallene omregnet til 64-bits 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 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 **or**, side 112.

Z

zInterval

zInterval $\sigma, \text{Liste[, Frekv[, CNivå]]}$

(Dataliste inndata)

zInterval $\sigma, \bar{x}, n \text{ [, CNivå]}$

(Summering statistikk inndata)

Beregner et z konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen (side 151).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 218).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall for et ukjent populasjonsgjenomsnitt

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 151).

x er et ikke-negativt heltall.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 218).

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 151).

$x1$ og $x2$ er ikke-negative heltall.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 218).

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 σ_1, σ_2 , *Liste1*, *Liste2*
[,*Frekv1*][,*Frekv2*,[*CNivå*]]

(Dataliste inndata)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}_1, n1, \bar{x}_2, n2$ [*CNivå*])

(Summering statistikk inndata)

Beregner et to-utvalgs z konfidensintervall.
En oversikt over resultatene lagres i
stat.results-variabelen. (Se side 151).

For informasjon om effekten av tomme
elementer i en liste, se "Tomme (åpne)
elementer" (side 218).

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. σx_1 , stat. σx_2	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 *frekvliste*.

En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 151).

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 218).

Utdata-variabel	Beskrivelse
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$
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 $p_0, x, n, [\text{Hypot}]$

Beregner en en-proporsjons z -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 151).

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 218).

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 151).

$x1$ 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 218).

Utdata-variabel	Beskrivelse
stat.z	Standard normalverdi beregnet for differansen av andelen
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 151).

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 218).

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
$Verdi1 + Verdi2 \Rightarrow verdi$	56	56
Returnerer summen av de to argumentene.	56+4	60
	60+4	64
	64+4	68
	68+4	72
$Liste1 + Liste2 \Rightarrow liste$	$\left\{ 22, \pi, \frac{\pi}{2} \right\} \rightarrow I1$	$\{ 22, 3.14159, 1.5708 \}$
$Matrise1 + Matrise2 \Rightarrow matrise$	$\left\{ 10, 5, \frac{\pi}{2} \right\} \rightarrow I2$	$\{ 10, 5, 1.5708 \}$
Returnerer en liste (eller matrise) som inneholder summene av tilsvarende elementer i <i>Liste1</i> og <i>Liste2</i> (eller <i>Matrise1</i> og <i>Matrise2</i>).	$I1+I2$	$\{ 32, 8.14159, 3.14159 \}$
Dimensjonene i argumentene må være like.	$15 + \{ 10, 15, 20 \}$	$\{ 25, 30, 35 \}$
	$\{ 10, 15, 20 \} + 15$	$\{ 25, 30, 35 \}$
$Verdi + Liste1 \Rightarrow liste$		
$Liste1 + Verdi \Rightarrow liste$		
Returnerer en liste med summene av <i>Verdi</i> og hvert element i <i>Liste1</i> .		
$Verdi + Matrise1 \Rightarrow matrise$	$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
$Matrise1 + Verdi \Rightarrow matrise$		
Returnerer en matrise med <i>Verdi</i> addert til hvert element på diagonalen til <i>Matrise1</i> . <i>Matrise1</i> må være kvadratisk.		
Merk: Bruk .+ (prikk pluss) for å addere et uttrykk til hvert element.		

- (subtrahere)		- tast
$Verdi1 - Verdi2 \Rightarrow verdi$	6-2	4
Returnerer <i>Verdi1</i> minus <i>Verdi2</i> .	$\pi - \frac{\pi}{6}$	2.61799
$Liste1 - Liste2 \Rightarrow liste$	$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\}$	$\{ 12, -1.85841, 0. \}$
$Matrise1 - Matrise2 \Rightarrow matrise$	$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 2 \end{bmatrix}$

–(subtrahere)



Subtraherer hvert element i *Liste2* (eller *Matrise2*) fra tilsvarende element i *Liste1* (eller *Matrise1*), og returnerer resultatene.

Dimensjonene i argumentene må være like.

Verdi – *Liste1* $\Rightarrow$ *liste*

$$15 - \{10, 15, 20\} \quad \{5, 0, -5\}$$

Liste1 – *Verdi* $\Rightarrow$ *liste*

$$\{10, 15, 20\} - 15 \quad \{-5, 0, 5\}$$

Subtraherer hvert *Liste1* element fra *Verdi* eller subtraherer *Verdi* fra hvert *Liste1* element og returnerer en liste over resultatene.

Verdi – *Matrise1* $\Rightarrow$ *matrise*

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

Matrise1 – *Verdi* $\Rightarrow$ *matrise*

Verdi – *Matrise1* returnerer en matrise av *Verdi* hver gang identitetsmatrisen trekkes fra *Matrise1*. *Matrise1* må være kvadratisk.

Matrise1 – *Verdi* returnerer en matrise av *Verdi* hver gang identitetsmatrisen subtraheres fra *Matrise1*. *Matrise1* må være kvadratisk.

Merk: Bruk .– (prikk minus) for å subtrahere et uttrykk fra hvert element.

•(multiplisere)



Verdi1 • *Verdi2* $\Rightarrow$ *verdi*

$$2 \cdot 3.45 \quad 6.9$$

Returnerer produktet av de to argumentene.

Liste1 • *Liste2* $\Rightarrow$ *liste*

$$\{1., 2, 3\} \cdot \{4, 5, 6\} \quad \{4, 10, 18\}$$

Returnerer en liste som inneholder produktene av samsvarende elementer i *Liste1* og *Liste2*.

Dimensjonene i listene må være like.

Matrise1 • *Matrise2* $\Rightarrow$ *matrise*

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 7 & 8 \\ 7 & 8 \\ 7 & 8 \end{bmatrix} \quad \begin{bmatrix} 42 & 48 \\ 105 & 120 \end{bmatrix}$$

Returnerer matriseproduktet av *Matrise1* og *Matrise2*.

Antallet kolonner i *Matrise1* må være likt antallet rader i *Matrise2*.

•(multiplisere)

 **tast**

Verdi • *Liste1* $\Rightarrow$ *liste*

$$\pi \cdot \{4,5,6\} \quad \{12.5664, 15.708, 18.8496\}$$

Liste1 • *Verdi* $\Rightarrow$ *liste*

Returnerer en liste med produktene av *Verdi* og hvert element i *Liste1*.

Verdi • *Matrise1* $\Rightarrow$ *matrise*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 \quad \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

Matrise1 • *Verdi* $\Rightarrow$ *matrise*

Returnerer en matrise med produktene av *Verdi* og hvert element i *Matrise1*.

$$6 \cdot \text{identity}(3) \quad \begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

Merk: Bruk • (prikk multipliser) for å multiplisere et uttrykk med hvert element.

/ (divider)

 **tast**

Verdi1 / *Verdi2* $\Rightarrow$ *verdi*

$$\frac{2}{3.45} \quad 0.57971$$

Returnerer kvotienten av *Verdi1* dividert med *Verdi2*.

Merk: Se også **Brøk-sjablon**, side 1.

Liste1 / *Liste2* $\Rightarrow$ *liste*

$$\frac{\{1,2,3\}}{\{4,5,6\}} \quad \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

Returnerer en liste som inneholder kvotientene av *Liste1* dividert med *Liste2*.

Dimensjonene i listene må være like.

Verdi / *Liste1* $\Rightarrow$ *liste*

$$\frac{6}{\{3,6,\sqrt{6}\}} \quad \{2, 1, 2.44949\}$$

Liste1 / *Verdi* $\Rightarrow$ *liste*

$$\frac{\{7,9,2\}}{7 \cdot 9 \cdot 2} \quad \left\{\frac{1}{18}, \frac{1}{14}, \frac{1}{63}\right\}$$

Returnerer en liste med kvotientene av *Verdi* dividert med *Liste1* eller *Liste1* dividert med *Verdi*.

Verdi / *Matrise1* $\Rightarrow$ *matrise*

$$\frac{\begin{bmatrix} 7 & 9 & 2 \end{bmatrix}}{7 \cdot 9 \cdot 2} \quad \begin{bmatrix} \frac{1}{18} & \frac{1}{14} & \frac{1}{63} \end{bmatrix}$$

Matrise1 / *Verdi* $\Rightarrow$ *matrise*

Returnerer en matrise med kvotientene av *Matrise1* / *Verdi*.

Merk: Bruk / (prikk divider) for å dividere et uttrykk med hvert element.

^ (potens) **tast***Verdi1* ^ *Verdi2* $\Rightarrow$ *verdi*

$$4^2 \quad 16$$

Liste1 ^ *Liste2* $\Rightarrow$ *liste*

$$\{2,4,6\}^{\{1,2,3\}} \quad \{2,16,216\}$$

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*.

I reell grunnmengde bruker brøkpotens som har forkortet eksponent med oddetall i nevner en rell forgreining i motsetning til hovedforgreining for kompleks modus.

Verdi ^ *Liste1* $\Rightarrow$ *liste*

$$\pi^{\{1,2,-3\}} \quad \{3.14159,9.8696,0.032252\}$$

Returnerer *Verdi* opphøyd i elementene i *Liste1*.

Liste1 ^ *Verdi* $\Rightarrow$ *liste*

$$\{1,2,3,4\}^{-2} \quad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

Returnerer elementene i *Liste1* opphøyd i *Verdi*.

kvadratMatrise1 ^ *heltall* $\Rightarrow$ *matrise*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \quad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

Returnerer *kvadratMatrise1* opphøyd i *heltall* -potens.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

kvadratMatrise1 må være en kvadratmatrise.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

Hvis *heltall* = -1, beregnes invers matrise.

Hvis *heltall* < -1, beregnes invers matrise opphøyd i en korrekt positiv potens.

x² (kvadrat) **tast***Verdi* ² $\Rightarrow$ *verdi*

$$4^2 \quad 16$$

Returnerer kvadratet av argumentet.

$$\{2,4,6\}^2 \quad \{4,16,36\}$$

Liste1 ² $\Rightarrow$ *liste*

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2 \quad \begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$$

Returnerer en liste med kvadrater av elementene i *Liste1*.

kvadratMatrise1 ² $\Rightarrow$ *matrise*

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^{\cdot^2} \quad \begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$$

x² (kvadrat)

 **tast**

Returnerer matrisens kvadrat av *kvadratMatrise1*. Dette er ikke det samme som å beregne kvadratet av hvert element. Bruk $\cdot^2$ for å beregne kvadratet av hvert element.

