TI-Nspire™
Referanseguide

Finn ut mer om TI Technology via online hjelp på education.ti.com/eguide.

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.

© 2006 - 2019 Texas Instruments Incorporated

Innhold

Uttrykkssjabloner **1**

Alfabetisk oversikt **7**

A 7
B 15
C 19
D 36
E 44
F 52
G 59
I 69
L 77
M 92
N 101
O 110
P 112
Q 119
R 122
S 136
T 156
U 168
V 169
W 170
X 172
Z 173

Symboler **179**

Tomme (åpne) elementer **202**

Snarveier/hurtigtaster for å legge inn matematiske uttrykk **204**

EOS™ (Ligningsoperativsystem)-hierarkiet **206**

Konstanter og verdier **208**

Feilkoder og feilmeldinger **209**

Varselkoder og meldinger **217**

Generell informasjon **219**

Hjelp på nettet (online) 219

Kontakt TIs brukerstøtte 219
Service og garantiinformasjoner 219

Stikkordregister 220

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

$\frac{\square}{\square}$

Merk: Se også **/ (divider)**, side 181.

Eksempel:

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

Eksponent-sjablon

[^] tast

$\square^{\square}$

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 182.

Eksempel:

2^3	8

Kvadratrot-sjablon

[ctrl] [x²] taster

$\sqrt{\square}$ **Merk:** Se også **√() (kvadratrot)**, side 191.

Eksempel:

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

N-te rot-sjablon

[ctrl] [^] taster

$\square$

$\sqrt[\square]{\square}$ **Merk:** Se også **rot()**, side 133.

Eksempel:

N-te rot-sjablon

ctrl ^ taster

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

e eksponent-sjablon

$e^{\square}$

Naturlig grunntall *e* opphøyd i en eksponent

Merk: Se også **e^()**, side 44.

e^x tast

e^{1}	2.71828182846

Logaritme-sjablon

ctrl 10^x taster

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

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

Merk: Se også **log()**, side 88.

Eksempel:

$\log_{4}(2.)$	0.5

Stykkevis sjablon (2-delers)

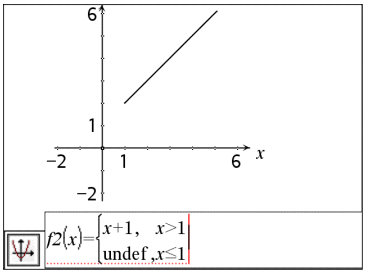
Katalog >

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

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 114.

Eksempel:



Stykkevis sjablon (N-delers)

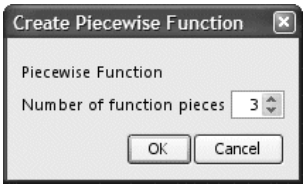
Katalog >

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 114.

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 155.

Eksempel:

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

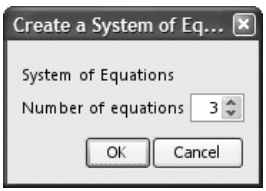
$$\text{solve}\left(\begin{cases} y=x^2-2 \\ x+2\cdot y=-1 \end{cases}, x, y\right)$$

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

Sjablon for ligningssystemer med N ukjente

Katalog >

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



Merk: Se også **system()**, side 155.

Eksempel:

Se eksemplet for Sjabloner for ligningssystemer (2 ligninger).

Sjablon for absoluttverdi

Katalog >

Merk: Se også **abs()**, side 7.

Eksempel:

Sjablon for absoluttverdi

Katalog >

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

gg°mm'ss.ss'' sjablon

Katalog >

□°□'□"

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.

Eksempel:

$$30°15'10" \qquad 0.528011$$

Matrise-sjablon (2 x 2)

Katalog >

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

Oppretter en 2 x 2-matrise.

Eksempel:

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

Matrise-sjablon (1 x 2)

Katalog >

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

Eksempel:

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

Matrise-sjablon (2 x 1)

Katalog >

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

Eksempel:

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

Matrise-sjablon (m x n)

Katalog >

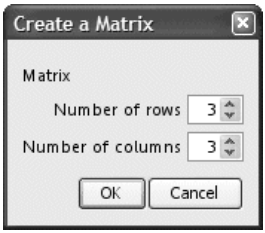
Sjablonen kommer til syne etter at du er blitt bedt om å spesifisere antallet rader og kolonner.

Eksempel:

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

Matrise-sjablon (m x n)

Katalog >



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)

Katalog >

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

Merk: Se også **Σ() (sumSeq)**, side 192.

Eksempel:

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

Produkt-sjablon (Π)

Katalog >

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

Merk: Se også **Π() (prodSeq)**, side 191.

Eksempel:

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

Første derivert-sjablon

Katalog>

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

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

Eksempel:

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

Første derivert-sjablon

Katalog>

Merk: Se også **d() (derivert)**, side 190.

Andre derivert-sjablon

Katalog>

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

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 190.

Eksempel:

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

Bestemt integral-sjablon

Katalog >

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

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 105.

Eksempel:

$$\int_0^{10} x^2 \, dx \qquad 333.333$$

Alfabetisk oversikt

Elementer med navn som ikke er alfabetiske (som f.eks. +, !, og >) er opplistet på slutten av dette avsnittet (side 179). 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(*Verdi1***)⇒***verdi*

abs(*Liste1***)⇒***liste*

abs(*Matrise1***)⇒***matrise*

Returnerer argumentets absoluttverdi.

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

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

Merk: Alle ubestemte variabler behandles som reelle variabler.

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

$$|2-3\cdot i| \qquad 3.60555$$

amortTbl() Katalog >

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

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

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

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 166.

- 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

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

amortTbl() Katalog >

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 **ΣInt ()** og **ΣPrn()**, side 192, og **bal()**, side 15.

and Katalog >

BoolskUttr1 and *BoolskUttr2* ⇒*Boolsk uttrykk*

Boolsk liste1 and *Boolsk liste2*⇒*Boolsk liste*

Boolsk matrise1 and *Boolsk matrise2*⇒*Boolsk matrise*

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

Heltall1 **and** *Heltall2*⇒*heltall*

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

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

I heksades grunntall-modus:

0h7AC36 and 0h3D5F	0h2C16

Viktig: Null, ikke bokstaven O.

I binær grunntall-modus:

0b100101 and 0b100	0b100

I desimalt grunntall-modus:

37 and 0b100	4

and

Katalog >

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

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

angle() vinkel

Katalog >

angle(*Verdi1***)⇒***verdi*

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

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

angle(*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.

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

ANOVA

Katalog >

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

Utfører en enveis analyse av varians for å sammenlikne gjennomsnitt for mellom 2 og 20 populasjoner. En oversikt over resultatene lagres i *stat.results*-variabelen (side 150).

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 150).

LevRad=0 for Blokk

LevRad=2,3,...,*Len*-1, for To Faktor, hvor *Len*=lengde(*Liste1*)=lengde(*Liste2*) = ... = lengde(*Liste10*) og *Len* / *LevRad* ∈ {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 kolonnefaktoren
stat.PValRow	Kolonnefaktorens sannsynlighetsverdi
stat.dfRow	Grader frihet for radfaktoren
stat.SSRow	Sum av kvadrater for radfaktoren
stat.MSRow	Gjennomsnitt av kvadrater for radfaktor

INTERAKSJON Utdata

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

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

FEIL Utdata

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

Ans (svar)

ctrl (–) taster

Ans⇒*verdi*

Returnerer resultatet av det sist behandlede uttrykket.

56	56
56+4	60
60+4	64

approx() (tilnærm)

Katalog >

approx(*Verdi1***)**⇒*verdi*

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

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

$\text{approx}\left(\frac{1}{3}\right)$	0.333333
$\text{approx}\left(\left\{\frac{1}{3},\frac{1}{9}\right\}\right)$	{0.333333,0.111111}
approx({sin(π),cos(π)})	{0.,-1.}
approx([√2 √3])	[1.41421 1.73205]
$\text{approx}\left(\left[\frac{1}{3}\ \ \frac{1}{9}\right]\right)$	[0.333333 0.111111]

approx(*Liste1***)**⇒*liste*

approx(*Matrise1***)**⇒*matrise*

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

approx({sin(π),cos(π)})	{0.,-1.}
approx([√2 √3])	[1.41421 1.73205]

▸approxFraction() Katalog >

Verdi ▸*approxFraction*([*Tol*])⇒*verdi*

Liste ▸**approxFraction**([*Tol*])⇒*liste*

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

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

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

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

$$0.83333333333333 \blacktriangleright \text{approxFraction}(5.\text{E}^{-}14) \qquad \frac{5}{6}$$

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

approxRational() Katalog >

approxRational(*Verdi*[, *Tol*]**)**⇒*verdi*

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

approxRational(*Matrise*[, *Tol*]**)**⇒*matrise*

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

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

$$\text{approxRational}(\{0.2,0.33,4.125\},5.\text{E}^{-}14) \qquad \left\{\frac{1}{5},\frac{33}{100},\frac{33}{8}\right\}$$

arccos() Se cos⁻¹(), side 27.

arccosh() Se cosh⁻¹(), side 28.

arccot() Se cot⁻¹(), side 29.

arccoth() Se coth⁻¹(), side 30.

arccsc() Se csc⁻¹(), side 32.

arccsch() **Se csch^{-1}(), side 33.**

arcsec() **Se sec^{-1}(), side 136.**

arcsech() **Se sech^{-1}(), side 137.**

arcsin() **Se sin^{-1}(), side 145.**

arcsinh() **Se sinh^{-1}(), side 146.**

arctan() **Se tan^{-1}(), side 157.**

arctanh() **Se tanh^{-1}(), side 158.**

() (utvid/sett sammen) Katalog >

augment(*Liste1, Liste2***)⇒***liste*

Returnerer en ny liste som er *Liste2* lagt til på slutten av *Liste1*.

augment({1,-3,2},{5,4}) {1,-3,2,5,4}

augment(*Matrise1***,** *Matrise2***)⇒***matrise*

Returnerer en ny matrise som er *Matrise2* lagt til på *Matrise1*. Når tegnet "," brukes, må matrisen ha like raddimensjoner, og *Matrise2* er lagt til på *Matrise1* som nye kolonner. Endrer ikke *Matrise1* eller *Matrise2*.

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

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

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

avgRC() (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 forhåndstildelt verdi eller aktuell "|" erstatning for variabelen.

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

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

$x:=2$ → 2

$\text{avgRC}(x^2-x+2,x)$ → 3.001

$\text{avgRC}(x^2-x+2,x,.1)$ → 3.1

$\text{avgRC}(x^2-x+2,x,3)$ → 6

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 166.

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

$\text{bal}(5,6,5.75,5000,,12,12)$ → 833.11

$tbl:=\text{amortTbl}(6,6,5.75,5000,,12,12)$

$$\begin{bmatrix} 0 & 0. & 0. & 5000. \\ 1 & -23.35 & -825.63 & 4174.37 \\ 2 & -19.49 & -829.49 & 3344.88 \\ 3 & -15.62 & -833.36 & 2511.52 \\ 4 & -11.73 & -837.25 & 1674.27 \\ 5 & -7.82 & -841.16 & 833.11 \\ 6 & -3.89 & -845.09 & -11.98 \end{bmatrix}$$

$\text{bal}(4,tbl)$ → 1674.27

bal() Katalog >

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 166.

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

avrundVerdi spesifiserer antallet desimalplasser for avrunding. Grunninnstilling=2.

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

Merk: Se også **ΣInt()** og **ΣPrn()**, side 192.

▸Base2 (Grunntall2) Katalog >

Heltall1 **▸Base2**⇒*heltall*

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.

256▸Base2	0b100000000
0h1F▸Base2	0b11111

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

▶Base10 (Grunntall10) Katalog >

Heltall1 ▶**Base10**⇒*heltall*

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.

0b10011▶Base10	19
0h1F▶Base10	31

▶Base16 (Grunntall16) Katalog >

Heltall1 ▶**Base16**⇒*heltall*

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 desimalt tall (grunntall 10). Resultatet vises i heksadesimal, uavhengig av grunntallets modus.

256▶Base16	0h100
0b111100001111▶Base16	0hF0F

▶Base16 (Grunntall16) Katalog >

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

binomCdf() katalog >

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≤X ≤*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 ≤ *øvreGrense*), sett *nedreGrense*=0

binomPdf() Katalog >

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) Katalog >

ceiling(*Verdi1***)**⇒*verdi*

Returnerer det nærmeste heltallet som er ≥ argumentet.

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

ceiling() (øvre) Katalog >

Argumentet kan være et reelt eller et komplekst tall.

Merk: Se også **floor()** (nedre).

ceiling(*Liste1***)⇒***liste*

ceiling(*Matrice1***)⇒***matrice*

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

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

centralDiff() Katalog >

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

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

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

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

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

Returnererden numeriske deriverte ved hjelp av derivasjonskvotient-formelen.

Hvis *verdi* er spesifisert, opphever den eventuell forhåndstildelt 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.

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

char() Katalog >

char(*Heltall***)⇒***tegn*

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

char() Katalog >

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

$\chi^2$2way Katalog >

χ^2**2way** *ObsMatrise*

chi22way *ObsMatrise*

Beregner en χ^2 test for samling av "tellinger" på toveis-tabellen i den observerte matrisen *ObsMatrise*. En oversikt over resultatene lagres i *stat.results*-variabelen (side 150).

For informasjon om effekten av tomme elementer i en matrise, se "Tomme (åpne) elementer" (side 202).

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

χ^2Cdf() Katalog >

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

chi2Cdf(*nedreGrense*,*øvreGrense*,*df***)**⇒*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*.

χ^2Cdf() Katalog >

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

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

χ^2GOF Katalog >

χ^2**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 150).

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

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

χ^2Pdf() Katalog >

χ^2**Pdf(***XVerd*,*df***)**$\Rightarrow$*tall* hvis *XVerd* er et tall, *liste* hvis *XVerd* er en liste

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

Beregner sannsynlighetstettheten (pdf) for c^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 202).

ClearAZ (slettAZ) Katalog >

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 168.**

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

ClrErr (SlettFeil) Katalog >

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 113, og **Try**, side 162.

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

For et eksempel på **ClrErr**, se eksempel 2 under **Try** -kommandoen, side 162.

colAugment() (kolUtvid) Katalog >

colAugment(*Matrise1***,** *Matrise2***)⇒***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*.

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

colDim() Katalog >

colDim(*Matrise***)⇒***uttrykk*

Returnerer antallet kolonner som ligger i *Matrise*.

Merk: Se også **radDim()**.

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

colNorm() Katalog >

colNorm(*Matrise***)⇒***uttrykk*

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

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

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

$\text{colNorm}(mat)$ 9

conj() Katalog >

conj(*Verdi1***)⇒***verdi*

conj(*Liste1***)⇒***liste*

conj(*Matrise1***)⇒***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() katalog >

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*.

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

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.

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

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.

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

$aa.a:=45$	45
$aa.b:=6.78$	6.78
CopyVar $aa.,bb.$	*Done*
getVarInfo()	$\begin{bmatrix} aa.a & \text{"NUM"} & \text{"}\square\text{"} & 0 \\ aa.b & \text{"NUM"} & \text{"}\square\text{"} & 0 \\ bb.a & \text{"NUM"} & \text{"}\square\text{"} & 0 \\ bb.b & \text{"NUM"} & \text{"}\square\text{"} & 0 \end{bmatrix}$

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 gruper, og legge til de medlemmene som ikke allerede finnes. Hvis en enkel (ikke i gruppe) variabel med navnet *Var2* finnes, oppstår det en feil.

corrMat() Katalog >

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

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

cos() trig tast

cos(*Verdi1***)⇒***verdi*

cos(*Liste1***)⇒***liste*

cos(*Verdi1***)** returnerer cosinus til argumentet som en verdi.

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

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

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

cos(*kvadratMatrise1***)⇒***kvadratMatrise*

Returnerer matrisens cosinus til *kvadratMatrise1*. Dette er ikke det samme som å beregne cosinus til hvert element.

Når en skalarfunksjon f(A) virker på *kvadratMatrise1* (A), beregnes resultatet av algoritmen:

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

kvadratMatrise1 må kunne diagonaliseres. Den kan heller ikke ha symbolske variabler som ikke er tildelt noen verdi.