.+ (prikk adder)

 **taster**

Matrise1 .+ *Matrise2* $\Rightarrow$ *matrise*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix} = \begin{bmatrix} 11 & 32 \\ 23 & 44 \end{bmatrix}$$

Verdi .+ *Matrise1* $\Rightarrow$ *matrise*

$$5 .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix} = \begin{bmatrix} 15 & 35 \\ 25 & 45 \end{bmatrix}$$

Matrise1 .+ *Matrise2* returnerer en matrise som er summen av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

Verdi .+ *Matrise1* returnerer en matrise som er summen av *Verdi* og hvert element i *Matrise1*.

.- (prikk subtr.)

 **taster**

Matrise1 .- *Matrise2* $\Rightarrow$ *matrise*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .- \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} -9 & -18 \\ -27 & -36 \end{bmatrix}$$

Verdi .- *Matrise1* $\Rightarrow$ *matrise*

$$5 .- \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} -5 & -15 \\ -25 & -35 \end{bmatrix}$$

Matrise1 .- *Matrise2* returnerer en matrise som er differansen mellom hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

Verdi .- *Matrise1* returnerer en matrise som er differansen av *Verdi* og hvert element i *Matrise1*.

.• (prikk mult.)

 **taster**

Matrise1 .• *Matrise2* $\Rightarrow$ *matrise*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .• \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 10 & 40 \\ 90 & 160 \end{bmatrix}$$

Verdi .• *Matrise1* $\Rightarrow$ *matrise*

$$5 .• \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 50 & 100 \\ 150 & 200 \end{bmatrix}$$

Matrise1 .• *Matrise2* returnerer en matrise som er produktet av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

.• (prikk mult.) **$\cdot$ $\times$ taster**

Verdi .• *Matrise1* returnerer en matrise med produktene av *Verdi* og hvert element i *Matrise1*.

. / (prikk divider) **$\div$ taster**

Matrise1 ./ *Matrise2* $\Rightarrow$ matrise

Verdi ./ *Matrise1* $\Rightarrow$ matrise

Matrise1 ./ *Matrise2* returnerer en matrise som er kvotient av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

Verdi ./ *Matrise1* returnerer en matrise som er kvotient av *Verdi* og hvert element i *Matrise1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} ./ \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} \\ \frac{1}{30} & \frac{1}{40} \end{bmatrix}$$

$$5 ./ \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{8} \end{bmatrix}$$

.^ (prikk potens) **$\wedge$ taster**

Matrise1 .^ *Matrise2* $\Rightarrow$ matrise

Verdi .^ *Matrise1* $\Rightarrow$ matrise

Matrise1 .^ *Matrise2* returnerer en matrise, der hvert element i *Matrise2* er eksponenten for samsvarende element i *Matrise1*.

Verdi .^ *Matrise1* returnerer en matrise, der hvert element i *Matrise1* er eksponenten for *Verdi*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .^ \begin{bmatrix} 0 & 2 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 27 & \frac{1}{4} \end{bmatrix}$$

$$5 .^ \begin{bmatrix} 0 & 2 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 25 \\ 125 & \frac{1}{5} \end{bmatrix}$$

-(negere) **$\neg$ tast**

-*Verdi* $\Rightarrow$ *verdi*

-*Liste1* $\Rightarrow$ *liste*

-*Matrise1* $\Rightarrow$ *matrise*

Returnerer argumentets negasjon.

For en liste eller matrise returneres alle elementene negert.

$$\begin{array}{cc} -2.43 & -2.43 \\ -\{-1, 0.4, 1.2\mathbb{E}19\} & \{1., -0.4, -1.2\mathbb{E}19\} \end{array}$$

I binær grunntall-modus:

Viktig: Null, ikke bokstaven O

- (negere)

 **tast**

Hvis argumentet er et binært eller heksadesimalt heltall, gir negasjonen komplementet til to.

```
-0b100101
0b11111111111111111111111111111111▶
```

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

% (prosent)

ctrl  **taster**

Verdi1 % $\Rightarrow$ *verdi*

Liste1 % $\Rightarrow$ *liste*

Matrise1 % $\Rightarrow$ *matrise*

argument

Returnerer **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 **ctrl** **enter**.

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*

Returnerer sann hvis *Uttr1* er bestemt å være lik *Uttr2*.

Returnerer 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 produkthåndboken.

Eksempel på funksjon som bruker matematiske testsymboler: =, $\neq$, <, ≤, >, ≥

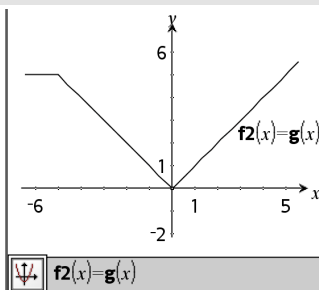
```
Define g(x)=Func
  If x≤-5 Then
    Return 5
  ElseIf x>-5 and x<0 Then
    Return -x
  ElseIf x≥0 and x≠10 Then
    Return x
  ElseIf x=10 Then
    Return 3
  EndIf
EndFunc
```

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.

$\leq$ (mindre enn eller lik)

  taster

$Uttr1 \leq Uttr2 \Rightarrow \text{Boolsk uttrykk}$

Se "=" (er lik) eksempel.

$Liste1 \leq Liste2 \Rightarrow \text{Boolsk liste}$

$Matrise1 \leq Matrise2 \Rightarrow \text{Boolsk matrise}$

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 $\leq$

> (større enn)

  taster

$Uttr1 > Uttr2 \Rightarrow \text{Boolsk uttrykk}$

Se "=" (er lik) eksempel.

$Liste1 > Liste2 \Rightarrow \text{Boolsk liste}$

$Matrise1 > Matrise2 \Rightarrow \text{Boolsk matrise}$

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.

$\geq$ (større enn eller lik med)

  -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 $\geq$

$\Rightarrow$ (logisk implikasjon)

  -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** $\langle \text{argument1} \rangle$ or $\langle \text{argument2} \rangle$ 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 $\Rightarrow$

⇔ (logisk dobbel implikasjon, XNOR)

  -taster

BoolskUttr1 ⇔ *BoolskUttr2* returnerer
Boolsk uttrykk

$5 > 3 \text{ xor } 3 > 5$ true

BoolskListe1 ⇔ *BoolskListe2* returnerer
Boolsk liste

$5 > 3 \Leftrightarrow 3 > 5$ false

$3 \text{ xor } 4$ 7

$3 \Leftrightarrow 4$ -8

BoolskMatrise1 ⇔ *BoolskMatrise2*
returnerer *Boolsk matrise*

$\{1,2,3\} \text{ xor } \{3,2,1\}$ $\{2,0,2\}$

$\{1,2,3\} \Leftrightarrow \{3,2,1\}$ $\{-3,-1,-3\}$

Heltall1 ⇔ *Heltall2* returnerer *Heltall*

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 $\Leftrightarrow$

! (fakultet)

 **tast**

Verdi1! ⇒ *verdi*

$5!$ 120

Liste1! ⇒ *liste*

$\{\{5,4,3\}\}!$ $\{120,24,6\}$

Matrise1! ⇒ *matrise*

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

Returnerer argumentets fakultet.

For en liste eller matrise, returneres en liste eller matrise av elementenes fakulteter.

& (legg til)

  **taster**

Streng1 & *Streng2* ⇒ *streng*

"Hello " & "Nick" "Hello Nick"

Returnerer en tekststreng som er *Streng2* lagt til *Streng1*.

d(*Uttr1*, *Var*[, *Orden*]) |
Var=*Verdi*⇒*verdi*

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

d(*Uttr1*, *Var*[, *Orden*])⇒*verdi*

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

d(*Liste1*, *Var*[, *Orden*])⇒*liste*

$$x:=3: \frac{d}{dx}(\{x^2, x^3, x^4\}) \quad \{6, 27, 108\}$$

d(*Matrise1*, *Var*[, *Orden*])⇒*matrise*

Unntatt mens du bruker den første syntaksen, må du lagre en numerisk verdi i variabel *Var* før du behandler **d()**. Se eksemplene.

d() kan brukes for å beregne første og andre deriverte i et punkt numerisk ved hjelp av automatiske derivasjonsmetoder.

Orden, hvis inkludert, må være=1 eller 2. Standard er 1.

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

Merk: Se også Første deriverte, side 5 eller Andre deriverte, side 6.

Merk: Algoritmen **d()** har en begrensning: den arbeider rekursivt gjennom det ikke-forenklede uttrykket og beregner den numeriske verdien av den første deriverte (og eventuelt den andre) og behandlingen av hvert deluttrykk, som kan føre til et uventet resultat.

$$\frac{d}{dx} \left(x \cdot (x^2 + x)^{\frac{1}{3}} \right) |_{x=0} \quad \text{undef}$$

$$\text{centralDiff} \left(x \cdot (x^2 + x)^{\frac{1}{3}}, x \right) |_{x=0} \quad 0.000033$$

Studer eksemplet til høyre. Den første deriverte av $x \cdot (x^2 + x)^{1/3}$ i $x=0$ er lik 0. Men siden den første deriverte av deluttrykket $(x^2 + x)^{1/3}$ er udefinert i $x=0$ og denne verdien blir brukt til å beregne den deriverte av hele uttrykket, rapporterer **d()** resultatet som udefinert og viser en varselmelding.

Hvis du støter på denne begrensningen, verifiserer du løsningen grafisk. Du kan også prøve å bruke **centralDiff()**.

$\int()$ (integral)

Katalog > 

$\int(Uttr1, Var, Nedre, Øvre) \Rightarrow verdi$

Returnerer integralet av *Uttr1* med hensyn på variabelen *Var* fra *Nedre* til *Øvre*. Kan brukes for å beregne den bestemte integralet numerisk ved hjelp av samme metode som `nInt()`.

$$\int_0^1 x^2 dx = 0.333333$$

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

Merk: Se også `nInt()`, side 106 og **Bestemt integral-sjablon**, side 6.

$\sqrt{}$ (kvadratrot)

  **taster**

$\sqrt{Verdi1} \Rightarrow verdi$

$$\sqrt{4} = 2$$

$\sqrt{Liste1} \Rightarrow liste$

$$\sqrt{\{9,2,4\}} = \{3,1.41421,2\}$$

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 **sqr t (...)**

Merk: Se også **Kvadratrot-sjablon**, side 1.

$\prod()$ (prodSeq)

Katalog > 

$\prod(Uttr1, Var, Nedre, Øvre) \Rightarrow uttrykk$

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

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

Finner *Uttrykk1* for hver verdi av *Var* fra *Nedre* til *Øvre*, og returnerer produktet av resultatene.

$$\prod_{n=1}^5 \left(\left\{ \frac{1}{n}, n, 2 \right\} \right) = \left\{ \frac{1}{120}, 120, 32 \right\}$$

Merk: Se også **Produkt-sjablon** ($\prod$), side 5.

$\prod(Uttr1, Var, Nedre, Nedre-1) \Rightarrow 1$

$$\prod_{k=4}^3 (k) = 1$$

$\prod(Uttr1, Var, Nedre, Øvre) \Rightarrow 1/\prod(Uttr1, Var, Øvre+1, Nedre-1)$ hvis $Øvre < Nedre$

$\prod()$ (prodSeq)

Katalog > 

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.

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) = 6$$
$$\prod_{k=4}^1 \left(\frac{1}{k} \right) \cdot \prod_{k=2}^4 \left(\frac{1}{k} \right) = \frac{1}{4}$$

$\Sigma()$ (sumSeq)

Katalog > 

$\Sigma(\text{Uttr1}, \text{Var}, \text{Nedre}, \text{Øvre}) \Rightarrow \text{uttrykk}$

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 $\text{Øvre} < \text{Nedre}-1$

$$\sum_{n=1}^5 \left(\frac{1}{n} \right) = \frac{137}{60}$$

$$\sum_{k=4}^3 (k) = 0$$

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.

$$\sum_{k=4}^1 (k) = -5$$
$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) = 4$$

$\Sigma\text{Int}()$

Katalog > 

$\Sigma\text{Int}(\text{NPmt1}, \text{NPmt2}, \text{N}, \text{I}, \text{PV}, [\text{Pmt}], [\text{FV}], [\text{PpY}], [\text{CpY}], [\text{PmtAt}], [\text{avrundVerdi}]) \Rightarrow \text{verdi}$

$\Sigma\text{Int}(\text{NPmt1}, \text{NPmt2}, \text{amortTabell}) \Rightarrow \text{verdi}$

$$\Sigma\text{Int}(1, 3, 12, 4.75, 20000, 12, 12) = -213.48$$

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 167.

- 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.

ΣInt(*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 **ΣPrn()**, nedenfor, og **Bal()**, side 15.

tbl:=amortTbl(12,12,4.75,20000,,12,12)

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

ΣInt(1,3,tbl) -213.48

ΣPrn(*NPmt1*, *NPmt2*, *N*, *I*, *PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*avrundVerdi*])⇒verdi

ΣPrn(1,3,12,4.75,20000,,12,12) -4916.28

ΣPrn(*NPmt1*,*NPmt2*,*amortTable*)⇒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 167.

- Hvis du utelater Pmt , grunninnstilles den til $Pmt = \mathbf{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.
- 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.

$\Sigma\mathbf{Prn}(NPmt1, NPmt2, amortTabell)$ beregner summen av hovedbetalinger basert på amortiseringstabell *amortTabell*. Argumentet *amortTabell* må være en matrise i den form som er beskrevet under $\mathbf{amortTbl}()$, side 7.

Merk: Se også $\Sigma\mathbf{Int}()$, over, og $\mathbf{Bal}()$, side 15.

$tbl := \mathbf{amortTbl}(12, 12, 4.75, 20000., 12, 12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\mathbf{Prn}(1, 3, tbl)$ -4916.28

(Indir.ref)

  **taster**

varNavnStreng

$xyz := 12$ 12

Refererer til variabelen med navnet *varNavnStreng*. På denne måten kan du bruke strenger for å opprette variabelnavn "innenfra" en funksjon.

$\#("x" \& "y" \& "z")$ 12

Oppretter eller refererer til variabelen xyz.

$10 \rightarrow r$ 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)**EE** tast*mantissa***E***eksponent*

23000. 23000.

Legger inn et tall i vitenskapelig fremstilling. Tallet blir tolket som en *mantissa* $\times 10^{\text{eksponent}}$.

2300000000.+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 **@E**. Eksempel: Skriv **2.3@E4** for å legge inn **2.3E4**.

g (gradian)**1** tast*UttrI***g** $\Rightarrow$ *Uttrykk*

I grader, gradian eller radian modus:

*UttrI***g** $\Rightarrow$ *Uttrykk* $\cos(50^g)$ 0.707107*ListeI***g** $\Rightarrow$ *liste* $\cos(\{0,100^g,200^g\})$ $\{1,0,-1.\}$ *MatriseI***g** $\Rightarrow$ *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 *UttrI* med $\pi/200$.

I grader-vinkelmodus, multipliseres *UttrI* med $g/100$.

I gradian modus, returneres *UttrI* uendret.

Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive **@g**.

r (radian)**1** tast*VerdiI***r** $\Rightarrow$ *verdi*

I grader, gradian eller radian modus:

*ListeI***r** $\Rightarrow$ *liste**MatriseI***r** $\Rightarrow$ *matrise*

ʀ (radian)

1 tast

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 ʀ 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.

$$\cos\left(\frac{\pi}{4r}\right) \quad 0.707107$$

$$\cos\left(\left\{0r, \left(\frac{\pi}{12}\right)r, -(\pi)r\right\}\right) \quad \{1, 0.965926, -1.\}$$

° (grader)

1 tast

Verdi ʀ° ⇒ *verdi*

Liste ʀ° ⇒ *liste*

Matrise ʀ° ⇒ *matrise*

Denne funksjonen gir deg en mulighet til å spesifisere en grader-vinkel mens du er i gradian eller radian modus.

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.

I grader, gradian eller radian modus:

$$\cos(45^\circ) \quad 0.707107$$

I radian modus:

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) \quad \{1, 0.707107, 0., 0.864976\}$$

°, ', " (grader/minutter/sekunder)

ctrl  **taster**

gg°mm'ss.ss" ⇒ *Uttrykk*

I Grader-vinkelmodus:

gg Et positivt eller negativt tall

$^{\circ}, ', ''$ (grader/minutter/sekunder)	  taster
<i>mm</i> Et ikke-negativt tall	$25^{\circ}13'17.5''$ 25.2215
<i>ss.ss</i> Et ikke-negativt tall	$25^{\circ}30'$ $\frac{51}{2}$

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 (').

$\angle$ (vinkel)	  taster
$[Radius, \angle _Vinkel] \Rightarrow vektor$ (polar inndata)	I radian modus og vektorformat innstilt på: rektangulær
$[Radius, \angle _Vinkel, Z_Koordinat] \Rightarrow vektor$ (sylindrisk inndata)	$\begin{bmatrix} 5 & \angle 60^{\circ} & \angle 45^{\circ} \\ & 1.76777 & 3.06186 & 3.53553 \end{bmatrix}$
$[Radius, \angle _Vinkel, \angle _Vinkel] \Rightarrow vektor$ (sfærisk inndata)	sylindrisk
Returnerer koordinater som en vektor, avhengig av vektorformatets modusinnstilling: rektangulær, sylindrisk eller sfærisk.	$\begin{bmatrix} 5 & \angle 60^{\circ} & \angle 45^{\circ} \\ & 3.53553 & 1.0472 & 3.53553 \end{bmatrix}$
Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive @<.	sfærisk
$(Størrelse \angle Vinkel) \Rightarrow kompleksVerdi$ (polar inndata)	$\begin{bmatrix} 5 & \angle 60^{\circ} & \angle 45^{\circ} \\ & 5. & 1.0472 & \angle 0.785398 \end{bmatrix}$
Setter inn en kompleks verdi i $(r\angle\theta)$ polar form. <i>Vinkelen</i> tolkes avhengig av aktuell vinkelmodus-innstilling.	I radian-vinkelmodus og rektangulært, komplekst format: $5+3 \cdot i - \left(10 \angle \frac{\pi}{4} \right) \quad -2.07107-4.07107 \cdot i$

_ (senket strek som et tomt element)**10^()****Katalog >** **10^ (Verdi)** ⇒ verdi $10^{1.5}$

31.6228

10^ (Liste) ⇒ liste

Returnerer 10 opphøyd i argumentets potens.

For en liste, returneres 10 opphøyd i elementenes potens i *Liste*.

10^(kvadratMatrise) ⇒ kvadratMatrise

Returnerer 10 opphøyd i potensen av *kvadratMatrise*. Dette er ikke det samme som å beregne 10 opphøyd i potens av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} = \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

^-1(resiprok)**Katalog >** **Verdi** ^-1 ⇒ verdi $(3.1)^{-1}$

0.322581

Liste ^-1 ⇒ liste

Returnerer den inverse verdien av et argument.

For en liste, returneres den inverse verdien av elementene i *Liste*.

kvadratMatrise ^-1 ⇒ kvadratMatrise

Returnerer den inverse verdien av *kvadratMatrise*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

kvadratMatrise må være en ikke-singulær kvadratisk matrise.

Uttr | BoolskUttr1 [and BoolskUttr2]...

$x+1 x=3$	4
-----------	---

Uttr | BoolskUttr1 [or BoolskUttr2]...

$x+55 x=\sin(55)$	54.0020
-------------------	---------

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” operatorer.

Med begrensningsoperatoren har du tre utgangstyper av funksjonalitet:

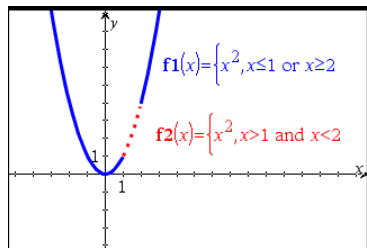
- Erstatninger
- Intervallbegrensninger
- Eksklusjoner

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.

$x^3-2\cdot x+7\rightarrow f(x)$	Done
$f(x) x=\sqrt{3}$	8.73205

Intervallbegrensninger tar form som en eller flere ulikheter som er føyd sammen av logiske “and” eller “or” operatorer. En intervallbegrensning tillater også forenkling som ellers kan være ugyldig eller ikke mulig å beregne.