Utform matrisene:

I Radian-vinkelmodus:

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

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

cos() trig tast

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

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.

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

**$\cos^{-1}$(*Verdi1*)⇒*verdi*

**$\cos^{-1}$(*Liste1*)⇒*liste*

$\cos^{-1}$(*Verdi1*) returnerer vinkelen som har cosinus lik *Verdi1*.

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

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

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

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

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

$\{1.5708, 1.36944, 1.0472\}$

I radian-vinkelmodus og rektangulært, kompleks format:

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

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

$\cos^{-1}()$ — trig tast

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

cosh() — Katalog >

cosh(*Verdi1***)⇒***verdi*

cosh(*Liste1***)⇒***liste*

cosh(*Verdi1***)** returnerer hyperbolsk cosinus til argumentet.

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

I Grader-vinkelmodus:

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

cosh(*kvadratMatrise1***)⇒***kvadratMatrise*

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

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

$$\begin{bmatrix}421.255 & 253.909 & 216.905\\ 327.635 & 255.301 & 202.958\\ 226.297 & 216.623 & 167.628\end{bmatrix}$$

$\cosh^{-1}()$ — Katalog >

cosh⁻¹(*Verdi1***)⇒***verdi*

cosh⁻¹(*Liste1***)⇒***liste*

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

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

cosh⁻¹(*Verdi1***)** returnerer invers hyperbolsk cosinus til argumentet.

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

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

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

I radian-vinkelmodus og rektangulært, kompleks format:

$\cosh^{-1}()$ Katalog >

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

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

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

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

cot() trig tast

cot(*Verdi1***)** ⇒ *verdi*

cot(*Liste1***)** ⇒ *liste*

Returnerer cotangens av *Verdi1* eller returnerer en liste med cotangens til alle elementene i *Liste1*.

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 Grader-vinkelmodus:

$\cot(45)$ 1.

I Gradian-vinkelmodus:

$\cot(50)$ 1.

I Radian-vinkelmodus:

$\cot(\{1,2.1,3\})$

$\{0.642093,-0.584848,-7.01525\}$

$\cot^{-1}()$ trig tast

cot $^{-1}$**(***Verdi1***)** ⇒*verdi*

cot $^{-1}$**(***Liste1***)** ⇒*liste*

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

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

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

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

$\cot^{-1}(1)$ .785398

coth() Katalog >

coth(*Verdi1***)⇒***verdi*

coth(*Liste1***)⇒***liste*

Returnerer hyperbolsk cotangens til *uttrykk1*, eller returnerer en liste med hyperbolsk cotangens til alle elementene i *liste1*.

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

coth⁻¹() Katalog >

coth⁻¹(*Verdi1***)⇒***verdi*

coth⁻¹(*Liste1***)⇒***liste*

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

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

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

count() (antall) Katalog >

count(*Verdi1ellerListe1* [,*Verdi2ellerListe2* [,…]])**⇒***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 202.

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

countIf() (tellIf)

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 202.

Merk: Se også **sumIf()**, side 154, og **frequency()**, side 57.

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

Teller alle elementer som er lik 3.

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

Teller alle elementer som er lik “def.”

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

Teller 1 og 3.

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

Teller 3, 5 og 7.

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

Teller 1, 3, 7 og 9.

cPolyRoots()

cPolyRoots(*Poly*,*Var***)⇒***liste*

cPolyRoots(*KoeffListe***)⇒***liste*

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

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

Merk: Se også **polyRoots()**, side 116.

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

crossP() (kryssprodukt) Katalog >

crossP(*Liste1***, *Liste2*)⇒*liste***

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({0.1,2.2,-5},{1,-0.5,0})
                    {-2.5,-5.,-2.25}
```

crossP(*Vektor1***, *Vektor2*)⇒*vektor***

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

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

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

csc() trig tast

csc(*Verdi1***)⇒*verdi***

csc(*Liste1***)⇒*liste***

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

I Grader-vinkelmodus:

```
csc(45)                            1.41421
```

I Gradian-vinkelmodus:

```
csc(50)                            1.41421
```

I Radian-vinkelmodus:

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

csc⁻¹() trig tast

csc⁻¹(*Verdi1***) ⇒ *verdi***

csc⁻¹(*Liste1***) ⇒ *liste***

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

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

csc⁻¹() trig tast

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 datamaskintastaturet ved å skrive `arccsc(...)`.

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

I Radian-vinkelmodus:

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

csch() Katalog >

csch(*Verdi1***)** ⇒ *verdi*

csch(*Liste1***)** ⇒ *liste*

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

$\text{csch}(3)$	0.099822
$\text{csch}(\{1,2.1,4\})$	$\{0.850918,0.248641,0.036644\}$

csch⁻¹() Katalog >

csch⁻¹(*Verdi***)** ⇒ *verdi*

csch⁻¹(*Liste1***)** ⇒ *liste*

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(...)`.

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

CubicReg Katalog >

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 150).

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

CubicReg

Katalog >

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 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: a ·x^3+b ·x^2+c ·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 X-*listen* som faktisk brukes i regresjonen basert på begrensninger i $Frekv$, $Kategoriliste$, og $Inkludert\ kategorier$
stat.YReg	Liste over de datapunkter i den endrede Y-*listen* som faktisk brukes i regresjonen basert på begrensninger i $Frekv$, $Kategoriliste$ og $inkludert\ kategorier$
stat.FreqReg	Liste over frekvenser som samsvarer med $stat.XReg$ og $stat.YReg$

cumulativeSum()

Katalog >

cumulativeSum($Liste1$**)**⇒$liste$

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

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

cumulativeSum() Katalog >

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.

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 202.

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

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

Cycle (Løkke) Katalog >

Cycle (Løkke)

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.

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

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

▸Cylind Katalog >

Vektor **▸Cylind**

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

Viser rad- eller kolonnevektor i sylindrisk form [r, ∠θ, z].

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

```
[2  2  3]▸Cylind
                [2.82843  ∠0.785398  3.]
```

dbd() Katalog >

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.

MM.DDÅÅ (format som vanligvis brukes i USA)

DDMM.ÅÅ (format som vanligvis brukes i Europa)

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

▸DD Katalog >

Verdi **▸DD⇒***verdi*

Liste1 **▸DD⇒***liste*

Matrise1 **▸DD⇒***matrise*

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

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

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

1▸DD	$\frac{9}{10}$°

I Radian-vinkelmodus:

(1.5)▸DD	85.9437°

▸Decimal

Katalog >

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)

Katalog >

Define *Var* **=** *Uttrykk*

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

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

Parametere, som f.eks. *Param1*, er plassholdere for å sette argumenter til funksjonen. Når du kaller opp en egendefinert funksjon, må du legge til argumenter (for eksempel verdier eller variabler) som samsvarer med 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 $g(x,y)=2\cdot x-3\cdot y$	*Done*
$g(1,2)$	-4
$1\rightarrow a$: $2\rightarrow b$: $g(a,b)$	-4
Define $h(x)$=when$(x<2,2\cdot x-3,-2\cdot x+3)$	*Done*
$h(-3)$	-9
$h(4)$	-5

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

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

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

Define (Definer) Katalog >

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

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

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

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

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

Define LibPriv (Definer BiblPriv) Katalog >

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 38.

Define LibPub (Definer BiblOff) Katalog >

Define LibPub *Var* **=** *Uttrykk*

Define LibPub *Funksjon*(*Param1*, *Param2*,

Define LibPub (Definer BiblOff) Katalog >

...) = *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 38.

deltaList() Se ΔList(), side 85.

DelVar katalog >

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

DelVar *Var*.

Sletter den angitte variabelen eller variabelgruppen fra minnet.

Hvis en eller flere av variablene er låst, viser denne kommandoen en feilmelding og sletter kun de ulåste variablene. Se **unLock, side 168.**

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

DelVar katalog >

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

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

delVoid() Katalog >

delVoid(*Liste1***)⇒***liste*

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

For mer informasjon om tomme elementer, se side 202.

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

det() Katalog >

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

  5E⁻14 ·**maks(dim(***kvadratMatrise***))** · **radNorm(***kvadratMatrise***)**

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

diag() Katalog >

diag(*Liste***)⇒***matrise*

diag(*radMatrise***)⇒***matrise*

diag(*kolonneMatrise***)⇒***matrise*

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

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

diag(*kvadratMatrise***)⇒***radMatrise*

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

kvadratMatrise må være kvadrat.

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

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

dim() 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}.

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

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. Agrumentene 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 produkthåndboken.

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

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

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

Linjenummeret kan angis som et uttrykk.

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

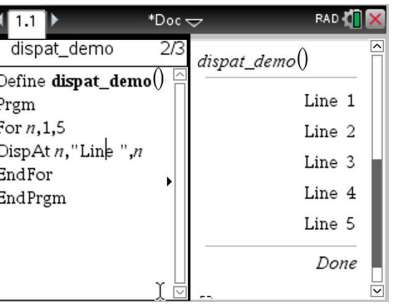
Denne kommandoen lar deg opprette en instrumentbordlignende visning av data fra programmer der verdien til et uttrykk eller en sensoravlesning oppdateres på samme linje.

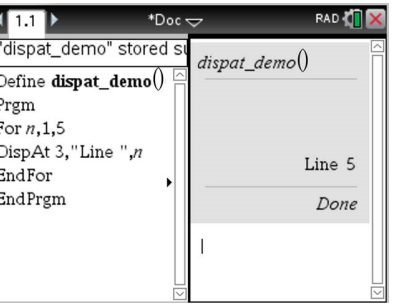
DispAtog Disp kan brukes i samme program.

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.

DispAt

Eksempel





Illustrerende eksempler:

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

Feilmeldinger:

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

▸DMS (GMS) Katalog > 📖

Verdi ▸*DMS*

Liste ▸**DMS**

Matrise ▸**DMS**

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

I Grader-vinkelmodus:

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

▸DMS (GMS) Katalog >

Tolker argumentet som en vinkel og viser ekvivalenten DMS (GGGGGG°MM ' SS.ss '') -tallet. Se °, ', '' (side 196) 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*

Returnerer "prikk"produktet av to lister.

dotP({1,2},{5,6})	17

dotP(*Vektor1***, *Vektor2***)⇒***uttrykk*

Returnerer "prikk"produktet av to vektorer.

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

Begge må være radvektorer, eller begge må være kolonnevektorer.

E

e^() [e^x] tast

e^(*Verdi1)***⇒***verdi*

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

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

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

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

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

e^(*Liste1***)⇒***liste*

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

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

e^()

e^(*kvadratMatrise1***)⇒***kvadratMatrise*

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

tast e^x

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

eff()

Katalog >

eff(*nominellRente,CpY***)⇒***verdi*

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 105.

$$\text{eff}(5.75,12) \qquad 5.90398$$

eigVc() (egenvektor)

Katalog >

eigVc(*kvadratMatrise***)⇒***matrise*

Returnerer en matrise som inneholder egenvektorer for en reell eller kompleks *kvadratMatrise*, der hver kolonne i resultatet samsvarer med en egenverdi. Merk at en egenvektor ikke er entydig; den kan skaleres av enhver konstant faktor. Egenvektorene er normalisert, dvs. at if V = [x1, x_2, ..., x_n], then:

$x_1^2 + x_2^2 + ... + x_n^2 = 1$

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

I rektangulært, kompleks format:

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

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

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

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

eigVl() (egenverdi) Katalog >

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:

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

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

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

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

Else Se If, side 70.

ElseIf Katalog >

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

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

EndFor Se For, side 55.

EndFunc Se Func, side 59.

EndIf

Se If, side 70.

EndLoop

Se Loop, side 91.

EndPrgm

Se Prgm, side 117.

EndTry

Se Try, side 162.

EndWhile

Se While, side 172.

euler ()

Katalog >

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

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

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

Bruker Eulers metode for å løse systemet

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

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

Differensialligning:

y'=0,001*y*(100-y) og y(0)=10

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

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

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

System av ligninger:

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

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

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

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

euler ()

Katalog >

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.

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

ListeMedAvhVarer er en liste over startverdier for avhengige variabler.

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

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

eval ()

Hub-meny

eval(*Uttr***)** ⇒ *streng*

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

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

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

Tilbakestill det blå elementet til AV.

eval () Hub-meny

Argumentet *Uttr* må forenkles til et reelt tall.

```
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.

```
fadein()                                     Done
```

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

iostr.SendAns
iostr.GetAns
iostr.GetStrAns

Merk: Les også **Get**(side 60), **GetStr** (side 67) og **Send** (side 137).

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

Exit (Avslutt) Katalog >

Exit

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

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

Program:

Exit (Avslutt) Katalog >

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

exp() tast

exp(*Verdi1***)⇒***verdi*

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

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

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

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

e^{1} 2.71828

e^{3^2} 8103.08

exp(*Liste1***)⇒***liste*

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

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

exp(*kvadratMatrise1***)⇒***kvadratMatrise*

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

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

expr() (uttrykk) Katalog >

expr (*String***)⇒***Uttrykk*

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

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

ExpReg Katalog >

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

Finner den eksponensielle regresjoneny = a · $(b)^x$for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 150).

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 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: a · $(b)^x$
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 *X-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste*, og *Inkludert kategorier*
stat.YReg	Liste over de datapunkter i den endrede *Y-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *inkludert kategorier*
stat.FreqReg	Liste over frekvenser som samsvarer med *stat.XReg* og *stat.YReg*

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.

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

Slik stopper du en beregning manuelt,

- **Grafregner:** Hold nede tasten [on], og trykk på [enter] 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() Katalog >

FCdf

FCdf() Katalog >

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

FCdf
(*nedGrense*,*øvGrense*,*dfTeller*,*dfNevner***)**
⇒*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($X \le$ *øvGrens*), set *nedGrens* = 0.

Fill (Fyll) Katalog >

Fill *Verdi, matriseVar* ⇒*matrise*

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

matriseVar må eksistere allerede.

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

Fill 1.01,*amatrix* *Done*

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

Fill *Verdi, listeVar* ⇒*liste*

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

listeVar må eksistere allerede.

$\{1,2,3,4,5\} \to alist$ $\{1,2,3,4,5\}$

Fill 1.01,*alist* *Done*

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

FiveNumSummary katalog >

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 150.)

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.

FiveNumSummary katalog >

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 202.

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) Katalog >

floor(*Verdi1***)⇒***heltall*

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

Argumentet kan være et reelt eller et komplekst tall.

floor(-2.14) -3.

floor(*Liste1***)⇒***liste*

floor(*Matrise1***)⇒***matrise*

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

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

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

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

For

Katalog >

For *Var*, *Lav*, *Høy* [, *Intervall*]

Blokk

EndFor

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

Var må ikke være en systemvariabel.

Intervall kan være positiv eller negativ. Grunnverdien er 1.

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

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

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

format()

Katalog >

format(*Verdi*[, *formatStreng*]**)**⇒*streng*

Returnerer *Verdi* som en tegnstreng basert på formatsjablonen.

formatStreng er en streng og må være av formen: "F[n]", "S[n]", "E[n]", "G[n][c]", hvor [ ] viser alternative muligheter.

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

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

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

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

format() Katalog >

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) Katalog >

fPart(*Uttr1***)⇒***uttrykk*

fPart(*Liste1***)⇒***liste*

fPart(*Matrise1***)⇒***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.

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

FPdf() Katalog >

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

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

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

freqTable▸liste() katalog >

freqTable▸liste(*Liste1***,***frekvHeltallListe***)** ⇒*liste*

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

freqTable▸liste() katalog >

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 202.

frequency() (frekvens) Katalog >

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

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

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

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

$datalist:=\{1,2,\mathbf{e},3,\pi,4,5,6,\text{"hello"},7\}$
$\{1,2,2.71828,3,3.14159,4,5,6,\text{"hello"},7\}$

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

Forklaring til resultat:

2 elementer fra *Dataliste* er ≤2,5

4 elementer fra *Dataliste* er >2,5 og ≤4,5

3 elementer fra *Dataliste* er >4,5

Elementet “hallo” er en streng og kan ikke plasseres i no pen av de definerte stolpene.

frequency() (frekvens) Katalog >

Elementer fra *Liste1* som ikke kan "plasseres i en stolpe" ignoreres. Tomme (åpne) elementer ignoreres også. For mer informasjon om tomme elementer, se side 202.

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) Katalog >

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 150).

eller H_a: σ1 > σ2, sett *Hypoth*>0

For H_a: σ1 ≠ σ2 (standard), sett *Hypoth* =0

For H_a: σ1 < σ2, sett *Hypoth*<0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" på side 202.

Utdata-variabel	Beskrivelse
stat.F	Beregnet Û -statistikk for datasekvensen
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.dfNumer	frihetsgrad for teller = n1-1
stat.dfDenom	frihetsgrad for nevner = n2-1
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i *Liste 1* og *Liste 2*

Utdata-variabel	Beskrivelse
stat.x1_bar stat.x2_bar	Utvalgets gjennomsnitt av datasekvenser i *Liste 1* og *Liste 2*
stat.n1, stat.n2	Utvalgenes størrelse

Func (Funk) Katalog >

Func
Blokk
EndFunc

Sjablon for oppretting av brukerdefinert funksjon.

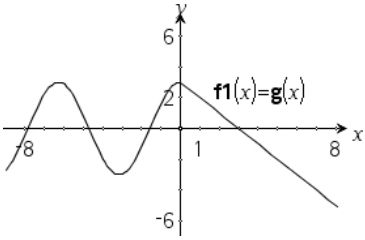
Blokk kan være ett enkelt utsagn, en rekke utsagn adskilt med ":"-tegnet, eller en rekke med utsagn på separate linjer. Funksjonen kan bruke **Returner**-kommandoen for å returnere et spesifikt resultat.

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***)⇒***uttrykk*

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 desimalpunkttall 1,0.

```
gcd(18,33)                          3
```

gcd(*Liste1, Liste2***)⇒***liste*

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

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

gcd() (største felles divisor) Katalog >

gcd(*Matrise1, Matrise2***)⇒***matrise*

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

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

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≤X ≤*øvreGrense*)⇒*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 ≤ *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:

- På forhånd, gjennom en **Send «LES ...»** -

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

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

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 67 og **Send**, side 137.

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

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

getDenom() (lesNevner) Katalog >

getDenom(*Brøk1***)⇒***verdi*

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([0|1]) ⇒ returnString

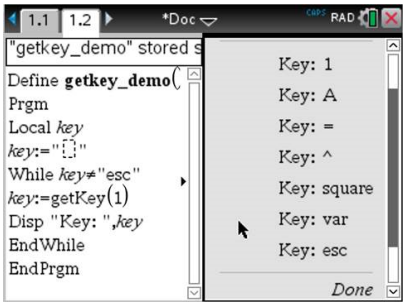
Beskrivelse:getKey() – tillater at et TI-Basic-program henter tastaturinndata – grafregner, stasjonær PC og emulator på skrivebordet.

Eksempel:

- keypressed := **getKey()** returnerer en tast eller en tom streng hvis ingen tast er trykket ned. Dette oppkallet returneres umiddelbart.
- keypressed := **getKey(1)** venter til en tast trykkes ned. Dette oppkallet setter utførelsen av programmet på pause til en tast trykkes ned.

```
getKey()
```

Eksempel:



Håndtering av tastetrykk:

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
Esc	Esc	"esc"
Styreplate – klikk oppe	Ikke relevant	"opp"
På	Ikke relevant	"start"
Kladdeapp	Ikke relevant	"kladdeblokk"
Styreplate – venstreklikk	Ikke relevant	"venstre"
Styreplate – midtklikk	Ikke relevant	"midtre"
Styreplate – høyreklikk	Ikke relevant	"høyre"
Dok	Ikke relevant	"dok"
Kategori	Kategori	"kategori"
Styreplate – klikk nede	Piltast ned	"ned"
Meny	Ikke relevant	"meny"
Ctrl	Ctrl	ingen retur
Skift	Skift	ingen retur
Variabel	Ikke relevant	"var"

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
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)
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	"?!"

Håndholdt enhet / emulatortast	Stasjonær PC	Returverdi
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 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,	Brukerstøtte

Hendelse	Enhet	Stasjonær PC – TI-Nspire™ Student Software
	behandle en hendelse	(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*

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.

```
getLangInfo()                    "en"
```

Engelsk = “en”

Dansk = “da”

Tysk = “de”

Finsk = “fi”

Fransk = “fr”

Italiensk = “it”

Nederlandsk = “nl”

Belgisk nederlandsk = “nl_BE”

Norsk = “no”

Portugisisk = “pt”

Spansk = “es”

Svensk = “sv”

getLockInfo() Katalog >

getLockInfo(*Var***)⇒***verdi*

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 88, og unLock, side 168.**

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

GetMode() (lesModus) Katalog >

GetMode(*ModusNavnHeltall***)⇒***verdi*

GetMode(0)⇒*liste*

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

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

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

Hvis du lagrer innstillingene med **GetMode (0) → *var***, kan du bruke **GetMode(*var*)** i en funksjon eller et program for midlertidig å gjenopprette innstillingene kun innenfor utføringen av funksjonen eller programmet. Se **GetMode()**, side 140.

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

Modus Navn	Modus Heltall	Innstille heltall
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*]

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

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

Merk: Se også **Get**, side 60 og **Send**, side 137.

For eksempler, se **Get**.

getType() Katalog >

getType(*var***)⇒***streng*

Returnerer en streng som angir dataens typevariabel *var*.

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

$\{1,2,3\}\rightarrow temp$	$\{1,2,3\}$
getType(*temp*)	"LIST"
$3\cdot i\rightarrow temp$	$3\cdot i$
getType(*temp*)	"EXPR"
DelVar *temp*	*Done*
getType(*temp*)	"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()	$\begin{bmatrix} x & \text{"NUM"} & \text{"}\square\text{"} & 1 \\ y & \text{"LIST"} & \text{"LibPriv "} & 0 \\ z & \text{"FUNC"} & \text{"LibPub "} & 0 \end{bmatrix}$
getVarInfo(*tmp3*)	"Error: Argument must be a string"
getVarInfo("tmp3")	[*volcyl2* "NONE" "LibPub " 0]

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.

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

getVarInfo() katalog >

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.

Goto (Gåtil) Katalog >

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

▸Grad Katalog >

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)▸Grad                         (1.66667)g
```

I Radian-vinkelmodus:

```
(1.5)▸Grad                         (95.493)g
```

I

identity() Katalog >

identity(*Heltall***)** ⇒ *matrise*

Returnerer identitetsmatrisen med dimensjonen *Heltall*.

Heltallet må være et positivt heltall.

```
identity(4)
```

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

Hvis *BooleanExpr*
Utsagn

Hvis *BooleanExpr*, **så**
Blokk
OgHvis

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

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

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

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

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

Hvis *BooleanExpr*, **så**
Blokk1
hvis ikke
Blokk2
OgHvis

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

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

Blokk1 og *Blokk2* kan være et enkelt utsagn.

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

Hvis

Katalog >

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 -x
        ElseIf x≥0 and x≠10 Then
        Return x
        ElseIf x=10 Then
        Return 3
        EndIf
        EndFunc
                                Done
g(-4)                            4
g(10)                            3
```

ifFn()

Katalog >

ifFn(*BooleanExpr*,*Value_If_true* [,*Value_If_false* [,*Value_If_unknown*]]**)** ⇒ *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 uttrykk, brukes det boolske uttrykket i hver posisjon i *BooleanExpr*.

```
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.

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

ifFn() Katalog >

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.

Value_If_false er ikke spesifisert. Udef brukes.

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

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

imag() Katalog >

imag(*Value1*) ⇒ *verdi*

Returnerer den imaginære delen av argumentet.

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

imag(*List1*) ⇒ *liste*

Returnerer en liste over de imaginære delene av elementene.

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

imag(*Matrix1*) ⇒ *matrise*

Returnerer en matrise over de imaginære delene av elementene.

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

Indireksjon Se #(), side 194.

inString() Katalog >

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

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

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

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

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

int() Katalog >

int(*Verdi***)** ⇒ *heltall*
int(*List1***)** ⇒ *liste*
int(*Matrix1***)** ⇒ *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.

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

intDiv() Katalog >

intDiv(*Number1***, *Number2***)** ⇒ *heltall*
intDiv(*List1***, *List2***)** ⇒ *liste*
intDiv(*Matrix1***, *Matrix2***)** ⇒ *matrise*

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

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

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

interpoler () Katalog >

interpoler(*xValue***, *xList***, *yList***, *yPrimeList***)** ⇒ *liste*

Denne funksjonen gjør følgende:

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

Differensialligning:
y'=-3•*y*+6•*t*+5 og *y*(0)=5

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

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

interpoler () Katalog >

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-verdilisten:

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

invχ^2() Katalog >

invχ^2(*Area*,*df*)**

invChi2(*Area*,*df*)**

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

invF() Katalog >

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

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

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

invBinom() Katalog >

invBinom (*CumulativeProb*,*NumTrials*,*Prob*, *OutputForm*)⇒** *skalar* eller *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 %?

invBinom() Katalog >

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.

$\text{invBinom}\left(0.77,30,\frac{1}{6}\right)$ 6

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

invBinomN() Katalog >

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 nettball. 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() Katalog >

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() Katalog >

invt(*Area***,***df***)**

invt() Katalog >

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

iPart() Katalog >

iPart(*Number***)** ⇒ *heltall*
iPart(*List1***)** ⇒ *liste*
iPart(*Matrix1***)** ⇒ *matrise*

Returnerer heltallsdelen av argumentet.

For lister og matriser, returnerer heltallsdelen for hvert element.

Argumentet kan være et reelt eller et komplekst tall.

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

irr() Katalog >

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

Finansiell funksjon som beregner internrente av retur av en investering.

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

CFList er en liste over kontantstrømbeløpene etter den innledende kontanstrømmen 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 97.

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

isPrime() Katalog >

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

isPrime(5)	true
isPrime(6)	false

isPrime() Katalog >

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

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

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

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

```
Define nextprim(n)=Func                    Done
                   Loop
                   n+1→n
                   If isPrime(n)
                   Return n
                   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 202.

```
a:=_                                         _
isVoid(a)                                 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.

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

lcm() (mfm) Katalog >

lcm(*Tall1***, *Tall2***)⇒***uttrykk***

lcm(*Liste1***, *Liste2***)⇒***liste***

lcm(*Matrise1***, *Matrise2***)⇒***matrise***

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

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

$\text{lcm}(6,9)$ → 18

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

left() (venstre) Katalog >

left(*kildeStreng***[, *Num***])⇒***streng***

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

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

left("Hello",2) → "He"

left(*Liste1***[, *Num***])⇒***liste***

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

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

left({1,3,-2,4},3) → {1,3,-2}

left(*Sammenlikning***)⇒***Uttrykk***

Returnerer venstre side av en ligning eller ulikhet.

libShortcut() katalog >

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

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

libShortcut() katalog >

Oppretter en variabelgruppe i den gjeldende oppgaven som inneholder referanser til alle objektene i det angitte bibliotekdokumentet *bibliotekNavnStreng*. Legger også gruppemedlemmene 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 39).

```
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) katalog >

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

Finner den lineære regresjoneny = a+b ·xfor listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 150.)

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.

LinRegBx (lineær regresjon) katalog >

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: a+b ·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 *X-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.YReg	Liste over de datapunkter i den endrede *Y-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.FreqReg	Liste over frekvenser som samsvarer med *stat.XReg* og *stat.YReg*

LinRegMx (lineær regresjon) katalog >

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

Finner den lineære regresjonen y = m ·x+b for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (side 150.)

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.

LinRegMx (lineær regresjon) katalog >

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 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: m ·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 *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.YReg	Liste over de datapunkter i den endrede *Y-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.FreqReg	Liste over frekvenser som samsvarer med *stat.XReg* og *stat.YReg*

LinRegtIntervals (lineær regresjon) katalog >

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 150.)

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 202).

Utdata-variabel	**Beskrivelse**
stat.RegEqn	Regresjonsligning: a+b ·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
[stat.LowerPred, stat.UpperPred]	Prediksjonsintervall for én enkeltobservasjon

Utdata-variabel	Beskrivelse
stat.MEPred	Prediksjonsintervallets feilmargin
stat.SEPred	Standardfeil for prediksjonen
stat.$\hat{y}$	a + b ·XVerdi

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*=α+βx. Den tester nullhypotesen H_0:β=0 (tilsvarende, ρ=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:β=ρ=0) skal testes mot.

For H_a: β≠0 og ρ≠0 (standard), sett *Hypot*=0

For H_a: β<0 og ρ<0, sett *Hypot*<0

For H_a: β>0 og ρ>0, sett *Hypot*>0

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 150).

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: a + b ·x
stat.t	*t*-observator for signifikanstest

Utdata-variabel	Beskrivelse
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*

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

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

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

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.

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

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

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

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

Δlist() (liste) Katalog >

ΔList(*Liste1***)⇒***liste*

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

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

ΔList({20,30,45,70})	{10,15,25}

list▸mat() Katalog >

list▸mat(*Liste* **[,** *elementerRerRad***])** ⇒*matrise*

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

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

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

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

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

ln() ctrl e^x taster

ln(*Verdi1***)⇒***verdi*

ln(*Liste1***)⇒***liste*

Returnerer argumentets naturlige logaritme.

For en liste, returneres elementenes naturlige logaritmer.

ln(2.)	0.693147

Hvis kompleks formatmodus er reell:

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

Hvis kompleks formatmodus er rektangulær:

ln() ctrl e^x taster

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:

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

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

LnReg katalog >

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

FInner den logaritmiske regresjonen y = a+b ·ln(x) for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 150).

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 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: a+b · 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 *X-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.YReg	Liste over de datapunkter i den endrede *Y-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.FreqReg	Liste over frekvenser som samsvarer med *stat.XReg* og *stat.YReg*

Local

Katalog >

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

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

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

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

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

Lock (Lås) Katalog >

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 168 og getLockInfo(), side 66.**

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

log() ctrl 10^x taster

log(*Verdi1***[,***Verdi2***])⇒***verdi*

log(*Liste1***[,***Verdi2***])⇒***liste*

Returnerer grunntallet -*Verdi2* argumentets logaritme.

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

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

Hvis *Uttr2* utelates, brukes 10 som grunntall.

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

Hvis kompleks formatmodus er reell:

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

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:

log() ctrl 10ˣ taster

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.

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

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

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

Logistic katalog >

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

Finner den logistiske regresjonen y = (c/(1+a ·e^{-bx})) for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 150).

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 202).

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

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 150).

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..

LogisticD katalog >

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 202).