$\text{nSolve}(x^3+2\cdot x^2-15\cdot x=0,x)$	0.
$\text{nSolve}(x^3+2\cdot x^2-15\cdot x=0,x) x>0 \text{ and } x<5$	3.



Eksklusjoner bruker relasjons-operatoren “ulik” (/= or ≠) for å ekskludere en spesifikk verdi fra å komme i betraktning.

→ (lagre)	ctrl	var	tast
<i>Verdi</i> → <i>Var</i>	$\frac{\pi}{4} \rightarrow myvar$ 0.785398		
<i>Liste</i> → <i>Var</i>	$2 \cdot \cos(x) \rightarrow y1(x)$ Done		
<i>Matrise</i> → <i>Var</i>	$\{1,2,3,4\} \rightarrow lst5$ { 1,2,3,4 }		
<i>Uttr</i> → <i>Funksjon</i> (<i>Param1</i> ,...)	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow matg$ $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$		
<i>Liste</i> → <i>Funksjon</i> (<i>Param1</i> ,...)	<i>"Hello"</i> → <i>str1</i> "Hello"		
<i>Matrise</i> → <i>Funksjon</i> (<i>Param1</i> ,...)			

Hvis variabel *Var* ikke eksisterer, opprettes *Var* og initialiserer den til *Verdi*, *Liste*, eller *Matrise*.

Hvis *Var* allerede eksisterer og ikke er låst eller beskyttet, erstattes innholdet med *Verdi*, *Liste*, eller *Matrise*.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive **= :** som en snarvei. Eksempel: Skriv **pi/4 = :**
minvar.

:= (tildele)	ctrl	={}	taster
<i>Var</i> := <i>Verdi</i>	$myvar := \frac{\pi}{4}$.785398		
<i>Var</i> := <i>Liste</i>	$y1(x) := 2 \cdot \cos(x)$ Done		
<i>Var</i> := <i>Matrise</i>	$lst5 := \{1,2,3,4\}$ { 1,2,3,4 }		
<i>Funksjon</i> (<i>Param1</i> ,...) := <i>Uttr</i>	$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$		
<i>Funksjon</i> (<i>Param1</i> ,...) := <i>Liste</i>	<i>str1</i> := "Hello" "Hello"		
<i>Funksjon</i> (<i>Param1</i> ,...) := <i>Matrise</i>			

Hvis variabel *Var* ikke eksisterer, opprettes *Var* og initialiserer den til *Verdi*, *Liste*, eller *Matrise*.

Hvis *Var* allerede eksisterer og ikke er låst eller beskyttet, erstattes innholdet med *Verdi*, *Liste*, eller *Matrise*.

© [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 produkhåndboken.

Define $g(n)$ =Func	
© Declare variables	
Local $i,result$	
$result:=0$	
For $i,1,n,1$ ©Loop n times	
$result:=result+i^2$	
EndFor	
Return $result$	
EndFunc	
	Done
$g(3)$	14

Ob, Oh

Ob binærTall

I desimalt grunntall-modus:	
Ob10+OhF+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:	
Ob10+OhF+10	0b11011

Resultatene vises i forhold til grunntall-modusen.

I heksades grunntall-modus:	
Ob10+OhF+10	0h1B

TI-Nspire™ CX II – Tegnekommandoer

Dette er et tilleggskdokument 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-grafregnere og TI-Nspire™-skrivebordsprogrammer for grafikkprogrammering.

TI-Nspire™ CX II-grafregnere 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 kladdemark.
- 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-grafregnavisningen. Dette betyr at det ikke er tilgjengelig i datamaskindokumentvisningen eller PublishView (.tnsp) 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

Grafikkskjerm bildet vil inneholde en overskrift øverst på skjermen som det ikke kan skrives i av grafikkkommandoer.

Tegningsområdet for grafikkskjerm bildet vil bli slettet (farge = 255, 255, 255) når grafikkskjerm bildet initialiseres.

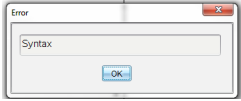
Grafikkskjerm	Standard
Høyde	212
Bredde	318
Farge	hvit: 255,255,255

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. Tegnede 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	Error Too few arguments The function or command is missing one or more arguments. 
Feil For mange argumenter	Funksjonen eller kommandoen inneholder for mange argumenter og kan ikke behandles.	Error Too many arguments The function or command contains an excessive number of arguments and cannot be evaluated. 
Feil Ugyldig datatype	Et argument er av feil datatype.	Error Invalid data type An argument is of the wrong data type. 

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

Katalog > 
CXII**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, bredde, høyde, startAngle, arcAngle*

Tegn en bue i det definerte avgrensede rektangelet med de angitte start- og buevinklene.

x, y: øvre venstre koordinat for avgrensede rektangel

bredde, høyde: dimensjoner av avgrensede 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](#)

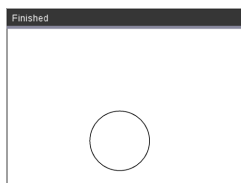
DrawCircle
 Katalog > 
CXII

DrawCircle *x, y, radius*

x, y: koordinat for sentrum

radius: radiusen på sirkelen

DrawCircle 150,150,40



Se også: [FillCircle](#)

DrawLine

Katalog > 
CXII

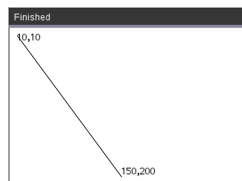
DrawLine $x1, y1, x2, y2$

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

Katalog > 
CXII

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.

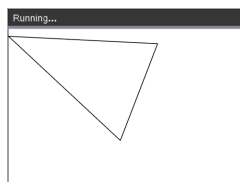
Uttrykk som evaluerer til en liste over reelle flyttall
 $xlist, ylist$

Uttrykk som evaluerer til ett enkelt reelt flyttall
 $x1, y1...xn, yn$ = koordinater for polygonhjørnene

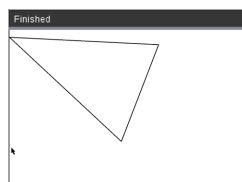
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

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 avgrensingsboks 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*

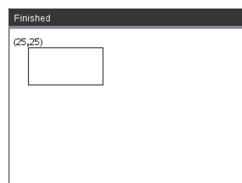
x, y: øvre venstre koordinat for rektangel

bredde, høyde: bredde og høyde for rektangel (rektangel tegn rett ned og til høyre fra startkoordinat).

Merk: Linjene tegnes i en avgrensingsboks 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

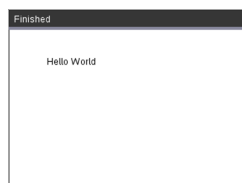
DrawText *x, y, exprOrString1*
[,exprOrString2]...

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*

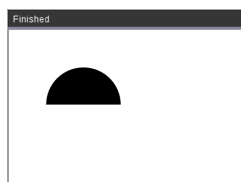
x, y: øvre venstre koordinat for avgrensede rektangel

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](#)-kommandoen

«arc angle» definerer vinkelåpningen for buen

FillArc 50,50,100,100,0,180

**FillCircle**
 Katalog >  CXII

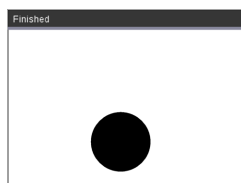
FillCircle *x, y, radius*

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](#)-kommandoen.

FillCircle 150,150,40



Her!

FillPoly
 Katalog >  CXII

FillPoly *xlist, ylist*

eller

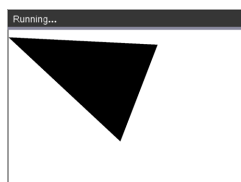
FillPoly *x1, y1, x2, y2, x3, y3...xn, yn*

Merk: Linjen og fargen er spesifisert av [SetColor](#) og [SetPen](#)

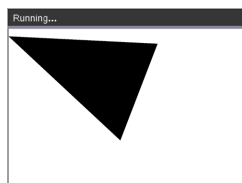
xlist:= {0,200,150,0}

ylist:= {10,20,150,10}

FillPoly *xlist,ylist*



FillPoly 0,10,200,20,150,150,0,10



FillRect

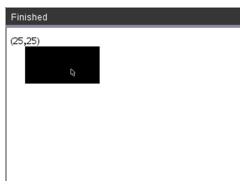
FillRect *x, y, bredde, høyde*

FillRect 25,25,100,50

x, y: øvre venstre koordinat for rektangel

bredde, høyde: bredde og høyde for rektangel

Tegn 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](#)

getPlatform()Katalog > 
CXII**getPlatform()**

getPlatform()

"dt"

Returnerer:

«dt» på skrivebordsprogrammer

«hh» på TI-Nspire™ CX-grafregnere

«ios» på TI-Nspire™ CX iPad®-app

PaintBuffer

Katalog > 
CXII

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

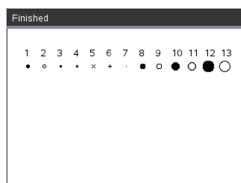
12 – største punkt, fast

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

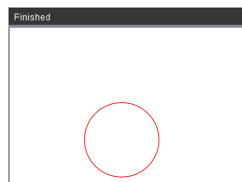
Red-value, Green-value, Blue-value

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

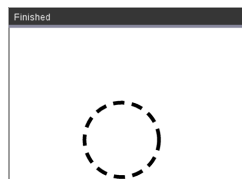
tykkelse: 1 <= tykkelse <= 3 | 1 er tynnest, 3 er tykkest

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

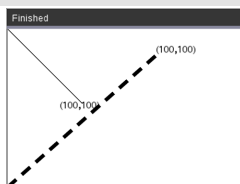
DrawLine 0,0,100,100

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

Katalog > 
CXII

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

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: [PaintBuffer](#)

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 40, og **isVoid()**, side 78.

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.

$\lfloor _ \rfloor$	–
$\gcd(100, _)$	–
$3 + _$	–
$\{5, _, 10\} - \{3, 6, 9\}$	$\{2, _, 1\}$

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**, **sumIf**, **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
$\text{cumulativeSum}(\{1, 2, _, 4, 5\})$	$\{1, 3, _, 7, 12\}$
$\text{cumulativeSum}\left(\begin{bmatrix} 1 & 2 \\ 3 & - \\ 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 2 \\ 4 & - \\ 9 & 8 \end{bmatrix}$

SortA og **SortD** flytter alle åpne (tomme) elementer innenfor det første utsagnet til bunnen.