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

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 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
```

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 · *uMatrise* = *pMatrise* · *matrise*

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

- Hvis du bruker ctrl enter 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 ·maks(dim(*Matrise*)) ·radNorm (*Matrise*)

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

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

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

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

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

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

M

mat▸list() (matrise til liste) Katalog >

mat▸list(*Matrise***)⇒***liste*

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

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

mat▸list([1 2 3]) — {1,2,3}

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

mat▸list(*m1*) — {1,2,3,4,5,6}

max() Katalog >

max(*Verdi1***, *Verdi2***)⇒***Uttrykk*

max(*Liste1***, *Liste2***)⇒***liste*

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.

$\max(2.3,1.4)$ → 2.3

$\max(\{1,2\},\{-4,3\})$ → $\{1,3\}$

max(*Liste***)⇒***Uttrykk*

Returnerer maksimumelementet i *liste*.

$\max(\{0,1,-7,1.3,0.5\})$ → 1.3

max(*Matrise1***)⇒***matrise*

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

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

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 202.

Merk: Se også **min()**.

mean() (gjennomsnitt) Katalog >

mean(*Liste***[, *frekvListe***])⇒***uttrykk*

Returnerer gjennomsnittet av elementene i *Liste*.

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

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

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 202.

mean() (gjennomsnitt) Katalog >

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

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

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

median() Katalog >

median(*Liste***[,** *frekvListe***])⇒***uttrykk*

Returnerer medianen av elementene i *Liste*.

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

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

median(*Matrise1***[,** *frekvMatrise***])** ⇒*matrise*

Returnerer en radvektor som inneholder medianer av kolonnene i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i *Matrise1*.

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

Merknader:

- Alle inndata i listen eller matrisen må forenkles til tall.
- Tomme (åpne) elementer i listen eller matrisen ignoreres. For mer informasjon om tomme elementer, se side 202.

MedMed katalog >

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

Beregner median-median-linjeny = (m ·x+b) for listene X og Y med frekvens $Frekv$. En oversikt over resultatene lagres i $stat.resultater$-variabelen. (Se side 150.)

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 202).

Utdata-variabel	**Beskrivelse**
stat.RegEqn	Ligning for median-median-linjen: m ·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 $Frekv$, $Kategoriliste$ og $Inkluder\ kategorier$
stat.YReg	Liste over de datapunkter i den endrede Y-$listen$ som faktisk brukes i regresjonen basert på begrensninger i $Frekv$, $Kategoriliste$ og $Inkluder\ kategorier$
stat.FreqReg	Liste over frekvenser som samsvarer med $stat.XReg$ og $stat.YReg$

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

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({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(*kildeStrengListe***, *Start*[, *Antall*]) ⇒*liste***

Returnerer *Antall* strenger fra listen over strenger *kildeStrengListe*, begynnende med elementnummer *Start*.

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(2.3,1.4)	1.4
min({1,2},{-4,3})	{-4,2}

min(*Liste***)⇒*Uttrykk***

Returnerer minimumselementet av *Liste*.

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

min() (minimum) Katalog >

min(*Matrise!***)⇒***matrise*

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

Merk: Se også **max()**.

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

mirr() Katalog >

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

Finansiell funksjon som returnerer modifisert rente av en investering.

finanseRente 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 kontanstrømmen 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 76.

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

mod() Katalog >

mod(*Verdi1***,** *Verdi2***)⇒***Uttrykk*

mod(*Liste1***,** *Liste2***)⇒***liste*

mod(*Matrise1***,** *Matrise2***)⇒***matrise*

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

mod() Katalog >

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

mod(x,0) = x

mod(x,y) = x − y 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 128

mRow() (mRad) Katalog >

mRow(*Verdi***,** *Matrise1***,** *Indeks***)⇒***matrise*

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

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

mRowAdd() (mRadAdd) Katalog >

mRowAdd(*Verdi***,** *Matrise1***,** *Indeks1***,** *Indeks2***)** ⇒*matrise*

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

Verdi × *rad Indeks1* + *rad Indeks2*

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

MultReg katalog >

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

Beregner multiple lineære regresjoner av liste *Y* for listene *X2, X2, …, X10*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 150).

Alle listene må ha samme dimensjon.

MultReg katalog >

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: b0+b1 ·x1+b2 ·x2+ …
stat.b0, stat.b1, …	Regresjonskoeffisienter
stat.R^2	Multippel determinasjonskoeffisient
stat.ŷList	ŷList = b0+b1 ·x1+ …
stat.Resid	Residualene fra regresjonen

MultRegIntervals katalog >

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 150).

Alle listene må ha samme dimensjon.

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: b0+b1 ·x1+b2 ·x2+ …
stat.ŷ	Et punktestimat: ŷ = b0 + b1 · xl + … for *XVerListe*
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, {b0,b1,b2,...}
stat.Resid	Residualene fra regresjonen

MultRegTests katalog >

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

Multippel lineær regresjonstest beregner en multippel lineær regresjon på de angitte dataene og beregner den globale statistiske F- og t-testobservatoren for koeffisientene.

En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 150).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 202).

Utdata

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: b0+b1 ·x1+b2 ·x2+ ...
stat.F	Global statistisk F-testobservator
stat.PVal	P-verdi knyttet til global F-observator
stat.R^2	Multippel determinasjonskoeffisient
stat.AdjR^2	Justert multippel 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	ŷList = b0+b1 ·x1+ . . .
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)

ctrl = -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*⇒*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}

nand (ikke både...og)

ctrl = -taster

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*

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

$\mathrm{nCr}(z,3)\|z=5$	10
$\mathrm{nCr}(z,3)\|z=6$	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*

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

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

nCr(*Matrise1***,** *Matrise2***)⇒***matrise*

Returnerer en matrise av kombinasjoner basert på samsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.

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

nDerivative() Katalog >

nDerivative(*Uttr1***,***Var=Verdi***[,***Orden***])** ⇒*verdi*

nDerivative(*Uttr1***,***Var***[,***Orden***])** | *Var=Verdi*⇒*verdi*

Returnerer den numeriske deriverte som er beregnet ved hjelp av automatiske derivasjonsmetoder.

Hvis *verdi* er spesifisert, opphever den eventuell forhåndstildelt 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**.

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

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·(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 **nDerivativ()** resultatet som udefinert og viser en varselmelding.

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

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

newList() (nyListe) Katalog >

newList(*numElementer***)**⇒*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***)** ⇒*matrise*

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

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

nfMax() Katalog >

nfMax(*Uttr1***, *Var***)**⇒*verdi*

nfMax(*Uttr1***, *Var*, *nedGrense***)**⇒*verdi*

nfMax(*Uttr1***, *Var*, *nedGrense*, *øvGrense***)** ⇒*verdi*

nfMax(*Uttr1*, *Var***) | *nedGrense*≤*Var* ≤*øvGrense*⇒*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.

$\text{nfMax}\left(-x^2-2\cdot x-1,x\right)$	-1.
$\text{nfMax}\left(0.5\cdot x^3-x-2,x,-5,5\right)$	5.

nfMin() Katalog >

nfMin(*Uttr1***, *Var***)**⇒*verdi*

nfMin(*Uttr1***, *Var*, *nedGrense***)**⇒*verdi*

nfMin(*Uttr1***, *Var*, *nedGrense*, *øvGrense***)** ⇒*verdi*

nfMin(*Uttr1*, *Var***) | *nedGrense*≤*Var* ≤*øvGrense*⇒*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.

$\text{nfMin}\left(x^2+2\cdot x+5,x\right)$	-1.
$\text{nfMin}\left(0.5\cdot x^3-x-2,x,-5,5\right)$	-5.

nInt() Katalog >

nInt(*Uttr1, Var, Nedre, Øvre***)⇒***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 ∫**(***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.

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

Det kommer til syne et varsel ("Tvilsom nøyaktighet") når det ser ut til at målet ikke er nådd.

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

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

nom() Katalog >

nom(*effektivRente*,*CpY***)⇒***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 perioder per år.

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

Merk: Se også **eff()**, side 45.

nor (verken ... eller) ctrl = -taster

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

*BoolskListe1***nor***BoolskListe2* returnerer *Boolsk liste*

*BoolskMatrise1***nor***BoolskMatrise2*

nor (verken ... eller) ctrl = -taster

returnerer *Boolsk matrise*

Returnerer negasjon av en logisk or-handling på de to argumentene. Returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Heltall1 **nor** *Heltall2*⇒*heltall*

Sammenlikner to reelle heltall bit-for-bit med en nor-handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis begge bitene er 1; ellers er resultatet 0. Den returnerte verdien representerer bit-resultatene og vises i grunntallmodus.

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

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

norm() katalog >

norm(*Matrise***)**⇒*uttrykk*

norm(*Vektor***)**⇒*uttrykk*

Returnerer Frobenius-normen.

$\text{norm}\left(\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\right)$	5.47723
$\text{norm}\left(\begin{bmatrix}1 & 2\end{bmatrix}\right)$	2.23607
$\text{norm}\left(\begin{bmatrix}1\\ 2\end{bmatrix}\right)$	2.23607

normCdf() Katalog >

normCdf(*nedreGrense*,*øvreGrense*[,μ[,σ]]**)** ⇒*tall* hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er lister

normCdf() Katalog >

Beregner sannsynligheten i normalfordelingen mellom *nedreGrense* og *øvreGrense* for den angitte μ (standard=0) og σ (standard=1).

For P(X ≤ *øvreGrense*), sett *nedreGrense* = −9**E**999.

normPdf() Katalog >

normPdf(*XVerdi[* [,μ [,σ]]**)**⇒*tall* hvis *XVerdi* er et tall, *liste* hvis *XVerdi* er en liste

Beregner sannsynlighetstetthet for normalfordelingen ved en spesifisert *XVerdi*-verdi for spesifisert μ og σ.

not (ikke) Katalog >

not *BoolksUttr1*⇒*Boolsk uttrykk*

Returnerer sann, usann eller en forenklet form av argumentet.

not *Heltall1*⇒*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≥3)	true
not 0hB0▶Base16	0hFFFFFFFFFFFFFF4F
not not 2	2

I heksades grunntall-modus:

Viktig: Null, ikke bokstaven O.

not 0h7AC36	0hFFFFFFFFFFF853C9

I binær grunntall-modus:

0b100101▶Base10	37
not 0b100101	
0b1111111111111111111111111111111▶	
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(*Verdi1*, *Verdi2***)⇒***Uttr1*

For heltall *Verdi1* og *Verdi2* der *Verdi1* ≥ *Verdi2* ≥ 0, er **nPr()** antall permutasjoner av *Verdi1* som tar *Verdi2* om gangen.

nPr(z,3)|z=5	60
nPr(z,3)|z=6	120
nPr({5,4,3},{2,4,2})	{20,24,6}
nPr([6 5; 4 3],[2 2; 2 2])	[30 20; 12 6]

nPr(*Verdi*, **0)⇒1**

nPr(*Verdi*, *negHeltall***)**

⇒ **1/((***Verdi***+1)** ·**(***Verdi***+2)**...
(*Verdi*−*negHeltall***))**

nPr(*Verdi*, *posHeltall***)**

⇒ *Verdi* ·**(***Verdi*−**1)...**
(*Verdi*−*posHeltall***+1)**

nPr(*Verdi*, *ikkeHeltall***)**

⇒*Verdi***! / (***Verdi*−*ikkeHeltall***)!**

nPr(*Liste1*, *Liste2***)⇒***liste*

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

nPr({5,4,3},{2,4,2})	{20,24,6}

nPr(*Matrise1*, *Matrise2***)⇒***matrise*

Returnerer en matrise av permutasjoner basert på tilsvarende elementpar i de to matrisene. Argumentene må ha samme matrisestørrelse.

nPr([6 5; 4 3],[2 2; 2 2])	[30 20; 12 6]

npv() Katalog >

npv(*Rentefot*,*CFO*,*CFListe*[,*CFFrekv*]**)**

Finansiell funksjon som beregner netto nåverdi; summen av nåverdier for kontanstrøm inn og ut. Et positivt resultat for npv forteller at en investering er lønnsom.

list1:={6000,⁻8000,2000,⁻3000}	{6000,⁻8000,2000,⁻3000}
list2:={2,2,2,1}	{2,2,2,1}
npv(10,5000,*list1*,*list2*)	4769.91

Rentefot er den renten som trekkes fra kontantstrømmene (pengekostnadene) over en periode.

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

npv() Katalog >

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) Katalog >

nSolve(*Ligning*,*Var*[=*Forslag*])⇒*tall eller feil_streng*

nSolve(*Ligning*,*Var*[=*Forslag*],*nedGrense*) ⇒*tall eller feil_streng*

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

nSolve(*Ligning*,*Var*[=*Forslag*]) | *nedGrense*≤*Var*≤*øvGrense* ⇒*tall eller feil_streng*

Søker iterativt etter en tilnærmet reell numerisk løsning for *Ligning* i variabelen. Spesifiser variabelen som:

variabel

– eller –

variabel = *reelt tall*

For eksempel er x gyldig, og det er x=3 også.

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

nSolve() prøver å bestemme enten ett punkt der rest er null, eller to relativt nære punkter, der rest har motsatte fortegn og størrelsen på resten ikke er for stor. Hvis den ikke kan oppnå dette med et lite antall utvalgspunkter, returnerer den strengen "fant ingen løsning."

$\text{nSolve}(x^2+5\cdot x-25=9,x)\|x<0$	-8.84429
$\text{nSolve}\left(\frac{(1+r)^{24}-1}{r}=26,r\right)\|r>0 \text{ and } r<0.25$	0.006886
$\text{nSolve}(x^2=-1,x)$	"No solution found"

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 150).

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 202.

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-variabel	Beskrivelse
stat. x	Populasjonens standardavvik av x
stat.n	Antall datapunkter
stat.MinX	Minimum av x-verdier
stat.Q_1X	Første kvartil av x
stat.MedianX	Median av x
stat.Q_3X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdier
stat.SSX	Sum av kvadratavvik fra gjennomsnittet av x

or (eller) Katalog >

BoolskUttr1 **or** *BoolskUttr2* 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 flerlinjede program- og funksjonsdefinisjoner, se avsnittet Kalkulator i produkthåndboken.

```
Define g(x)=Func                    Done
              If x≤0 or x≥5
              Goto end
              Return x·3
              Lbl end
              EndFunc
g(3)                                9
g(0)          A function did not return a value
```

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

I heksades grunntall-modus:

```
0h7AC36 or 0h3D5F                   0h7BD7F
```

Viktig: Null, ikke bokstaven O.

I binær grunntall-modus:

or (eller) Katalog >

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

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

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

Merk: Se **xor.**

0b100101 or 0b100	0b100101

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

ord() (num. tegnkode) Katalog >

ord(*Streng***)⇒***heltall*

ord(*Liste1***)⇒***liste*

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

ord("hello")	104
char(104)	"h"
ord(char(24))	24
ord({"alpha","beta"})	{97,98}

P

P▶Rx() Katalog >

P▶Rx(*rUttr***, θ***Uttr***)⇒***uttrykk*

P▶Rx(*rListe***, θ***Liste***)⇒***liste*

P▶Rx(*rMatrise***, θ***Matrise***)⇒***matrise*

Returnerer ekvivalent x-koordinat av (r, θ) paret.

I Radian-vinkelmodus:

P▶Rx(4,60°)	2.
P▶Rx({⁻3,10,1.3},{π/3,⁻π/4,0})	{⁻1.5,7.07107,1.3}

P▸Rx() Katalog >

Merk: θ-argumentet tolkes enten som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus. Hvis argumentet er et uttrykk, kan du bruke °, G eller r for å hoppe over vinkelmodus-innstillingen midlertidig.

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

P▸Ry() Katalog >

P▸Ry(*rVerdi***, θ***Verdi***)⇒***verdi***

P▸Ry(*rListe***, θ***Liste***)⇒***liste***

P▸Ry(*rMatrise***, θ***Matrise***)⇒***matrise***

Returnerer ekvivalent y-koordinat av (r, θ)-paret.

Merk: θ-argumentet tolkes enten som grader, gradianer eller radianer, avhengig av aktuell vinkelmodus.

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

I Radian-vinkelmodus:

$$P\blacktriangleright Ry(4,60°) \qquad 3.4641$$

$$P\blacktriangleright Ry\left(\{-3,10,1.3\},\left\{\frac{\pi}{3},\frac{-\pi}{4},0\right\}\right) \qquad \{-2.59808,-7.07107,0\}$$

PassErr (SendFeil) Katalog >

PassErr

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 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 **ClrErr** (**SlettFeil**), side 23, og **Try**, side 162.

For et eksempel på **PassErr**, se eksempel 2 under **Try** -kommandoen, side 162.

PassErr (SendFeil) Katalog >

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

piecewise() (stykkevis) Katalog >

piecewise(*Uttr1* [**,** *Betingelse1*[**,** *Uttr2*[**,** *Betingelse2* [**,** ...]]]]**)**

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

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

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

poissCdf() Katalog >

poissCdf(λ**,***nedreGrense***,***øvreGrense***)⇒***tall* hvis *nedreGrense* og *øvreGrense* er tall, *liste* hvis *nedreGrense* og *øvreGrense* er lister

poissCdf(λ**,***øvreGrense***)**(for P(0≤X ≤*øvreGrense*)⇒*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 ≤ *øvreGrense*), sett *nedreGrense*=0

poissPdf() Katalog >

poissPdf(λ**,***XVerd***)⇒***tall* hvis *XVerd* er et tall, *liste* hvis *XVerd* er en liste

Beregner en sannsynlighet for diskret Poisson-fordeling med spesifisert gjennomsnitt λ.

▶Polar Katalog >

Vektor ▶**Polar**

$[1 \quad 3.]$▶Polar	$[3.16228 \quad \angle 71.5651]$

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

Viser *vektor* i polar form [r∠ θ]. 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 125.

kompleksVerdi ▶**Polar**

Viser *kompleksVektor* i polar form.

- Grader-vinkelmodus returnerer (r∠ θ).
- 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∠ θ).

I Radian-vinkelmodus:

$(3+4\cdot i)$▶Polar	$e^{.927295\cdot i}\cdot 5$
$\left(\left\{4 \angle \frac{\pi}{3}\right\}\right)$▶Polar	$e^{1.0472\cdot i}\cdot 4.$

I Gradian-vinkelmodus:

$(4\cdot i)$▶Polar	$(4\angle 100.)$

I Grader-vinkelmodus:

$(3+4\cdot i)$▶Polar	$(5 \angle 53.1301)$

polyEval() Katalog >

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

polyEval(*Liste1***,** *Liste2***)⇒***uttrykk*

polyEval({1,2,3,4},2)	26
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() Katalog >

polyRoots(*Poly***,***Var***)** ⇒*liste*

polyRoots(*KoeffListe***)** ⇒*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 lsite over sanne røtter for koeffisientene i *KoeffListe*.

Merk: Se også **cPolyRoots()**, side 31.

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

PowerReg (PotensReg) katalog >

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

Finner potensregresjoneny = $(a \cdot (x)^b)$for listene *X* og *Y* med frekvensen *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen (side 150).

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.

PowerReg (PotensReg) katalog >

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

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

Prgm Katalog >

Prgm
Blokk
EndPrgm

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

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

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

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

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
```

prodSeq()

Se Π(), side 191.

Produkt (PI)

Se Π(), side 191.

Product()

Katalog >

product(*Liste*[, *Start*[, *slutt*]])⇒*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(*Matrise1*[, *Start*[, *slutt*]])⇒*matrise*

Returnerer en radvektor som inneholder produktet av elementene i kolonnene i *Matrise1*. *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 ignoreres. For mer informasjon om tomme elementer, se side 202.

propFrac() (ekteBrøk)

Katalog >

propFrac(*Verdi1*[, *Var*])⇒*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 ctrl enter 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 ·maks(dim(*Matrise*)) ·radNorm (*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 desimalpunkttallet (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*

$$qm \quad \begin{bmatrix}0.123091&0.904534&0.408248\\0.492366&0.301511&-0.816497\\0.86164&-0.301511&0.408248\end{bmatrix}$$

$$rm \quad \begin{bmatrix}8.12404&9.60114&11.0782\\0.&0.904534&1.80907\\0.&0.&0.\end{bmatrix}$$

QuadReg (KvadReg)

katalog >

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

Finner den kvadratiske polynomiske regresjoneny = a ·x²+b ·x+cfor listene *X* og *Y* med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 150.)

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

QuadReg (KvadReg) katalog >

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 202).

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 *Frekv, Kategoriliste* og *Inkluder kategorier*
stat.YReg	Liste over de datapunkter i den endrede *Y-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv, Kategoriliste* og *Inkluder kategorier*
stat.FreqReg	Liste over frekvenser som samsvarer med *stat.XReg* og *stat.YReg*

QuartReg Katalog >

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

Finner den fjerdegrads polynomiske regresjonen

y = a ·x^4+b ·x^3+c · x^2+d ·x+efor listene *X* og *Y* med frekvens *Frekv*. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 150.)

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 202).

Utdata-variabel	**Beskrivelse**
stat.RegEqn	Regresjonsligning: a ·x^4+b ·x^3+c ·x^2+d ·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 *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.YReg	Liste over de datapunkter i den endrede *Y-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.FreqReg	Liste over frekvenser som samsvarer med *stat.XReg* og *stat.YReg*

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([3 -4 2],[0 π/4 1.5])	[0. 2.94771 0.643501]

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*) pararguments-entene.

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([3 -4 2],[0 π/4 1.5])	[3 4.07638 5/2]

▶Rad Katalog >

Value1▶*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*
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.

Angir det tilfeldige tallet for start.

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: Verdiene 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 [μ−3•σ, μ+3•σ].

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*

Returnerer den reelle delen av argumentet.

real(2+3·*i*)	2

real(*List1***)** ⇒ *liste*

Returnerer den reelle delen av alle elementer.

real({1+3·*i*,3,*i*})	{1,3,0}

real(*Matrix1***)** ⇒ *matrise*

Returnerer den reelle delen av alle elementer.

real([1+3·*i* 3; 2 *i*])	[1 3; 2 0]

▶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 115.

$\left(\begin{bmatrix} 3 & \angle\frac{\pi}{4} & \angle\frac{\pi}{6} \end{bmatrix}\right)$ ▶Rect	$\begin{bmatrix} 1.06066 & 1.06066 & 2.59808 \end{bmatrix}$

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.

I Radian-vinkelmodus:

$\left(4\cdot e^{\frac{\pi}{3}}\right)$ ▶Rect	11.3986
$\left(\left(4\angle\frac{\pi}{3}\right)\right)$ ▶Rect	$2.+3.4641\cdot i$

▶Rect

Katalog >

Merk: Du må bruke parentes for å legge inn polar (r∠θ).

I Gradian-vinkelmodus:

$$((1 \angle 100))\blacktriangleright\text{Rect} \qquad i$$

I Grader-vinkelmodus:

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

Merk: For å skrive ∠, velg den fra symbollisten i Katalogen.

ref()

Katalog >

ref(*Matrix1*[, *Tol*]**)** ⇒ *matrise*

Returnerer eliminasjonsformen av *Matrix1*.

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

- Hvis du bruker ctrl enter eller stiller modusen **Auto eller Tilnærmet** til Tilnærmet, utføres beregningene med flyttallsregning.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
  5E−14 •max(dim(*Matrix1*)) •rowNorm (*Matrix1*)

Unngå udefinerte elementer i *Matrix1*. De kan føre til uventede resultater.

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

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

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

ref() Katalog >

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.

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

Merk: Se også **rref()**, page 135.

RefreshProbeVars Katalog >

RefreshProbeVars

Gir deg tilgang til sensordata fra alle tilkoblede sensorer gjennom TI-Basic-programmet.

StatusVar-verdi	Status
statusVar =0	Normal (fortsett med programmet)
statusVar =1	Vernier DataQuest™-applikasjonen er i datainnsamlingsmodus. **Merk:** Vernier DataQuest™-applikasjonen må være i målermodus for at denne kommandoen skal fungere.
statusVar =2	Vernier DataQuest™-applikasjonen er ikke startet.
statusVar =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
```

RefreshProbeVars

Katalog >

```
Disp «ikke klar. Prøv igjen
senere»

EndIf

EndPrgm
```

Merk: Dette kan også brukes med TI-Innovator™-senter.

remain()

Katalog >

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•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 97.

remain(7,0)	7
remain(7,3)	1
remain(⁻7,3)	⁻1
remain(7,⁻3)	1
remain(⁻7,⁻3)	⁻1
remain({12,⁻14,16},{9,7,⁻5})	{3,0,1}

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

Forespør

Katalog >

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*r²
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:

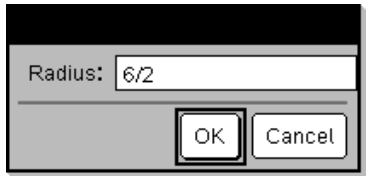
Definer *func*(*arg1*, ...*argn*) = *brukerens svar*

Programmet kan så bruke den definerte funksjonen *funk*(). *promptString* skal veilede brukeren i å legge inn et passende *bruker-svar* som fullfører funksjonsdefinisjonen.

Merk: Kalkulatoren Forespør -kommandoen inne i et brukerdefinert program, men ikke inne i en funksjon.

Slik stopper du et program som inneholder en **Request**-kommando inne i en infinitt løkke:

- **Grafregner:** Hold nede tasten [on], og trykk på [enter] flere ganger.



Resultat etter trykk på **OK**:

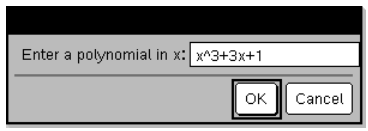
```
Stråle: 6/2
Area-= 28,2743
```

Definere et program:

```
Define 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}
```

Forespør

Katalog >

- **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 130.

ForespørStr

Katalog >

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 [on], og trykk på [enter] 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 128.

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 Katalog >

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

```
Define factorial(nn)=
Func
Local answer,counter
1→answer
For counter,1,nn
answer·counter→answer
EndFor
Return answer
EndFunc
```

```
factorial(3)                    6
```

right() Katalog >

right(*List1*[, *Num*]**)** ⇒ *liste*

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

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

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

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("Hello",2)                "lo"
```

right(*Comparison***)** ⇒ *uttrykk*

Returnerer høyre side av en ligning eller ulikhet.

rk23 () Katalog >

rk23(*Expr*, *Var*, *depVar*, {*Var0*, *VarMax*}, *depVar0*, *VarStep* [, *diftol*]**)** ⇒ *matrise*

rk23(*SystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep*[, *diftol*]**)** ⇒ *matrise*

rk23(*ListOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep*[, *diftol*]**)** ⇒ *matrise*

Differensialligning:

y'=0,001*y*(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

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

med $depVar(Var0)=depVar0$ i intervallet $[Var0, VarMax]$. 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.

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

$ListOfDepVars0$ er en liste over opprinnelige verdier for avhengige variabler.

Hvis $VarStep$ beregnes til et tall som ikke er null: tegn($VarStep$) = tegn($VarMax$-$Var0$) og løsninger returneres ved $Var0$+i*$VarStep$ for alle i=0,1,2,... slik at $Var0$+i*$VarStep$ er i $[var0, VarMax]$ (får kanskje ikke en løsningsverdi ved $VarMax$).

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

Samme ligning med $diftol$ satt til 1.E−6

rk23(0.001·y·(100−y),t,y,{0,100},10,1,1.E-6)

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

System av ligninger:

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

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

$$\text{rk23}\left(\begin{cases} -y1+0.1\cdot y1\cdot y2 \\ 3\cdot y2-y1\cdot y2 \end{cases}, t, \{y1, y2\}, \{0,5\}, \{2,5\}, 1\right)$$

0.	1.	2.	3.	4.
2.	1.94103	4.78694	3.25253	1.82848
5.	16.8311	12.3133	3.51112	6.27245

rk23 ()

Katalog >

diftol er feiltoleransen (grunnverdi på 0,001).

root()

Katalog >

root(*Value***)** ⇒ *rot*
root(*Value1***, *Value2***)** ⇒ *rot*

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

root(*Value1***, *Value2***)** returnerer *Value2*-roten av *Value1*. *Value1* kan være en reell eller kompleks flytende desimalpunktkonstant, 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()

Katalog >

rotate(*Integer1***[,***#ofRotations***])** ⇒ *heltall*

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.

0b00000000000000111101011000011010 1

Bit helt til høyre roterer mot venstre.

produserer:

0b10000000000000011110101100001101 0

I binær grunntall-modus:

rotate(0b11111111111111111111111111111111)	0b10000000000000000000000000000001▶
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)	0h80000000000001E3
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).

rotate() Katalog >

Resultatene vises i forhold til grunntall-modusen.

rotate(*List1*[**,***#ofRotations*]**)** ⇒ *liste*

Returnerer en kopi av *List1* som er rotert mot høyre eller mot venstre av *#ofRotations*-elementer. Endrer ikke *List1*.

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

I desimalgrunntall-modus:

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

rotate(*String1*[**,***#ofRotations*]**)** ⇒ *streng*

Returnerer en kopi av *String1* som er rotert mot høyre eller mot venstre av *#ofRotations*-tegnene. Endrer ikke *String1*.

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

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"bcda"

round() Katalog >

round(*Value1*[**,** *digits*]**)** ⇒ *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(1.234567,3)	1.235

round(*List1*[**,** *digits*]**)** ⇒ *liste*

Returnerer en liste over elementer avrundet til spesifisert antall sifre.

round({π,√2,ln(2)},4)	{3.1416,1.4142,0.6931}

round(*Matrix1*[**,** *digits*]**)** ⇒ *matrise*

Returnerer en matrise over elementer avrundet til spesifisert antall sifre.

round([ln(5) ln(3); π e^1],1)	[1.6 1.1; 3.1 2.7]

rowAdd() Katalog >

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

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

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

rowDim() Katalog >

rowDim(*Matrix***) ⇒** *uttrykk*

Returnerer antallet rader i *Matrix*.

Merk: Se også **colDim()**, side 24.

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

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

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.

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

rowSwap() Katalog >

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

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

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

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

rref() Katalog >

rref(*Matrix1***[, *Tol*]) ⇒** *matrise*

Returnerer den reduserte eliminasjonsformen av *Matrix1*.

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

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 ctrl enter eller stiller modusen **Auto eller Tilnærmet** til Tilnærmet, utføres beregningene med flyttallsregning.
- Hvis *Tol* utelates eller ikke blir brukt, blir grunninnstillingstoleransen beregnet som:
  5E−14 •max(dim(*Matrix1*)) •rowNorm (*Matrix1*)

Merk: Se også **ref()**, page 126.

S

sec() trig tast

sec(*Verdi1***)** ⇒ *verdi*

sec(*Liste1***)** ⇒ *liste*

Returnerer sekans til *Verdi1* eller returnerer en liste med sekans til hvert element i *Liste1*.

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

I Grader-vinkelmodus:

sec(45)	1.41421
sec({1,2.3,4})	{1.00015,1.00081,1.00244}

sec⁻¹ () trig tast

sec⁻¹(*Verdi1***)** ⇒ *verdi*

sec⁻¹(*Liste1***)** ⇒ *liste*

Returnerer vinkelen som har sekans lik *Verdi1* eller returnerer en liste med inverse sekanser til hvert element i *Liste1*.

I Grader-vinkelmodus:

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

$\sec^{-1}()$ — trig tast

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 **`arcsec(...)`**.

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

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

sech() — Katalog >

sech(*Verdi1***)** ⇒ *verdi*

sech(*Liste1***)** ⇒ *liste*

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

$\text{sech}(3)$	0.099328
$\text{sech}(\{1,2.3,4\})$	$\{0.648054,0.198522,0.036619\}$

$\text{sech}^{-1}()$ — Katalog >

sech^{-1}(*Verdi1***)** ⇒ *verdi*

sech^{-1} (*Liste1***)** ⇒ *liste*

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(...)`**.

I Radian-vinkelmodus og Rectangular-kompleksmodus:

$\text{sech}^{-1}(1)$	0
$\text{sech}^{-1}(\{1,-2,2.1\})$	$\{0,2.0944\cdot i,8.\text{E}-15+1.07448\cdot i\}$

Send — Hub-meny

Send *uttrEllerStreng1*[, *uttrEllerStreng2*] ...

Programmeringskommando: Sender én eller flere TI-Innovator™ Hub kommandoer til en tilkoblet hub.

Eksempel: Slå på det blå elementet i den innebygde RGB LED-skjermen i 0,5 sekunder.

```
Send "SET COLOR.BLUE ON TIME .5"
                                Done
```

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

Send — Hub-meny

uttrEllerStreng må være en gyldig TI-Innovator™ Hub kommando. Vanligvis inneholder *uttrEllerStreng* en **"INNSTILL ..."** -kommandofor å kontrollere en enhet eller en **«LES ...»**-kommando for å etterspørre data.

Argumentene sendes suksessivt til hubben.

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

Merk: Se også **Get** (side 60), **GetStr** (side 67), og **eval()** (side 48).

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

Eksempel: Send en kalkulert frekvens til hubbens innebygde høyttaler. Bruk den spesielle variabelen *iostr.SendAns* for å vise hubkommandoen med uttrykket som er evaluert.

$n:=50$	50
$m:=4$	4
Send "SET SOUND eval(m·n)"	*Done*
iostr.SendAns	"SET SOUND 200"

seq() (sekv) — Katalog >

seq(*Uttr*, *Var*, *Lav*, *Høy*(, *Trinn*)**)**⇒*liste*

Øker *Var* fra *Lav* til *Høy* med trinn på *Intervall*, behandler *Uttr* og returnerer resultatene som en liste. Det opprinnelige innholdet i *Var* er fremdeles der etter at **seq()** er fullført.