$\{5, 4, 3, _, 1\} \rightarrow \text{list1}$	$\{5, 4, 3, _, 1\}$
$\{5, 4, 3, 2, 1\} \rightarrow \text{list2}$	$\{5, 4, 3, 2, 1\}$
$\text{SortA list1, list2}$	Done
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	$\{5,3,2,1,4\}$
$I1:=\{1,2,3,4,5\}; I2:=\{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
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	pi
θ	theta
∞	infinity
$\leq$	<=
$\geq$	>=
$\neq$	/=
$\Rightarrow$ (logisk implikasjon)	=>
$\Leftrightarrow$ (logisk dobbel implikasjon, XNOR)	<=>
$\rightarrow$ (lagre-operator)	=:
$ $ (absoluttverdi)	abs (...)
$\sqrt{()}$	sqrt (...)
$\Sigma()$ (Sum-sjablon)	sumSeq (...)
$\Pi()$ (Produkt-sjablon)	prodSeq (...)
$\sin^{-1}()$, $\cos^{-1}()$, ...	arcsin (...) , arccos (...) , ...
$\Delta\text{List}()$	deltaList (...)

Fra tastaturet på datamaskinen

Hvis du skal legge inn dette:	Skriv dette:
i (imaginær konstant)	@i
e (naturlig log-grunn tall e)	@e
E (vitenskapelig notasjon)	@E
T (transponert)	@t

Hvis du skal legge inn dette:	Skriv dette:
$^{\circ}$ (radianer)	@r
$^{\circ}$ (grader)	@d
$^{\circ}$ (gradianer)	@g
$\angle$ (vinkel)	@<
► (konvertering)	@>
►Decimal, ►approxFraction (), osv.	@>Decimal, @>approxFraction(), osv.

EOS™ (Ligningsoperativsystem)-hierarkiet

Dette avsnittet 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 (°), 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 halveis 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.

Ekspensiering

Ekspensiering (^) og element-for-element-ekspensiering (.^) blir behandlet fra høyre til venstre. Uttrykket 2^3^2 blir for eksempel behandlet som det samme som $2^{(3^2)}$ for å produsere 512. Dette er forskjellig fra $(2^3)^2$, som er 64.

Negasjon

For å legge inn et negativt tall, trykk på $\boxed{-}$ fulgt av tallet. Postoperasjoner og ekspensiering 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

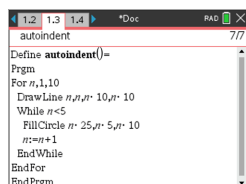
Auto-innrykk i Programmeringseditor

TI-Nspire™ programeditor gir automatisk innrykking av utsagn i en blokkommando.

Blokkommandoer er If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

Programeditor vil automatisk legge til mellomrom foran programkommandoer i en blokkommando. Den avsluttende kommandoen til blokken blir innrettet likt med åpningskommandoer.

Eksemplet nedenfor viser automatisk innrykk i nestede blokkommandoer.



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

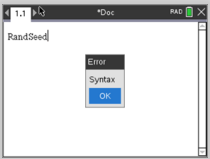
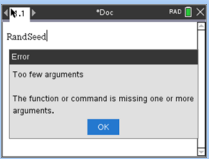
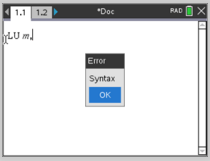
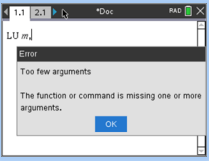
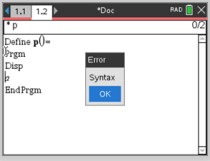
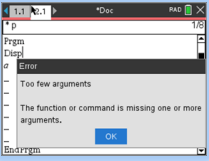
Feil

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
Mangler EndTry	Forventet EndTry , men fant en annen end-

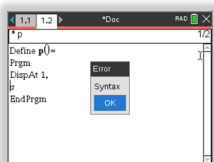
Feiltilstand	Ny melding
	setning
«Then» utelatt etter If <condition>	Mangler If..Then
«Then» utelatt etter Elself <condition>	Then mangler i blokken: Elself.
Når «Then», «Else» og «Elself» ble støttet på utenfor kontrollblokkene	Else er ugyldig utenfor blokkene: If..Then..Endif eller Try..EndTry
«Elself» vises utenfor «If..Then..Endif»-blokken	Elself er ugyldig utenfor blokk: If..Then..Endif
«Then» vises utenfor «If....Endif»-blokken	Then er ugyldig utenfor blokkene: If..Endif

Syntaksfeil

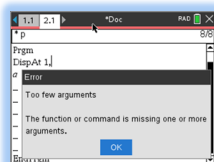
Hvis kommandoer som forventer ett eller flere argumenter blir oppkalt med en ufullstendig liste over argumenter, vil en «**For få argumenter-feil** » vises istedenfor for «**syntaks**»-feil

Gjeldende atferd	Ny CX II-atferd
 The screenshot shows the TI-84 Plus CE calculator interface. The command 'RandSeed' is entered. An error dialog box is displayed with the message 'Syntax' and an 'OK' button.	 The screenshot shows the TI-84 Plus CE calculator interface. The command 'RandSeed' is entered. An error dialog box is displayed with the message 'Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.
 The screenshot shows the TI-84 Plus CE calculator interface. The command 'LU m' is entered. An error dialog box is displayed with the message 'Syntax' and an 'OK' button.	 The screenshot shows the TI-84 Plus CE calculator interface. The command 'LU m' is entered. An error dialog box is displayed with the message 'Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.
 The screenshot shows the TI-84 Plus CE calculator interface. The command 'p' is entered. An error dialog box is displayed with the message 'Syntax' and an 'OK' button.	 The screenshot shows the TI-84 Plus CE calculator interface. The command 'p' is entered. An error dialog box is displayed with the message 'Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.

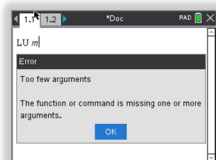
Gjeldende atferd



Ny CX II-atferd



Merk: Når en ufullstendig liste over argumenter ikke etterfølges av et komma, er feilmeldingen: «for få argumenter». Dette er det samme som i tidligere versjoner.



Konstanter og verdier

Den følgende tabellen inneholder konstanter og deres verdier, som er tilgjengelige når du utfører enhetsomregninger. 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	8987551787.3682 _m/_F
_Fc	Faraday-konstant	96485,33289 _coul/_mol
_g	Gravitasjonens akselerasjon	9,80665 _m/_s ²
_Gc	Gravitasjonskonstant	6,67408E-11 _m ³ /_kg/_s ²
_h	Plancks konstant	6,626070040E-34 _J _s
_k	Boltzmanns konstant	1,38064852E-23 _J/_°K
_μ0	Permeabilitet i vakuum	1,2566370614359E-6 _N/_A ²
_μb	Bohr-magneton	9,274009994E-24 _J _m ² /_Wb
_Me	Elektronets hvilemasse	9,10938356E-31 _kg
_Mμ	Myon-masse	1,883531594E-28 _kg
_Mn	Nøytronets hvilemasse	1,674927471E-27 _kg
_Mp	Protonets hvilemasse	1,672621898E-27 _kg
_Na	Avogadros tall	6,022140857E23 /_mol
_q	Elektronladning	1,6021766208E-19 _coul
_Rb	Bohr-radius	5,2917721067E-11 _m
_Rc	Molar gasskonstant	8,3144598 _J/_mol/_°K
_Rdb	Rydbergs konstant	10973731,568508/_m
_Re	Elektron-radius	2,8179403227E-15 _m
_u	Atommasse	1,660539040E-27 _kg
_Vm	Molarvolum	2,2413962E-2 _m ³ /_mol
_ε0	Permittivitet i vakuum	8,8541878176204E-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 163.

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: • 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	Avbryt Det ble trykket på tasten  eller  under en lang beregning eller mens et program ble utført.
190	Sirkulær definisjon Denne meldingen komme til syne for å unngå at du slipper opp for minne under uendelig erstatning av variable verdier. For eksempel vil $a+1 \rightarrow a$, der a er en udefinert variabel, forårsake denne feilen.
200	Ugyldig begrensingsuttrykk For eksempel vil $\text{løs}(3x^2-4=0,x) \mid x<0$ eller $x>5$ produsere denne feilmeldingen, fordi begrensningen er skilt med "eller" istedenfor "og".
210	Ugyldig datatype Et argument er av feil datatype.
220	Avhengig grense
230	Dimensjon 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.
235	Dimensjonsfeil. Ikke nok elementer i listene.
240	Dimensjonsfeil 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.
250	Divisjon med null
260	Grunnmengdefeil Et argument må være i en spesifisert grunnmengde. For eksempel er $\text{tilf}(0)$ ikke gyldig.
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 nSolve må være en ligning i én variabel. Det kan ikke inneholde noen annen variabel enn den variabelen som vi er interessert i.
320	Det første argumentet til løs eller kLøs må være en ligning eller ulikhet For eksempel er $\text{løs}(3x-4,x)$ ugyldig fordi det første argumentet ikke er en ligning.

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.
600	Ugyldig tabell
605	Ugyldig bruk av enheter

Feilkode	Beskrivelse
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 blokksyntaks
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.
830	Overflyt
850	Fant ikke programmet

Feilkode	Beskrivelse
	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	Reserverte 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: • <code>sto →</code> • <code>:=</code> • 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
1020	Intern feil
1030	Overtredelse av beskyttet minne

Feilkode	Beskrivelse
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: • 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: • Delen xxx kan bestå av mellom 1 og 16 tegn. • Delen yyy kan ha 1 til 15 tegn. Les mer om dette i Bibliotek-avsnittet dokumentasjonen.