Grunnverdien for *Intervall* = 1.

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

Merk: For å tvinge frem et tilnærmet desimalresultat,

Grafregner: Trykk på [ctrl] [enter].
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() — Katalog >

seqGen(*Uttr*, *Var*, *avhVar*, {*Var0*, *VarMaks*}[, *ListeAvInnlLedd* [, *VarIntervall* [,*Loftverdi*]]]**)** ⇒*liste*

Genererer de første 5 leddene i tallfølgen $u(n) = u(n\text{-}1)^2/2$, med $u(1)$=**2** og *VarIntervall*=**1**.

seqGen() Katalog >

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* [**,** *Loftverdi*]]]**)** ⇒*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**.

$$\text{seqGen}\left(\frac{(u(n-1))^2}{n},n,u,\{1,5\},\{2\}\right)$$

$$\left\{2,2,\frac{4}{3},\frac{4}{9},\frac{16}{405}\right\}$$

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:

$$\text{seqGen}\left(\left\{\frac{1}{n},\frac{u2(n-1)}{2}+u1(n-1)\right\},n,\{u1,u2\},\{1,5\},\left[\frac{_}{2}\right]\right)$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix}$$

Merk: Tomrommet (_) i matrisen med innledende ledd over brukes for å angi at det innledende leddet for u1(n) er beregnet ved hjelp av den eksplisitte tallfølgeformelen u1 (n)=1/n.

seqn() Katalog >

seqn(*Uttr***(***u***,** *n* [**,** *ListeMedInnlLedd*[**,** *nMaks* [**,** *Loftverdi*]]])**)**⇒*liste*

Genererer en liste over ledd for tallfølge *u* (*n*)=*Uttr*(*u*, *n*) som følger: Øker *n* fra 1 til *nMaks* med 1, beregner *u*(*n*) for tilsvarende verdier av *n* ved hjelp av formel *Uttr*(*u*, *n*) og *ListeMedInnlLedd*, og returnerer resultatene som en liste.

seqn(*uttr***(***n* [**,** *nMaks* [**,** *Loftverdi*]])**)**⇒*liste*

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

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

$$\left\{2,1,\frac{1}{3},\frac{1}{12},\frac{1}{60},\frac{1}{360}\right\}$$

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

seqn() Katalog >

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

Hvis $nMaks$ mangler, innstilles $nMaks$ på 2500

Hvis $nMax$=0, innstilles $nMaks$ på 2500

Merk: seqn() kaller **seqGen( )** med $n0$=**1** og $nintervall$ =**1**

SetMode() (lesModus) Katalog >

SetMode(*modusNavnHeltall***,** *innstillingHeltall***)** ⇒*heltall*

SetMode(*liste***)** ⇒*heltallsliste*

Kun gyldig innenfor en funksjon eller et program.

SetMode(*modusNavnHeltall***,** *innstillingHeltall***)** setter foreløpig modus *modusNavnHeltall* til den nye innstillingen *innstillingHeltall*, og returnerer et heltall som samsvarer med den opprinnelige innstillingen av den modusen. Endringen er begrenset til hvor lenge det varer å utføre programmet/funksjonen.

modusNavnHeltall spesifiserer hvilken modus du vil stille inn. Det må være en av modus-heltallene fra tabellen nedenfor.

innstilleHeltall spesifiserer den nye innstillingen for modusen. Det må være en av innstillingsheltallene fra listen nedenfor for den spesifikke modusen som du stiller inn.

Vis tilnærmet verdi av π ved hjelp av grunninnstillingen for Vis sifre, og vis så π med en innstilling på Fast2. Kontroller for å se at grunninnstillingen gjenopprettes etter at programmet utføres.

```
Define prog1()=Prgm                    Done
               Disp π
               setMode(1,16)
               Disp π
               EndPrgm
prog1()
                                   3.14159
                                   3.14
                                      Done
```

SetMode() (lesModus) Katalog >

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 66.

Merk: De aktuelle modusinnstillingene sendes til påkalte underrutiner. Hvis en underrutine 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
Eksponensielt format	3	**1**=Normal, **2**=Vitenskapelig, **3**=Teknisk
Reell eller kompleks	4	**1**=Reell, **2**=Rektangulær, **3**=Polar
Auto eller tilnærm.	5	**1**=Auto, **2**=Tilnærmet
Vektorformat	6	**1**=Rektangulær, **2**=Sylindrisk, **3**=Sfærisk
Grunntall	7	**1**=Desimal, **2**=Heks, **3**=Binær

shift(*Heltall1*[,*#avSkift*])⇒*heltall*

Forskyver (skifter) bitene i et binært heltall. Du kan legge inn *Heltall1* med hvilket som helst grunntall. Det konverteres automatisk til 64-bit binær form med fortegn. Hvis *Heltall1* er for stort for denne formen, vil en symmetrisk modulusoperasjon bli brukt til å konvertere tallet inn i gyldig verdiområde. For mer informasjon, se ▶**Base2**, side 16.

Hvis *#avSkift* er positiv, er skift til venstre. Hvis *#avSkift* er negativ, er skift til høyre. Grunninnstilling er ⁻1 (skiftes èn 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.

0b00000000000000111101011000011010

Setter inn 0 hvis biten helt til venstre er 0,

eller 1 hvis biten helt til venstre er 1.

produserer:

0b00000000000000011110101100001101 0

Resultatene vises i forhold til grunntall-modusen. Ledende nuller vises ikke.

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).

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 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 "udef".

shift(*Streng1* [**,***#avSkift*]**(⇒***streng*

Returnerer en kopi av *Streng1* skiftet mot høyre eller mot venstre av *#ofShifts*-tegn. Endrer ikke *Streng1*.

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(*Verdi1***)⇒***verdi*

sign(*Liste1***)⇒***list e*

sign(*Matrise1***)⇒***matrise*

For reell og komplekst *Verdi1*, returnerer *Verdi1* / **abs(***Verdi1***)** når *Verdi1* ≠ 0.

Returnerer 1 hvis *Verdi1* er positiv.

Returnerer ⁻1 hvis *Verdi1* 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(*koeffMatrise***,** *konstVektor***(,** *Tol***))** ⇒*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 84.

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 ·max(dim(*koeffMatrise*))
  ·radNavn*koeffMatrise*)

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

Løsningen er x=⁻3 og y=2.

Løs:

ax + by = 1

cx + dy = 2

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

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

simult(*koeffMatrise***,** *konstMatrise***(,** *Tol***))** ⇒*matrise*

Løser multiple systemer av lineære ligninger, hvor hvert system har samme ligningskoeffisienter men forskjellige konstanter.

Hver kolonne i *konstMatrise* må inneholde konstantene for et ligningssystem. Hver kolonne i resultatmatrisen inneholder løsningen for det tilsvarende systemet.

Løs:

x + 2y = 1

3x + 4y = ⁻1

x + 2y = 2

3x + 4y = ⁻3

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

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

sin()

trig tast

sin(*Verdi1***)⇒***verdi*

sin(*Liste1***)⇒***liste*

sin(*Verdi1***)** returnerer sinus til argumentet.

sin(*Liste1***)** returnerer en liste over sinus til alle elementer i *Liste1*.

Merk: Argumentet tolkes som grader, gradianer eller som radianer, avhengig av aktuell vinkelmodus. Du kan bruke °, 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
$\sin(45°)$	0.707107

sin(*kvadratMatrise1***)⇒***kvadratMatrise*

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

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

$$\begin{bmatrix}0.9424 & -0.04542 & -0.031999\\ -0.045492 & 0.949254 & -0.020274\\ -0.048739 & -0.00523 & 0.961051\end{bmatrix}$$

$\sin^{-1}()$

trig tast

sin⁻¹(*Liste1***)⇒***liste*

sin⁻¹(*Verdi1***)** returnerer vinkelen med sinus lik *Verdi1*.

sin⁻¹(*Liste1***)** returnerer en liste over invers sinus til hvert element i *Liste1*.

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

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

I Grader-vinkelmodus:

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

I Gradian-vinkelmodus:

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

I Radian-vinkelmodus:

sin⁻¹() trig tast

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

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

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I 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(*Verdi1***)⇒***verdi***

sinh(*Liste1***)⇒***liste***

sinh (*Verdi1***)** returerer hyperbolsk sinus til argumentet.

sinh (*Liste1***)** returnerer en liste over hyperbolsk sinus til hvert element i *Liste1*.

$\sinh(1.2)$ 1.50946

$\sinh(\{0,1.2,3.\})$ $\{0,1.50946,10.0179\}$

sinh(*kvadratMatrise1***)⇒***kvadratMatrise***

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Radian-vinkelmodus:

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

sinh⁻¹() Katalog >

sinh⁻¹(*Verdi1***)⇒***verdi***

sinh⁻¹(*Liste1***)⇒***liste***

sinh⁻¹(*Verdi1***)** returnerer invers hyperbolsk sinus til argumentet.

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

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

$\sinh^{-1}(\{0,2.1,3\})$ $\{0,1.48748,1.81845\}$

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

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:

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

$$\begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

SinReg katalog >

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 150.)

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..

SinReg katalog >

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 202).

Utdata-variabel	Beskrivelse
stat.RegEqn	Regresjonsligning: a ·sin(bx+c)+d
stat.a, stat.b, stat.c, stat.d	Regresjonskoeffisienter
stat.Rest	Residualene fra regresjonen
stat.XReg	Liste over de datapunkter i den endrede *X-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.YReg	Liste over de datapunkter i den endrede *Y-listen* som faktisk brukes i regresjonen basert på begrensninger i *Frekv*, *Kategoriliste* og *Inkluder kategorier*
stat.FreqReg	Liste over frekvenser som samsvarer med *stat.XReg* og *stat.YReg*

SortA (SorterSt) Katalog >

SortA*Liste1*[, *Liste2*] [, *Liste3*]...

SortA*Vektor1*[, *Vektor2*] [, *Vektor3*] ...

Sorterer elementene i det første argument i stigende rekkefølge.

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.

Alle argumentene må være navn på lister eller vektorer. Alle argumentene må ha like dimensjoner.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
SortA *list1*	*Done*
list1	$\{1,2,3,4\}$
$\{4,3,2,1\} \rightarrow list2$	$\{4,3,2,1\}$
SortA *list2*,*list1*	*Done*
list2	$\{1,2,3,4\}$
list1	$\{4,3,2,1\}$

SortA (SorterSt) Katalog >

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 202.

SortD (SorterSy) Katalog >

SortD *Liste1*[, *Liste2*] [, *Liste3*] ...

SortD *Vektor1*[,*Vektor2*] [,*Vektor3*] ...

Identisk med **SortA**, bortsett fra at **SortD** sorterer elementene i fallende rekkefølge.

Tomme (åpne) elementer innenfor det første utsagnet flyttes til bunnen. For mer informasjon om tomme elementer, se side 202.

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
SortD *list1*,*list2*	*Done*
list1	$\{4,3,2,1\}$
list2	$\{3,4,1,2\}$

▸Sphere (sfærisk) Katalog >

Vektor **▸Sphere**

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

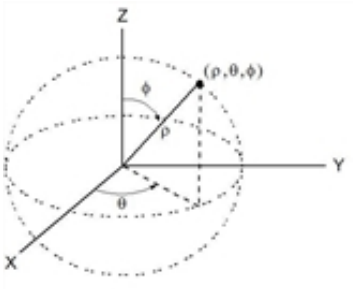
Viser rad- eller kolonnevektor i sfærisk form [ρ ∠θ ∠ϕ].

Vektor må være av dimensjon 3 og kan enten være en rad- eller en kolonnevektor.

Merk: ▸Sphere er en visningsformat-instruksjon, ikke en omregningsfunksjon. Du kan bare bruke den på slutten av en kommandolinje.

$$[1 \quad 2 \quad 3] \blacktriangleright \text{Sphere}$$
$$[3.74166 \quad \angle 1.10715 \quad \angle 0.640522]$$

$$\left(\begin{bmatrix} 2 & \angle\frac{\pi}{4} & 3 \end{bmatrix}\right) \blacktriangleright \text{Sphere}$$
$$[3.60555 \quad \angle 0.785398 \quad \angle 0.588003]$$



sqrt() (kvdrt) Katalog >

sqrt(*Verdi1***)⇒***verdi*

sqrt(*Liste1***)⇒***liste*

Returnerer kvadratroten til argumentet.

For en liste, returneres kvadratroten til alle elementene i *Liste1*.

Merk: Se også **Kvadratrot-sjablon**, side 1.

$\sqrt{4}$	2
$\sqrt{\{9,2,4\}}$	$\{3,1.41421,2\}$

stat.results (stat.resultats) 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 analyse.

xlist:={1,2,3,4,5}	{1,2,3,4,5}
ylist:={4,8,11,14,17}	{4,8,11,14,17}

LinRegMx *xlist*,*ylist*,1: *stat.results*

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"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.AdjR²	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.r²	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.σx	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.σy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.σx1	stat.UpperPred

stat.b7	stat.ExpMatrix	stat.MSRow	stat.σx2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.$\bar{x}$
stat.b9	stat.FBlock	stat.$\hat{p}$	stat.Σx²	stat.$\bar{x}$1
stat.b10	stat.Fcol	stat.$\hat{p}$1	stat.Σxy	stat.$\bar{x}$2
stat.bList	stat.FInteract	stat.$\hat{p}$2	stat.Σy	stat.$\bar{x}$Diff
stat.χ^2	stat.FreqReg	stat.$\hat{p}$Diff	stat.Σy²	stat.$\bar{x}$List
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg
stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat.$\bar{y}$
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat.$\hat{y}$
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat.$\hat{y}$List
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

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

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.

Se **stat.results** -eksemplet.

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 202

$\text{stDevPop}(\{1,2,5,-6,3,-2\})$ 3.59398

$\text{stDevPop}(\{1.3,2.5,-6.4\},\{3,2,5\})$ 4.11107

stDevPop(*Matrise1***[, *FrekvMatrise*])** ⇒*matrise*

Returnerer en radvektor av populasjonens standardavvik i kolonnene i *Matrise1*.

Hvert *frekvMatrise* element teller antallet forekomster av det tilsvarende elementet i Matrise1.

Merk: *Matrise1* må ha minst to rader. Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 202.

$$\text{stDevPop}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right)$$
$$\begin{bmatrix} 3.26599 & 2.94392 & 1.63299 \end{bmatrix}$$

$$\text{stDevPop}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix},\begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right)$$
$$\begin{bmatrix} 2.52608 & 5.21506 \end{bmatrix}$$

stDevSamp() (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 202

$\text{stDevSamp}(\{1,2,5,-6,3,-2\})$ 3.937

$\text{stDevSamp}(\{1.3,2.5,-6.4\},\{3,2,5\})$ 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 202.

$$\text{stDevSamp}\left(\begin{bmatrix}1 & 2 & 5\\ -3 & 0 & 1\\ 5 & 7 & 3\end{bmatrix}\right)$$

$$\begin{bmatrix}4. & 3.60555 & 2.\end{bmatrix}$$

$$\text{stDevSamp}\left(\begin{bmatrix}-1.2 & 5.3\\ 2.5 & 7.3\\ 6 & -4\end{bmatrix},\begin{bmatrix}4 & 2\\ 3 & 3\\ 1 & 7\end{bmatrix}\right)$$

$$\begin{bmatrix}2.7005 & 5.44695\end{bmatrix}$$

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 produkthåndboken.

```
i:=0                                   0
Define prog1()=Prgm                    Done
              For i,1,10,1
              If i=5
              Stop
              EndFor
              EndPrgm
prog1()                                Done
i                                      5
```

Lagre Se → (lagre), side 200.

String() (Streng) Katalog >

string(*Uttr***)**⇒*string*

Forenkler *Uttr* og returnerer resultatet som en tegnstreng.

```
string(1.2345)                         "1.2345"
string(1+2)                            "3"
```

subMat() (undermatrise) Katalog >

subMat(*Matrise1*[, *startRad*] [, *startKol*] [, *endRad*) [, *endKol*]) ⇒*matrise*

Returnerer den spesifiserte undermatrisen til *Matrise1*.

Grunninnstillinger: *startRad*=1, *startKol*=1, *endRad*=siste rad, *endKol*=siste kolonne.