Feilkode	Beskrivelse
1170	Ugyldig bruk av banenavn for bibliotek • En verdi kan ikke tildeles et banenavn som bruker Define, := eller sto ➔. • Et banenavn kan ikke erklæres som en lokal variabel eller brukes som parameter i en funksjonsdefinisjon eller programdefinisjon.
1180	Ugyldig variabelnavn på bibliotek Pass på at navnet: • Ikke inneholder punktum • Ikke begynner med senket strek • Ikke består av mer enn 15 tegn Les mer om dette i Bibliotek-avsnittet dokumentasjonen.
1190	Bibliotek-dokumentet ble ikke funnet: • Kontroller om biblioteket er i mappen MittBibl. • Oppdater biblioteker. Les mer om dette i Bibliotek-avsnittet dokumentasjonen.
1200	Bibliotek-variabler ble ikke funnet: • Kontroller om bibliotek-variablene eksisterer i den første oppgaven i biblioteket. • Forsikre deg om at bibliotek-variabelen er blitt definert som BiblOff eller BiblPriv. • Oppdater biblioteker. Les mer om dette i Bibliotek-avsnittet dokumentasjonen.
1210	Ugyldig navn på snarvei til bibliotek. Pass på at navnet: • Ikke inneholder punktum • Ikke begynner med senket strek • Ikke består av mer enn 16 tegn • Ikke er et reservert navn Les mer om dette i Bibliotek-avsnittet i dokumentasjonen.
1220	Grunnmengdefeil: Funksjonene tangentLinje og normalLinje støtter kun funksjoner med reelle verdier.
1230	Grunnmengdefeil. Trigonometriske omregningsoperatorer støttes ikke i Grader- eller Gradian-vinkelmodus.
1250	Argumentfeil

Feilkode	Beskrivelse
	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.

Varselkoder og meldinger

Du kan bruke funksjonen **warnCodes()** for å lagre varselkodene som ble generert da et uttrykk ble behandlet. Denne tabellen opplyster hver numeriske varselkode og dens assosierte melding.

Se **warnCodes()** på side 172 for et eksempel på lagring av varselkoder.

Varselkode	Melding
10000	Kommandoen kan gi falske løsninger.
10001	Derivasjon av en ligning kan gi en ugyldig ligning.
10002	Tvilsom løsning
10003	Tvilsom nøyaktighet
10004	Kommandoen kan utelate løsninger.
10005	KLøs kan spesifisere flere nullpunkter.
10006	Løs kan spesifisere flere nullpunkter.
10007	Flere løsninger kan eksistere. Prøv å angi passende øvre og nedre grenser, og/eller en gjetning. Eksempler ved bruk av solve(): • solve(Ligning, Var=Forslag) nedGrens<Var<øvgrens • solve(Ligning, Var) nedGrens<Var<øvgrens • solve(Ligning,Var=Forslag)
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
10015	1^∞ eller 1^undef erstattet av 1
10016	1^undef erstattet av 1
10017	Overflyt 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.

Varselkode	Melding
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	Tallstørrelsen er multiplisert med identitetsmatrisen.
10024	Resultat oppnådd ved bruk av tilnærmet aritmetikk.
10025	Ekvivalens kan ikke verifiseres i EXACT-modus.
10026	Avhengighet kan være ignorert. Spesifiser avhengighet i formen "\" 'Variabel MatteTestSymbol Konstant' eller en forbindelse av disse formene, for eksempel 'x<3 og 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 ditt land for informasjon om lengden og betingelsene i garantien eller om produktservice.

Begrenset garanti. Denne garantien påvirker ikke dine lovmessige rettigheter.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Stikkordregister

,	
', fremstilling minutter	197
-	
-, subtrahere[*]	180
!	
!, fakultet	190
"	
", fremstilling sekunder	197
#	
#, Indir.ref	195
#, omregnings-operator	223
%	
%, prosent	186
&	
&, legg til	190
*	
*, multiplisere	181
.	
., prikk subtraksjon	184
*, prikk multiplikasjon	184
./, prikk divisjon	185
^., prikk potens	185
+. , prikk addisjon	184
:	
:=, tildele	201
^	
^, potens	183

, begrensingsoperator	200
+	
+, addere	180
/	
/, dividere[*]	182
=	
≠, ulik[*]	187
=, er lik	186
>	
>, større enn	188
Π	
Π, produkt, sjablon for	5
Π, produkt[*]	192
Σ	
Σ(), sum[*]	193
ΣInt()	193
ΣPrn()	194
√	
√, kvadratrot[*]	192
∫	
∫, integral[*]	192
≤	
≤, mindre enn eller lik	188
≥	
≥, større enn eller lik med	189

►		Oh, heksadesimal indikator	202
► Cylind, vise som sylindrisk vektor		1	
[Sylind]	36		
► Polar, vise som polar vektor[Polar]	116	10^(), tier-potens	199
►, omregner til gradian vinkel[Grad]	70	2	
►approxFraction()	13		
►DD, vises som desimalvinkel[DD]	37	2-delers stykkevis funksjon	
►Desimal, vise resultat som desimal		sjablon for	2
[Decimal]	37	2-utvalg F test	59
►DMS, vise som		A	
grader/minutter/sekunder			
[DMS (GMS)]	44		
►Grunntall10, vise som desimalt		abs(), absoluttverdi	7
heltall[Grunntall10]	18	Absoluttverdi	
►Grunntall16, vise som heksadesimal		sjablon for	3-4
[Grunntall16]	18	addere, +	180
►Grunntall2, vise som binær		amortiseringstabell, amortTb()	7, 15
[Grunntall2]	16	amortTb(), amortiseringstabell	7, 15
►Rad, omregne til radian vinkel	123	and, Boolsk operator	8
►Rect, vise som rektangulær vektor	126	andrerderivert	
►Sphere, vise som sfærisk vektor		sjablon for	6
[Sfære (kule)]	150	angle(), vinkel	9
→		ANOVA, enveis varians-analyse	9
→, lagre	201	ANOVA2-veis, toveis varians-analyse	10
⇒		Ans, siste svar	12
⇒, logisk implikasjon[*]	189, 220	antall betingede elementer i en liste,	
⇔		tellIf()	31
⇔, logisk dobbel implikasjon[*]	190	antall elementer i en liste, antall()	31
©		antall(), antall elementer i en liste	31
©, kommentar	202	approx(), tilnærmet	12
°		approxRational()	13
°, grader fremstilling[*]	197	arccos()	13
°, grader/minutter/sekunder[*]	197	arccosh()	13
0		arccot()	13
Ob, binær indikator	202	arccoth()	13
		arccsc()	13
		arccsch()	14
		arcsec()	14
		arcsech()	14
		arcsin()	14
		arsinh()	14
		arctan()	14
		arctanh()	14

argumenter i TVM-funksjoner	167	ceiling(), øvre	19
augment(), utvid/sett sammen	14	centralDiff()	20
avgRC(), gjennomsnittlig endringshastighet	15	char(), tegnstreng	20
avrund, round()	135	χ^2 Cdf()	21
avslutt		χ^2 GOF	22
funksjon, EndFunc	60	ClearAZ	23
avslutt, Exit	50	colDim(), matrisens kolonnedimensjon	24
B		colNorm(), matrisens kolonnenorm	24
begrensningsoperator " "	200	conj(), kompleks konjugert	24
begrensningsoperator, rekkefølge av beregning	222	constructMat(), konstruer matrise	25
behandle polynom, polyEval()	116	$\cos^{-1}$, invers cosinus	27
behandling, rekkefølge av	222	cos(), cosinus	26
bestemt integral		cosh $^{-1}()$, hyperbolsk, invers cosinus	29
sjablon for	6	cosh(), hyperbolsk cosinus	28
bibliotek		cosinus, cos()	26
lage snarveier til objekter	80	$\cot^{-1}()$, invers cotangens	30
BibOff	39	cot(), cotangens	29
BibPriv	39	cotangens, cot()	29
binomCdf()	19, 76	coth $^{-1}()$, hyperbolsk invers cotangens	30
binomPdf()	19	coth(), hyperbolsk cotangens	30
binær		cPolyRoots()	32
indikator, Ob	202	crossP(), kryssprodukt	32
vise, ►Grunntall2	16	$\csc^{-1}()$, invers cosekans	33
Boolske operatører		csc(), cosekans	33
⇒	189, 220	csch $^{-1}()$, invers hyperbolsk cosekans	34
⇔	190	csch(), hyperbolsk cosekans	33
and	8	CubicReg, kubisk regresjon	34
eller	112	cumulativeSum(), kumulativ sum	35
enten ... eller	173	cycle, Løkke	35
ikke	108	Cycle, løkke	35
ikke både...og	102	D	
verken ... eller	106	d(), første deriverte	191
brøker		dager mellom datoer, dbd()	36
ekteBrøk	119	dbd(), dager mellom datoer	36
sjablon for	1	Define, definer	37
C		Definer	37
c 2 2-veis	21	Definer BibOff	39
c 2 Pdf()	22	Definer BibPriv	39
Cdf()	54	definer, Define	37

definere		ekte brøk, propFrac	119
felles (offentlig) funksjon eller		eliminasjonsform, ref()	127
program	39	eller (Boolsk), eller	112
privat funksjon eller program ..	39	eller, Boolsk operator	112
deltaList()	40	else if, ElseIf	47
DelVar, slett variabel	40	else, Else	71
delVoid(), fjern gamle elementer ...	40	ElseIf, else if	47
deriverte		en-variabel-statistikk, OneVar	111
første deriverte, d()	191	end	
numerisk derivert, nDeriv()	105	For...EndFor	56
numerisk derivert, nDerivative()	104	if, EndIf	71
desimal		stigningstall, EndLoop	92
heltall vise, 4Grunntall10	18	while, EndWhile	173
vinkel-visning, ►DD	37	end if, EndIf	71
det(), matrisedeterminant	41	end stigningstall, EndLoop	92
diag(), matrisediagonal	41	end while, EndWhile	173
dim(), dimensjon	42	endfunksjon, EndFunc	60
dimensjon, dim()	42	Endret internrente av retur, mirr(),	98
DispAt	42	EndTry, avslutt prøv	163
dividere heltall, intDiv()	74	EndWhile, end while	173
dividere, /	182	enhetsvektor, unitV()	169
dotP(), prikk produkt	45	enten ... eller ..., Boolsk eksklusiv eller	173
		EOS (Equation Operating System) ..	222
E		Equation Operating System	
e eksponent		(Ligningsoperativsystem)	
sjablon for	2	(EOS)	222
e i en potens, e^()	45, 51	er lik, =	186
E, eksponent	196	erstatning med " " operator	200
e^(), e i en potens	45	etikettNavn, Lbl	78
eff), omregn nominell til effektiv		euler(), Euler function	48
rente	46	Exit, avslutt	50
effektiv rente, eff()	46	exp(), e i en potens	51
egendefinerte funksjoner	37	expr(), streng til uttrykk	52
egendefinerte funksjoner og		ExpReg, eksponensiell regresjon	52
programmer	39		
egenvektor, eigVc()	46	F	
egenverdi, eigVl()	47	factor(), faktor	53
eigVc(), egenvektor	46	faktor, factor()	53
eigVl(), egenverdi	47	fakultet, !	190
ekskludering med " " operator	200	feil og problemløsning	
eksponensiell regresjon, ExpReg	52	send feil, SendFeil	114
eksponent, E	196	slett feil, SlettFeil	23
eksponenter		Fill, matrise fyller	54
sjablon for	1		

vis, 4Grunntall16	18	invers, x^{-1}	199
heltall, int()	74	invF()	75
heltallsdel, iPart()	77	invNorm(), invers kumulativ normalfordeling)	76
Hent	61, 212	invt()	77
hent/returner variabelinformasjon, getVarInfo(.....)	66, 69	Inv χ^2 ()	75
hurtigtaster	220	iPart(), heltallsdel	77
hvis, Hvis	71	irr(), internrente internrente, irr()	77
hyperbolic (hyperbolsk) invers tangens, $\tanh^{-1}()$	159	isPrime(), primtest	78
hyperbolsk cosinus, cosh()	28	isVoid(), test for tomrom	78
invers cosinus, $\cosh^{-1}()$	29	K	
invers sinus, $\sinh^{-1}()$	147	kolUtvid	24
sinus, sinh()	147	kombinasjoner, nCr()	103
tangens, tanh()	158	Kommandoen Wait	171
høyre(), høyre	132	kommentar, ©	202
høyre, høyre()	132	kompleks konjugert, conj()	24
høyre, right()	74	konstruer matrise, constructMat() ..	25
I		kopiere variabel eller funksjoner, CopyVar	25
identitet(), identitetsmatrise	71	korrelasjonsmatrise, corrMat()	26
identitetsmatrise, identitet()	71	korMat(), korrelasjonsmatrise	26
If, if	71	kryssprodukt, crossP()	32
iffn()	72	kubisk regresjon, CubicReg	34
ikke både...og, Boolsk operator	102	kumulativ sum, cumulativeSum() ..	35
ikke, Boolsk operator	108	kvadratisk regresjon, QuadReg	120
imag(), imaginær del	73	kvadratrot sjablon for	1
imaginær del, imag()	73	kvadratrot, v()	192
Indir.ref, #	195	kvadratrot, $\sharp()$	151
innenfor streng, inString()	73	L	
innstillinger, les aktuell	67	lagre symbol, &	201
inString(), innenfor streng	73	Lbl, etikettNavn	78
int(), heltall	74	lcm, minste felles multiplum	79
intDiv(), dividere heltall	74	left(), venstre	79
integral, f	192	legg til, &	190
interpolere(), interpolere	74	lengde på streng	42
invers cosinus, $\cos^{-1}()$	27	les modus, lesModus()	141
invers kumulativ normalfordeling (invNorm())	76		
invers sinus, sin / ()	146		
invers tangens, $\tan^{-1}()$	158		

les/returner		Local, lokal variabel	88
lesNevner, getDenom()	63	Logaritme	
lesTeller, getNum()	68	sjablon for	2
lesModus(), les modus	141	logaritmer	86
lesModus(), les modus-innstillinger	67	logaritmisk regresjon, LnReg	87
libShortcut(), lage snarveier til		logisk dobbel implikasjon, $\Leftrightarrow$	190
bibliotekobjekter	80	logisk implikasjon, $\Rightarrow$	189, 220
ligningssystemer (2-ligning)		Logistic, logistisk regresjon	90
sjablon for	3	LogisticD, logistisk regresjon	91
ligningssystemer (N-ligning)		logistisk regresjon, Logistic	90
sjablon for	3	logistisk regresjon, LogisticD	91
lineær regresjon, LinRegAx	81	lokal variabel, Local	88
lineær regresjon, LinRegBx	80, 82	lokal, Local	88
LinRegBx, lineær regresjon	80	Loop, stigningstall	92
LinRegMx, lineær regresjon	81	LU (lower-upper), matrisens nedre-	
LinRegIntervals, lineær regresjon	82	øvre dekomposisjon	93
LinRegTest	84	Lås, lås variabel eller variabelgruppe	89
linSolve()	85	låse opp variabler og variabelgrupper	169
Δ list(), differensliste	86	låse variabler og variabelgrupper	89
list $\blacktriangleright$ mat(), liste til matrise	86		
liste til matrise, list $\blacktriangleright$ mat()	86	M	
liste, antall betingede elementer i	31	maksimum, max()	94
liste, antall elementer i	31	mat $\blacktriangleright$ list(), matrise til liste	93
lister		matrices (matriser)	
differens, @liste()	86	undermatrise, subMat()	156
differenser i en liste, @list()	86	matrise (1 $\times$ 2)	
kryssprodukt, crossP()	32	sjablon for	4
kumulativ sum, cumulativeSum()	35	matrise (2 $\times$ 1)	
liste til matrise, list $\blacktriangleright$ mat()	86	sjablon for	4
maksimum, max()	94	matrise (2 $\times$ 2)	
matrise til liste, mat Δ liste()	93	sjablon for	4
midtstreng, mid()	97	matrise (m $\times$ n)	
minimum, min()	97	sjablon for	4
nye, newList()	104	matrise til liste, mat Δ liste()	93
prikk produkt, dotP()	45	matriser	
produkt, product()	119	determinant, det()	41
sorter fallende, SortD	150	diagonal, diag()	41
sorter stigende, SortA	149	dimensjon, dim()	42
summering, sum()	155	egenvektor, eigVc()	46
tomme elementer i	218	egenverdi, eigVl()	47
utvid/sett sammen, utvid()	14	eliminasjonsform, ref()	127
ln(), naturlig logaritme	86	fill, Fylle	54
LnReg, logaritmisk regresjon	87	identitet, identitet()	71

kolonnedimensjoner, colDim()	24	mindre enn eller lik, {	188
kolonnenorm, colNorm()	24	minimum, min()	97
kumulativ sum, cumulativeSum()	35	minste felles multiplum, lcm	79
liste til matrise, list→mat()	86	mirr(), endret internrente av retur	98
maksimum, max()	94	mod(), modul	98
matrise til liste, mat←liste()	93	modul, mod()	98
minimum, min()	97	modus-innstillinger, lesModus()	67
nedre-øvre dekomposisjon, LU (lower-upper)	93	moduser	
nye, newMat()	105	lesing, lesModus()	141
prikk addisjon, .+	184	mRow(), matrise radhandling	99
prikk divisjon, .P	185	mRowAdd(), matrise	
prikk multiplikasjon, *	184	radmultiplikasjon og	
prikk potens, .^	185	addisjon	99
prikk subtraksjon, .N	184	multiplisere, *	181
produkt, product()	119	Multiplpel lineær regresjon ttest	101
QR faktorisering, QR	120	MultReg	99
radaddisjon, rowAdd()	136	MultRegIntervals()	
raddimensjon, rowDim()	136	(MultRegIntervaller)	100
radhandling, mRow()	99	MultRegTests() (MultRegTester)	101
radmultiplikasjon og addisjon, mRowAdd()	99		
radnorm, rowNorm()	136	N	
radskift, rowSwap()	136	n-te rot	
redusert eliminasjonsform, rref()	136	sjablon for	1
summering, sum()	155	naturlig logaritme, ln()	86
tilfeldig, tilfMat()	125	nCr(), kombinasjoner	103
transponert, T	157	nDerivative(), numerisk derivert	104
undermatrise, subMat()	155	nedre, floor()	55
utvid/sett sammen, augment()	14	negasjon, legge inn negative tall	223
max(), maksimum	94	netto nåverdi, npv()	109
mean(), gjennomsnitt	94	newList(), ny liste	104
med (gitt at),	200	newMat(), ny matrise	105
median-median linjeregresjon, MedMed	95	nfMax(), numerisk	
median(), median	95	funksjonsmaksimum	105
median, median()	95	nfMin(), numerisk	
MedMed, median-median linjeregresjon	95	funksjonsminimum	105
mid(), midtstreng	97	nInt(), numerisk integral	106
midtstreng, mid()	97	nom), omregn effektiv til nominell rente	106
min(), minimum	97	nominell rente, nom()	106
		norm(), Frobenius-norm	107
		normCdf()	107
		normPdf()	108
		nPr(), permutasjoner	109
		npv(), netto nåverdi	109

nSolve(), numerisk løsning	110	potens, ^	183
numerisk		potensregresjon,	
derivert, nDeriv()	105	PowerReg	117, 129, 131, 160
derivert, nDerivative()	104	PowerReg, potensregresjon	117
integral, nInt()	106	Prgm, definer program	118
løsning, nSolve()	110	prikkP	
ny		addere, +	184
liste, newList()	104	divisjon, /	185
matrise, newMat()	105	multiplikasjon, *	184
når, when()	172	potens, ^	185
		produkt, dotP()	45
O		subtraksjon, -	184
objekter		primtallstest, isPrime()	78
lage snarveier til bibliotek	80	prodSeq()	119
omregne		product(), produkt	119
►Rad	123	produkt, P()	192
4Grad	70	produkt, product()	119
omregnings-operator (#)	223	programmer	
OneVar, en-variabel-statistikk	111	definere felles (offentlig) bibliotek	39
operatorer		definere privat bibliotek	39
rekkefølge av behandling	222	programmer og programmering	
ord(), numerisk tegnkode	113	slett feil, SlettFeil	23
Overfører øyeblikkelig kontroll til den		try, Try	163
neste iterasjonen i aktuell		vis I/O-skjerm, Vis	138
løkke (For, While, eller		vis I/O skjerm, Vis	42
Loop).	36	programmering	
		definer program, Prgm	118
P		send feil, SendFeil	114
P►Rx(), rektangulær x-koordinat ...	113	vis data, Vis	42, 138
P►Ry(), rektangulær y-koordinat ...	114	propFrac, ekte brøk	119
Pdf()	57	prosent, %	186
permutasjoner, nPr()	109		
poissCdf()	115	Q	
poissPdf()	115	QR faktorisering, QR	120
polar		QR, QR faktorisering	120
koordinat, R►Pr()	123	QuadReg, kvadratisk regresjon	120
koordinat, R►Pθ()	123	QuartReg, fjerdegrads regresjon ...	121
vektor-visning, ►Polar	116		
polyEval(), behandle polynom	116	R	
polynomer		R, radian	196
behandle, polyEval()	116	R►Pr(), polarkoordinat	123
tilfeldig, tilfPoly()	125	R►Pθ(), polarkoordinat	123
PolyRoots()	117	radian, R	196