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

$$\text{subMat}(m1,2,1,3,2) \qquad \begin{bmatrix}4&5\\7&8\end{bmatrix}$$

$$\text{subMat}(m1,2,2) \qquad \begin{bmatrix}5&6\\8&9\end{bmatrix}$$

Sum (Sigma) Se G(), side 192.

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 202.

$$\text{sum}(\{1,2,3,4,5\}) \qquad 15$$

$$\text{sum}(\{a,2\cdot a,3\cdot a\})$$

"Error: Variable is not defined"

$$\text{sum}(\text{seq}(n,n,1,10)) \qquad 55$$

$$\text{sum}(\{1,3,5,7,9\},3) \qquad 21$$

sum(*Matrise1*[, *Start*[, *Slutt*]])⇒*matrise*

Returnerer en radvektor som inneholder summene av elementene i kolonnene i *Matrise1*.

Start og *Slutt* er alternativer. De spesifiserer et radområde.

Et tomt utsagn produserer et tomt resultat. Tomme (åpne) elementer i *Matrise1* ignoreres. For mer informasjon om tomme elementer, se side 202.

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

$$\text{sum}\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\right) \qquad \begin{bmatrix}12&15&18\end{bmatrix}$$

$$\text{sum}\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix},2,3\right) \qquad \begin{bmatrix}11&13&15\end{bmatrix}$$

sumIf() Katalog >

sumIf(*Liste*,*Kriterium*[, *SumListe*])⇒*verdi*

$$\text{sumIf}(\{1,2,\mathbf{e},3,\pi,4,5,6\},2.5<?<4.5)$$

12.859874482

$$\text{sumIf}(\{1,2,3,4\},2<?<5,\{10,20,30,40\})$$

70

sumIf() Katalog >

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 llster og regneark kan du bruke et celleområde istedenfor *Liste* og *sumListe*.

Tomme (åpne) elementer ignoreres. For mer informasjon om tomme elementer, se side 202.

Merk: Se også **countIf()**, side 31.

sumSeq() Se Σ(), side 192.

system() Katalog >

system(*Verdi1* [**,** *Verdi2* [**,** *Verdi3* [**,** ...]]]**)**

Returnerer et ligningssystem, formatert som en liste. Du kan også opprette et system med en sjablon.

T(transponert) katalog >

Matrise1^T⇒*matrise*

Returnerer den komplekse konjugerte transponerte av *Matrise1*.

Merk: Du kan sette inn denne operatoren fra datamaskintastaturet ved å skrive `@t`.

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

tan() trig tast

tan(*Verdi1***)**⇒*verdi*

tan(*Liste1***)**⇒*liste*

tan(*Verdi1***)** returnerer tangens til argumentet.

tan(*Liste1***)** returnerer en liste over tangensene til alle elementene i *Liste1*.

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

I Grader-vinkelmodus:

$$\tan\left(\left(\frac{\pi}{4}\right)^r\right) \qquad 1.$$

$$\tan(45) \qquad 1.$$

$$\tan(\{0,60,90\}) \qquad \{0.,1.73205,\text{undef}\}$$

I Gradian-vinkelmodus:

$$\tan\left(\left(\frac{\pi}{4}\right)^r\right) \qquad 1.$$

$$\tan(50) \qquad 1.$$

$$\tan(\{0,50,100\}) \qquad \{0.,1.,\text{undef}\}$$

I Radian-vinkelmodus:

$$\tan\left(\frac{\pi}{4}\right) \qquad 1.$$

$$\tan(45°) \qquad 1.$$

$$\tan\left(\left\{\pi,\frac{\pi}{3},-\pi,\frac{\pi}{4}\right\}\right) \qquad \{0.,1.73205,0.,1.\}$$

tan(*kvadratMatrise1***)**⇒*kvadratMatrise*

Returnerer matrisetangensen av *kvadratMatrise1*. Dette er ikke det samme som å beregne tangens for hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

I Radian-vinkelmodus:

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

$$\begin{bmatrix}-28.2912&26.0887&11.1142\\12.1171&-7.83536&-5.48138\\36.8181&-32.8063&-10.4594\end{bmatrix}$$

tan() trig tast

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

$\tan^{-1}()$ trig tast

$\tan^{-1}$(*Verdi1***)⇒***verdi***

$\tan^{-1}$(*Liste1***)⇒***liste***

$\tan^{-1}$(*Verdi1***)** returnerer vinkelen med tangens til *Verdi1***.**

$\tan^{-1}$(*Liste1***)** returnerer en liste over de inverse tangenser til hvert element i *Liste1*.

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

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

$\tan^{-1}$(*kvadratMatrise1***)⇒***kvadratMatrise***

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

kvadratMatrise1 må kunne diagonaliseres. Resultatet inneholder alltid flytende desimaltall.

I Grader-vinkelmodus:

$\tan^{-1}(1)$	45

I Gradian-vinkelmodus:

$\tan^{-1}(1)$	50

I Radian-vinkelmodus:

$\tan^{-1}(\{0,0.2,0.5\})$	$\{0,0.197396,0.463648\}$

I Radian-vinkelmodus:

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

$$\begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

tanh() Katalog >

tanh(*Verdi1***)⇒***verdi***

tanh(*Liste1***)⇒***liste***

tanh(*Verdi1***)** returnerer hyperbolsk tangens til argumentet.

tanh(*Liste1***)** returnerer en liste av hyperbolske tangenser til hvert element i *Liste1*.

tanh(*kvadratMatrise1***)⇒***kvadratMatrise***

$\tanh(1.2)$	0.833655
$\tanh(\{0,1\})$	$\{0.,0.761594\}$

I Radian-vinkelmodus:

tanh() Katalog >

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.

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

$$\begin{bmatrix}-0.097966 & 0.933436 & 0.425972\\ 0.488147 & 0.538881 & -0.129382\\ 1.28295 & -1.03425 & 0.428817\end{bmatrix}$$

$\tanh^{-1}()$ Katalog >

$\tanh^{-1}$(*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(...)`**.

I rektangulært, kompleks format:

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

$\tanh^{-1}(\{1,2.1,3\})$

$\{\text{undef},0.518046-1.5708\cdot i,0.346574-1.57$

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

$\tanh^{-1}$(*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 radian-vinkelmodus og rektangulært, kompleks format:

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

$$\begin{bmatrix}-0.099353+0.164058\cdot i & 0.267834-1.4908\\ -0.087596-0.725533\cdot i & 0.479679-0.9473\\ 0.511463-2.08316\cdot i & -0.878563+1.7901\end{bmatrix}$$

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

tCdf() Katalog >

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 *øvGrense* for spesifisert grader av frihet *df*.

tCdf() Katalog >

For P(X ≤ *øvreGrense*), sett *nedreGrense* = −9E999.

Text Katalog >

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 128 eller **RequestStr**, side 130.

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.

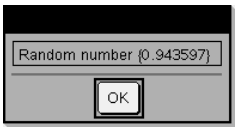
Innenfor malen Prgm...EndPrgm fullfører du hver linje ved å trykke på ↵ istedenfor på enter. På tastaturet på datamaskinen, hold nede **Alt** og trykk på **Enter**.

```
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) Se If, side 70.

tInterval katalog >

tInterval *Liste*[**,***Frekv*[**,***CNivå*]]

(Dataliste-inndata)

tInterval $\bar{x}$**,***sx***,***n***[,***CNivå***]**

tInterval katalog >

(Oppsummerende statistikk-inndata)

Beregner et *t*-konfidensintervall. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 150.)

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 202).

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.σx	Utvalgets standardavvik
stat.n	Lengde av datasekvensen med utvalgsgjennomsnitt

tInterval_2Samp Katalog >

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 150).

Felles=**1** fellesvarianser; *Felles*=**0** gjør ikke fellesvarianser.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 202).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall som inneholder konfidensnivå-sannsynligheten for en fordeling

Utdata-variabel	Beskrivelse
stat.$\bar{x}$1-$\bar{x}$2	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.df	Grader-av-frihet
stat.$\bar{x}$ 1, stat.$\bar{x}$2	Utvalgsgjennomsnitt av datasekvensene fra normal tilfeldig fordeling
stat.σx1, stat.σx2	Utvalgets standardavvik for *Liste 1* og *Liste 2*
stat.n1, stat.n2	Antall utvalg i datasekvenser
stat.sp	Det felles standardavviket. Beregnet når *Felles* = JA.

tPdf() Katalog >

tPdf(*XVerd***,***df***)⇒***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***)⇒***verdi*

Returnerer diagonalsummen (summen av alle elementene på hoveddiagonalen) til *kvadratMatrise*.

$\text{trace}\left(\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\right)$	15
$a:=12$	12
$\text{trace}\left(\begin{bmatrix}a&0\\1&a\end{bmatrix}\right)$	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 209.

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.

```
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
```

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.

$$eigenvals\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

Merk: Se også **ClrErr**, side 23, og **PassErr**, side 113.

```
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
```

Try

Katalog >

EndTry

EndPrgm

tTest

Katalog >

tTest $\mu 0$,*Liste*[,*Frekv*[,*Hypot*]]

(Dataliste inndata)

tTest $\mu 0$,$\bar{x}$,*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 150).

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: $\mu > \mu 0$, set *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 202).

Utdata-variabel	Beskrivelse
stat.t	$(\bar{x} - \mu 0)$ / (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 *Liste*
stat.sx	Utvalgets standardavvik av datasekvensen
stat.n	Utvalgenes størrelse

tTest_2Samp

Katalog >

tTest_2Samp *Liste1*,*Liste2*[,*Frekv1*[,*Frekv2*[,*Hypot*[,*Felles*]]]]

tTest_2Samp Katalog >

(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 150).

Test H_0: $\mu1 = \mu2$, mot ett av følgende:

For H_a: $\mu1 < \mu2$, sett *Hypot*<0

For H_a: $\mu1 \neq \mu2$ (standard), sett *Hypot*=0

For H_a: $\mu1 > \mu2$, 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 202).

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 *Liste*1 *og* Liste2
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i *Liste* 1 *og* Liste2
stat.n1, stat.n2	Utvalgenes størrelse
stat.sp	Det felles standardavviket. Beregnet når *Felles*=1.

tvmFV() Katalog >

tvmFV(*N,I,PV,Pmt*,[*PpY*],[*CpY*],[*PmtAt*]**)** ⇒*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 166. Se også **amortTbl()**, side 7.

tvmI() Katalog >

tvmI(*N***,***PV***,***Pmt***,***FV***,**[*PpY*]**,**[*CpY*]**,**[*PmtAt*]**)** ⇒*verdi*

Finansiell funksjon som beregner rentefoten per år.

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 166. Se også **amortTbl()**, side 7.

```
tvmI(240,100000,-1000,0,12,12)        10.5241
```

tvmN() Katalog >

tvmN(*I***,***PV***,***Pmt***,***FV***,**[*PpY*]**,**[*CpY*]**,**[*PmtAt*]**)** ⇒*verdi*

Finansiell funksjon som beregner antallet betalingsperioder.

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 166. Se også **amortTbl()**, side 7.

```
tvmN(5,0,-500,77641,12,12)            120.
```

tvmPmt() Katalog >

tvmPmt(*N***,***I***,***PV***,***FV***,**[*PpY*]**,**[*CpY*]**,**[*PmtAt*]**)** ⇒*verdi*

Finansiell funksjon som beregner beløpet for hver betaling.

Merk: Argumenter som brukes i TVM-funksjonene er beskrevet i tabellen over TVM-argumenter, side 166. Se også **amortTbl()**, side 7.

```
tvmPmt(60,4,30000,0,12,12)            -552.496
```

tvmPV() Katalog >

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 166. Se også **amortTbl()**, side 7.

TVM-argument*	Beskrivelse	Datatype
N	Antall betalingsperioder	reelt tall
I	Årlig rente (rentefot)	reelt tall
PV	Nåverdi	reelt tall
Pmt	Betalingsbeløp	reelt tall
FV	Fremtidig verdi	reelt tall
PpY	Antall betalinger pr. år, grunninnstilling=1	heltall > 0
CpY	Antall renteperioder pr. år, grunninnstilling=1	heltall > 0
PmtAt	Betaling som forfaller ved slutten eller begynnelsen av hver periode, grunninnstilling=avslutt	heltall (0=avslutte, 1=begynne)

* Disse tidsverdi-for-penger-argumentnavnene likner TVM-variabelnavnene som f.eks. **tvm.pv** og **tvm.pmt**) som brukes av *Calculator* applikasjonens finansløser.Men finansielle funksjoner lagrer ikke argumentverdiene eller resultatene til TVM-variablene.

TwoVar Katalog >

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

Beregner 2-variabels statistiske observatorer. En oversikt over resultatene lagres i *stat.resultater*-variabelen. (Se side 150.)

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 202.

Utdata-variabel	**Beskrivelse**
stat.$\bar{x}$	Gjennomsnitt av x-verdier
stat. x	Sum av x-verdier
stat. x2	Sum av x2 verdier
stat.sx	Utvalgets standardavvik til x
stat. x	Populasjonens standardavvik til x
stat.n	Antall datapunkter
stat.$\bar{y}$	Gjennomsnitt av y-verdier
stat. y	Sum av y-verdier
stat. y^2	Sum av y2-verdier
stat.sy	Utvalgets standardavvik til y
stat. y	Populasjonens standardavvik til y
stat. xy	Sum av x ·y -verdier
stat.r	Korrelasjonskoeffisient

Utdata-variabel	Beskrivelse
stat.MinX	Minimum av x-verdier
stat.Q_1X	Første kvartil av x
stat.MedianX	Median av x
stat.Q_3X	Tredje kvartil av x
stat.MaxX	Maksimum av x-verdier
stat.MinY	Minimum av y-verdier
stat.Q_1Y	Første kvartil av y
stat.MedY	Median av y
stat.Q_3Y	Tredje kvartil av y
stat.MaxY	Maksimum av y-verdier
stat. $(x-)^2$	Sum av kvadratavvik fra gjennomsnittet av x
stat. $(y-)^2$	Sum av kvadrat for avvik fra gjennomsnittet av y

U

unitV() (enhetsV) Katalog >

unitV(*Vektor1***)⇒***vektor*

Returnerer enten en rad- eller kolonne-enhetsvektor, avhengig av formen på *Vektor1*.

Vektor1 må være enten en enkel-rad-matrise eller en enkel-kolonne-matrise.

unitV([1 2 1])	[0.408248 0.816497 0.408248]
unitV([1; 2; 3])	[0.267261; 0.534522; 0.801784]

unLock Katalog >

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

unLock*Var***.**

Låser opp spesifisert variabel eller variabelgruppe. Låste variabler kan ikke modifiseres eller slettes.

Se **Lock**, side 88, og**getLockInfo()**, side 66.

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*

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), ignoreres dette elementet, og det tilsvarende elementet i den andre listen ignoreres også. For mer informasjon om tomme elementer, se side 202.

$\text{varPop}(\{5,10,15,20,25,30\})$ 72.9167

varSamp() (utvalgets varians) Katalog >

varSamp(*Liste*[**,** *frekvListe*]**)⇒***uttrykk*

Returnerer utvalgets varians for *Liste*.

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

Merk: *Liste* må inneholde minst to elementer.

Hvis et element i en av listene er tomt (åpent), ignoreres dette elementet, og det tilsvarende elementet i den andre listen ignoreres også. For mer informasjon om tomme elementer, se side 202.

$\text{varSamp}(\{1,2,5,{}^{-}6,3,{}^{-}2\})$ $\frac{31}{2}$

$\text{varSamp}(\{1,3,5\},\{4,6,2\})$ $\frac{68}{33}$

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.

$\text{varSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}\right)$ $\begin{bmatrix} 4.75 & 1.03 & 4 \end{bmatrix}$

$\text{varSamp}\left(\begin{bmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{bmatrix},\begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}\right)$ $\begin{bmatrix} 3.91731 & 2.08411 \end{bmatrix}$

varSamp() (utvalgets varians) Katalog >

Hvis et element i en av matrisene er tomt (åpent), ignoreres dette elementet, og det tilsvarende elementet i den andre matrisen ignoreres også. For mer informasjon om tomme elementer, se side 202.

Merk: *Matrise1* må inneholde minst to rader.

W

Wait Katalog >

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 [on], og trykk på [enter] 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: 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 ()

Katalog >

warnCodes(*Uttr1***,** *StatusVar***)⇒***uttrykk*

Behandler uttrykk *Uttr1*, returnerer resultatet, og lagrer kodene for alle genererte varsler i listevariabelen *StatusVar*. Hvis ingen varsler er generert, tildeler denne funksjonen *StatusVar* en tom liste.

Uttr1 kan være et hvilket som helst gyldig matematisk uttrykk i TI-Nspire™ eller TI-Nspire™ CAS. Du kan ikke bruke en kommando eller tildeling som *Uttr1*.

StatusVar må være et gyldig variabelnavn.

Se side 217 for en liste over varselkoder og assosierte meldinger.

warnCodes(det([1.23456E-999]),*warn*)	1.23456E-999
warn	{10029}

when() (når)

Katalog >

when(*Betingelse***,** *santResultat* **[,** *usantResultat***][,** *ukjentResultat***])**
⇒*uttrykk*

Returnerer *santResultat*, *usantResultat*, eller *ukjentResultat*, avhengig av om *Betingelse* er sann, usann eller ukjent. Returnerer inndata hvis det er for få argumenter til å spesifisere korrekt resultat.

Utelat både *usantResultat* og *ukjentResultat* for å definere et uttrykk bare i det området der *Betingelse* er sann.

when(*x*<0,*x*+3)|*x*=5	undef

Bruk et **udef** *usantResultat* for å definere et uttrykk som bare plotter grafen på et intervall.

when() er nyttig for å definere rekursive funksjoner.

when(*n*>0,*n*·*factoral*(*n*−1),1)→*factoral*(*n*)	*Done*
factoral(3)	6
3!	6

While

Katalog >

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 sum_of_recip(n)=Func
                       Local i,tempsum
                       1→i
                       0→tempsum
                       While i≤n
                       tempsum+1/i→tempsum
                       i+1→i
                       EndWhile
                       Return tempsum
                       EndFunc
                                          Done
sum_of_recip(3)                           11/6
```

X

xor (enten ...eller ...)