RandSeed, tilfeldig startverdi	126	S	
real(), reell	126		
redusert eliminasjonsform, rref() ..	136	sannsynlig normalfordeling,	
reell, real()	126	normCdf()	107
ref(), eliminasjonsform	127	sannsynlig student-t-fordeling, tCdf()	159
RefreshProbeVars	128	sannsynlighetstetthet for student-t,	
regresjon		tPdf()	162
fjerdegrads, QuartReg	121	sannsynlighetstetthet for student-t,	
kvadratisk, QuadReg	120	tPdf()	162
lineær regresjon, LinRegAx	81	sannsynlighetstetthet, normPdf() ..	108
lineær regresjon, LinRegBx	80	sec ⁻¹ (), invers sekans	137
logaritmisk, LnReg	87	sec(), sekans	137
Logistisk	90	sech ⁻¹ (), invers hyperbolsk sekans ..	138
logistisk, Logistic	91	sech(), hyperbolsk secant	138
median-median-linje, MedMed ..	95	sekvens, seq()	139
MultReg	99	send feil, SendFeil	114
potensregresjon, PowerReg	117	SendFeil, send feil	114
sinus, SinReg	148	seq(), sekvens	139
Regresjon		seqGen()	140
lineær regresjon, LinRegBx	82	seqn()	140
regresjoner		sequence, seq()	140
eksponensiell, ExpReg	52	sfærisk vektor-visning, ►Sphere	150
kubisk, CubicReg	34	shift(), skift	143
potensregresjon,		sign(), fortegn	144
PowerReg . 117, 129, 131, 160		simult(), simultane ligninger	144
rektangulær vektor-visning, ►Rect ..	126	simultane ligninger, simult()	144
rektangulær x-koordinat, P►Rx() ...	113	sin ⁻¹ (), invers sinus	146
remain(), rest	129	sin(), sinus	146
resiprok, x ⁻¹	199	sinh ⁻¹ (), hyperbolsk invers sinus	147
rest, remain()	129	sinh(), hyperbolsk sinus	147
resultat verdier, statistikker	152	SinReg, sinusregresjon	148
resultater, statistikk	151	sinus, sin()	146
Return, retur	132	sinusregresjon, SinReg	148
right, right()	48, 172	sjabloner	
rk23(), Runge Kutta-funksjon	132	(Π), produkt	5
rotate(), rotere	134	2-delers stykkevis funksjon	2
rotere, rotate()	134	Absoluttverdi	3-4
round(), avrund	135	andredrivert	6
rowAdd(), radaddisjon i matrise	136	bestemt integral	6
rowDim(), raddimensjon i matrise ..	136	brøk	1
rowNorm(), radnorm i matrise	136	eksponent	2
rowSwap(), radskift i matrise	136	eksponent	1
rref(), redusert eliminasjonsform ..	136	første deriverte	5

kvadratrot	1	varians, variance()	170
ligningssystemer (2-ligning)	3	stdDevPop(), populasjonens standardavvik	152
ligningssystemer (N-ligning)	3	stdDevSamp(), utvalgets standardavvik	153
Logaritme	2	stigningstall, Loop	92
matrise (1 × 2)	4	Stoppkommando	154
matrise (2 × 1)	4	streng	
matrise (2 × 2)	4	dimensjon, dim()	42
matrise (m × n)	4	lengde	42
n-te rot	1	strenger	
stykkevis funksjon (N-delers) ...	2	brukes for å opprette	
sum (G)	5	variabelnavn	223
skift, shift()	143	format, format()	56
slett		formatering	56
feil, SlettFeil	23	høyre, høyre()	74, 132
åpne elementer fra liste	40	Indir.ref, #	195
Slett	206	innenfor, inString	73
slettAZ	23	legg til, &	190
slette		midtstreng, mid()	97
variabel, DelVar	40	rotere, rotate()	134
SlettFeil, slett feil	23	skift, shift()	143
snarveier, tastatur	220	streng til uttrykk, expr()	52
SortA, sorter stigende	149	tegnkode, ord()	113
SortD, sorter fallende	150	tegnstreng, char()	20
sorterer		uttrykk til streng, string()	154
fallende, SortD	150	venstre, left()	79
stigende, SortA	149	string(), uttrykk til streng	154
språk		strings	
hente språkinformasjon	66	right, right()	48, 172
sqrt(), kvadratrot	151	stykkevis funksjon (N-delers)	
standardavvik, stdDev()	152-153, 170	sjablon for	2
stat.resultater	151	stykkevis()	115
stat.verdier	152	større enn eller lik med, 	189
statistikk		større enn, >	188
en-variabel-statistikk, OneVar ..	111	største felles divisor, gcd()	60
fakultet, !	190	subMat(), undermatrise	155-156
gjennomsnitt, mean()	94	subtrahere, -	180
kombinasjoner, nCr()	103	sum (G)	
median, median()	95	sjablon for	5
permutasjoner, nPr()	109	sum av hovedbetalinger	194
standardavvik, stdDev()	152-153, 170	sum av rentebetalinger	193
tilfeldig startverdi, RandSeed ..	126	sum(), summering	155
tilfeldig norm, tilfNorm()	125	sum, Σ()	193
to-variable resultater, TwoVar ..	167		

sumIf()	155	tilMat(), tilfeldig matrise	125
summering, sum()	155	tilnærmet, approx()	12
sumSeq()	156	tInterval, t konfidensintervall	160
svar (siste), Ans	12	tInterval_2Samp, toutvalg t konfidensintervall	161
T		to-variable resultater, TwoVar	167
T, transponert	157	tomme (åpne) elementer	218
t test, tTest	164	tomrom, test for	78
$\tan^{-1}()$, invers tangens	158	trace()	162
$\tan()$, tangens	157	transponert, T	157
tangens, $\tan()$	157	Try, feil håndteringskommando	163
$\tanh^{-1}()$, hyperbolsk invers tangens	159	try, Try	163
$\tanh()$, hyperbolsk tangens	158	Try, try	163
tCdf(), sannsynlig student t-fordeling	159	tTest, t test	164
tegn		tTest_2Samp, to-utvalgs t-test	164
streng, char()	20	TVM-argumenter	167
tegnkode, ord()	113	tvmFV()	165
tegne	207-209	tvmI()	166
tegnstreng, char()	20	tvmN()	166
Tekstkommando	160	tvmPmt()	166
teller dager mellom datoer, dbd()	36	tvmPV()	167
tellIf(), antall betingede elementer i en liste	31	TwoVar, to-variable resultater	167
test for tomrom, isVoid()	78	U	
Test_2S, 2-utvalg F test	59	ulik, $\neq$	187
tidsverdi for penger, antall betalinger	166	undermatrise, subMat()	155-156
tidsverdi for penger, betalingsbeløp	166	unitV(), enhetsvektor	169
tidsverdi for penger, Fremtidig verdi	165	unLock, lås opp variabel eller variabelgruppe	169
tidsverdi for penger, nåverdi	167	uttrykk	
tidsverdi for penger, Rente	166	streng til uttrykk, expr()	52
tier-potens, $10^{\wedge}()$	199	utvid/sett sammen, augment()	14
tilf(), tilfeldig nummer	124	V	
tilfBin, tilfeldig tall	124	variabel	
tilfeldig		opprette navn fra en tegnstreng	223
matrise, tilfMat()	125	variabler	
norm, tilfNorm()	125	lokal, Local	88
polynom, tilfPoly()	125	slette alle enkelttegn	23
startverdi, RandSeed	126	slette, DelVar	40
tilfeldig utvalg	125	variabler og funksjoner	
tilfInt(), tilfeldig heltall	124	kopiere	25
tilfNorm(), tilfeldig norm	125	variabler, låse og låse opp	67, 89, 169
tilfPoly(), tilfeldig polynom	125		
tilfSamp()	125		

varians, variance()	170	zInterval_1Prop, en-proporsjons z konfidensintervall	175
varPop()	170	zInterval_2Prop, to-proporsjons z konfidensintervall	175
varSamp(), utvalgets varians	170	zInterval_2Samp, to-utvalgs z konfidensintervall	176
varselkoder og meldinger	236	zTest	177
vektorer		zTest_1Prop, en-proporsjons z-test ..	177
enhet, unitV()	169	zTest_2Prop, to-proporsjons z-test ..	178
kryssprodukt, crossP()	32	zTest_2Samp, to-utvalgs z-test	179
Overfører øyeblikkelig kontroll til den neste iterasjonen i aktuell løkke (For, While, eller Loop).	36		
prikk produkt, dotP()	45	Ø	
venstre, left()	79	øvre, ceiling()	19-20, 32
verken ... eller, Boolsk operator	106		
vinkel, angle()	9	Å	
vis data, Vis	42, 138	åpne (tomme) elementer	218
Vis, vis data	42, 138	åpne elementer, fjern	40
vise som			
binær, 4Grunntall2	16		
desimalt heltall, 4Grunntall10 ..	18		
desimalvinkel, ►DD	37		
grader/minutter/sekunder, ►DMS	44		
heksadesimal, 4Grunntall16	18		
polar vektor, ►Polar	116		
rektangulær vektor, ►Rect	126		
sfærisk vektor, ►Sphere	150		
sylindrisk vektor, ►Cylind	36		
W			
warnCodes(), Warning codes	172		
when(), når	172		
while, While	173		
While, while	173		
X			
x^{-1} , resiprok	199		
x^2 , kvadrat	183		
XNOR	190		
Z			
zInterval, z konfidensintervall	174		