Katalog >

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 111.

```
true xor true                    false
5>3 xor 3>5                      true
```

Heltall1 **xor** *Heltall2* ⇒ *heltall*

I heksades grunntall-modus:

Viktig: Null, ikke bokstaven O.

```
0h7AC36 xor 0h3D5F               0h79169
```

I binær grunntall-modus:

```
0b100101 xor 0b100               0b100001
```

xor (enten ...eller ...) Katalog >

Sammenlikner to reelle heltall bit-for-bit med en **xor** -handling. Internt er begge heltallene omregnet til 64-biters binære tall med fortegn. Når tilsvarende biter sammenliknes, er resultatet 1 hvis en av bitene (men ikke begge) er 1; resultatet er 0 hvis begge bitene er 0 eller begge biter er 1. Returnert verdi representerer bit-resultatene og vises i 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 111.

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

Z

zInterval Katalog >

zInterval σ,*Liste*[,*Frekv*[,*CNivå*]]

(Dataliste inndata)

zInterval σ,x̄,*n* [,*CNivå*]

(Summering statistikk inndata)

Beregner et *z* konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen (side 150).

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 202).

Utdata-variabel	Beskrivelse
stat.CLower, stat.Upper	Konfidensintervall for et ukjent populasjonsgjennomsnitt

Utdata-variabel	Beskrivelse
stat.$\bar{x}$	Utvalgets gjennomsnitt av datasekvensen fra normal tilfeldig fordeling
stat.ME	Feilmargin
stat.sx	Utvalgets standardavvik
stat.n	Lengde av datasekvensen med utvalgsgjennomsnitt
stat.σ	Kjent populasjons standardavvik for datasekvens *Liste*

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 150).

x er et ikke-negativt heltall.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 202).

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 150).

x1 og *x2* er ikke-negative heltall.

For informasjon om effekten av tomme elementer i en liste, se "Tomme (åpne) elementer" (side 202).

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 σ_1,σ_2,$\bar{x}$*1*,*n1*,$\bar{x}$,,*n2*[*CNivå*])

(Summering statistikk inndata)

Beregner et to-utvalgs *z* konfidensintervall. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 150).

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

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.σx1, stat.σx2	Utvalgets standardavvik for *Liste 1* og *Liste 2*
stat.n1, stat.n2	Antall utvalg i datasekvenser
stat.r1, stat.r2	Kjent populasjons standardavvik for datasekvens *Liste 1* og *Liste 2*

zTest Katalog >

zTest $\mu 0$,σ,*Liste*,[*Frekv*[,*Hypot*]]

(Dataliste inndata)

zTest $\mu 0$,σ,$\bar{x}$,*n*[,*Hypot*]

(Summering statistikk inndata)

Utfører en *z*-test med frekvens *frekvliste*. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 150).

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 202).

Utdata-variabel	Beskrivelse
stat.z	($\bar{x}$ − μ0) / (σ / sqrt(n))
stat.P-Verdi	Minste sannsynlighet som null-hypotesen kan forkastes ved
stat.$\bar{x}$	Utvalgsgjennomsnitt av datasekvensen i *Liste*
stat.sx	Utvalgets standardavvik av datasekvensen. Returneres kun for inndata *Data*.
stat.n	Utvalgenes størrelse

zTest_1Prop Katalog >

zTest_1Prop *p0*, *x*,*n*[,*Hypot*]

Beregner en en-proporsjons *z* -test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 150).

x er et ikke-negativt heltall.

Test H_0: $p = p0$ mot ett av følgende:

For H_a: $p > p0$, sett *Hypot*>0

For H_a: $p \neq p0$ *(standard)*, sett *Hypot*=0

For H_a: $p < p0$, sett *Hypot*<0

zTest_1Prop

Katalog >

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

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

Katalog >

zTest_2Prop *x1*,*n1*,*x2*,*n2*[,*Hypot*]

Beregner en to-proporsjons *z*-test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 150).

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: $p < p0$, sett *Hypot*<0

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

Utdata-variabel	Beskrivelse
stat.z	Standard normalverdi beregnet for differansen av andelene
stat.PVal	Minste signifikansnivå som null-hypotesen kan forkastes ved
stat.$\hat{p}$1	Beregnet andel av suksesser i utvalg 1
stat.$\hat{p}$2	Beregnet andel av suksesser i utvalg 2
stat.$\hat{p}$	Beregnet samlet andel av suksesser
stat.n1, stat.n2	Antall utvalg som er tatt i forsøk 1 og 2

zTest_2Samp σ_1,σ_2,*Liste1*,*Liste2*[,*Frekv1*[,*Frekv2*[,*Hypot*]]]

(Dataliste inndata)

zTest_2Samp σ_1,σ_2,$\bar{x}1$,*n1*,$\bar{x}2$,*n2*[,*Hypot*]

(Summering statistikk inndata)

Beregner en to-utvalgs *z*-test. En oversikt over resultatene lagres i *stat.results*-variabelen. (Se side 150).

Test H_0: $\mu1 = \mu2$, mot ett av følgende:

For H_a: $\mu1 < \mu2$, sett *Hypot*<0

For H_a: $\mu1 \neq \mu2$ (standard), sett *Hypot*=0

For H_a: $\mu1 > \mu2$, *Hypot*>0

For informasjon om effekten av tomme elementer i en liste, se “Tomme (åpne) elementer” (side 202).

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 *Liste*1 *og* Liste2
stat.sx1, stat.sx2	Utvalgets standardavvik til datasekvenser i *Liste* 1 *og* Liste2
stat.n1, stat.n2	Utvalgenes størrelse

Symboler

+ (addere) — [+] tast

Verdi1 + *Verdi2* ⇒*verdi*

Returnerer summen av de to argumentene.

56	56
56+4	60
60+4	64
64+4	68
68+4	72

Liste1 + *Liste2*⇒*liste*

Matrise1 + *Matrise2*⇒*matrise*

Returnerer en liste (eller matrise) som inneholder summene av tilsvarende elementer i *Liste1* og *Liste2* (eller *Matrise1* og *Matrise2*).

Dimensjonene i argumentene må være like.

$\left\{22,\pi,\frac{\pi}{2}\right\}\rightarrow l1$	$\{22,3.14159,1.5708\}$
$\left\{10,5,\frac{\pi}{2}\right\}\rightarrow l2$	$\{10,5,1.5708\}$
$l1+l2$	$\{32,8.14159,3.14159\}$

Verdi + *Liste1*⇒*liste*

Liste1 + *Verdi*⇒*liste*

Returnerer en liste med summene av *Verdi* og hvert element i *Liste1*.

$15+\{10,15,20\}$	$\{25,30,35\}$
$\{10,15,20\}+15$	$\{25,30,35\}$

Verdi + *Matrise1*⇒*matrise*

Matrise1 + *Verdi*⇒*matrise*

Returnerer en matrise med *Verdi* addert til hvert element på diagonalen til *Matrise1*. *Matrise1* må være kvadratisk.

Merk: Bruk **.+** (prikk pluss) for å addere et uttrykk til hvert element.

$20+\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$	$\begin{bmatrix}21 & 2\\3 & 24\end{bmatrix}$

−(subtrahere) — [−] tast

Verdi1 − *Verdi2*⇒*verdi*

Returnerer *Verdi1* minus *Verdi2*.

$6-2$	4
$\pi-\frac{\pi}{6}$	2.61799

Liste1 − *Liste2*⇒*liste*

Matrise1 − *Matrise2*⇒*matrise*

$\left\{22,\pi,\frac{\pi}{2}\right\}-\left\{10,5,\frac{\pi}{2}\right\}$	$\{12,{}^{-}1.85841,0.\}$
$[3\ \ 4]-[1\ \ 2]$	$[2\ \ 2]$

−(subtrahere) [−] tast

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⇒liste

Liste1 − Verdi⇒liste

Subtraherer hvert *Liste1* element fra *Verdi* eller subtraherer *Verdi* fra hvert *Liste1* element og returnerer en liste over resultatene.

$15-\{10,15,20\}$	$\{5,0,-5\}$
$\{10,15,20\}-15$	$\{-5,0,5\}$

Verdi − Matrise1⇒matrise

Matrise1 − Verdi⇒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.

$20-\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$	$\begin{bmatrix}19 & -2\\-3 & 16\end{bmatrix}$

Merk: Bruk **.−** (prikk minus) for å subtrahere et uttrykk fra hvert element.

•(multiplisere) [×] tast

Verdi1•Verdi2⇒verdi

Returnerer produktet av de to argumentene.

$2\cdot 3.45$	6.9

Liste1•Liste2⇒liste

Returnerer en liste som inneholder produktene av samsvarende elementer i *Liste1* og *Liste2*.

Dimensjonene i listene må være like.

$\{1.,2,3\}\cdot\{4,5,6\}$	$\{4,10,18\}$

Matrise1•Matrise2⇒matrise

Returnerer matriseproduktet av *Matrise1* og *Matrise2*.

Antallet kolonner i *Matrise1* må være likt antallet rader i *Matrise2*.

$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix}\cdot\begin{bmatrix}7 & 8\\7 & 8\\7 & 8\end{bmatrix}$	$\begin{bmatrix}42 & 48\\105 & 120\end{bmatrix}$

•(multiplisere) [×] tast

Verdi •Liste1⇒liste

Liste1 •Verdi⇒liste

Returnerer en liste med produktene av *Verdi* og hvert element i *Liste1*.

$\pi\cdot\{4,5,6\}$	$\{12.5664,15.708,18.8496\}$

Verdi •Matrise1⇒matrise

Matrise1•Verdi⇒matrise

Returnerer en matrise med produktene av *Verdi* og hvert element i *Matrise1*.

$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\cdot 0.01$	$\begin{bmatrix}0.01 & 0.02\\0.03 & 0.04\end{bmatrix}$
$6\cdot\text{identity}(3)$	$\begin{bmatrix}6 & 0 & 0\\0 & 6 & 0\\0 & 0 & 6\end{bmatrix}$

Merk: Bruk **.•**(prikk multipliser) for å multiplisere et uttrykk med hvert element.

/ (divider) [÷] tast

Verdi1 / Verdi2⇒verdi

Returnerer kvotienten av *Verdi1* dividert med *Verdi2*.

Merk: Se også **Brøk-sjablon**, side 1.

$\frac{2}{3.45}$	.57971

Liste1 / Liste2⇒liste

Returnerer en liste som inneholder kvotientene av Liste1 *dividert med Liste2*.

Dimensjonene i listene må være like.

$\frac{\{1.,2,3\}}{\{4,5,6\}}$	$\left\{0.25,\frac{2}{5},\frac{1}{2}\right\}$

Verdi / Liste1 ⇒ liste

Liste1 / Verdi ⇒ liste

Returnerer en liste med kvotientene av *Verdi* dividert med *Liste1* eller *Liste1* dividert med *Verdi*.

$\frac{6}{\{3,6,\sqrt{6}\}}$	$\{2,1,2.44949\}$
$\frac{\{7,9,2\}}{7\cdot 9\cdot 2}$	$\left\{\frac{1}{18},\frac{1}{14},\frac{1}{63}\right\}$

Verdi / Matrise1⇒ matrise

Matrise1 / Verdi ⇒ matrise

Returnerer en matrise med kvotientene av *Matrise1 /Verdi*.

Merk: Bruk **. /** (prikk divider) for å dividere et uttrykk med hvert element.

$\frac{[7\ \ 9\ \ 2]}{7\cdot 9\cdot 2}$	$\begin{bmatrix}\frac{1}{18} & \frac{1}{14} & \frac{1}{63}\end{bmatrix}$

^ (potens) — [^] tast

Verdi1 ^ *Verdi2* ⇒ *verdi*

Liste1 ^ *Liste2* ⇒ *liste*

Returnerer det første argument opphøyd i det andre argumentet.

Merk: Se også **Eksponent-sjablon**, side 1.

For en liste, returneres elementene i *Liste1* opphøyd i tilsvarende elementer i *Liste2*.

I reell grunnmengde bruker brøkpotens som har forkortet eksponent med oddetall i nevner en rell forgreining i motsetning til hovedforgreining for kompleks modus.

4^2 → 16

$\{2,4,6\}^{\{1,2,3\}}$ → $\{2,16,216\}$

Verdi ^ *Liste1* ⇒ *liste*

Returnerer *Verdi* opphøyd i elementene i *Liste1*.

$\pi^{\{1,2,-3\}}$ → $\{3.14159,9.8696,0.032252\}$

Liste1 ^ *Verdi* ⇒ *liste*

Returnerer elementene i *Liste1* opphøyd i *Verdi*.

$\{1,2,3,4\}^{-2}$ → $\left\{1,\frac{1}{4},\frac{1}{9},\frac{1}{16}\right\}$

kvadratMatrise1 ^ *heltall* ⇒ *matrise*

Returnerer *kvadratMatrise1* opphøyd i *heltall* -potens.

kvadratMatrise1 må være en kvadratmatrise.

Hvis *heltall* = ⁻1, beregnes invers matrise.

Hvis *heltall* < ⁻1, beregnes invers matrise opphøyd i en korrekt positiv potens.

$\begin{bmatrix}1&2\\3&4\end{bmatrix}^2$ → $\begin{bmatrix}7&10\\15&22\end{bmatrix}$

$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{-1}$ → $\begin{bmatrix}-2&1\\\frac{3}{2}&\frac{-1}{2}\end{bmatrix}$

$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{-2}$ → $\begin{bmatrix}\frac{11}{2}&\frac{-5}{2}\\\frac{-15}{4}&\frac{7}{4}\end{bmatrix}$

x^2 (kvadrat) — [x²] tast

Verdi1 ² ⇒ *verdi*

Returnerer kvadratet av argumentet.

*Liste1*² ⇒ *liste*

Returnerer en liste med kvadrater av elementene i *Liste1*.

kvadratMatrise1 ² ⇒ *matrise*

4^2 → 16

$\{2,4,6\}^2$ → $\{4,16,36\}$

$\begin{bmatrix}2&4&6\\3&5&7\\4&6&8\end{bmatrix}^2$ → $\begin{bmatrix}40&64&88\\49&79&109\\58&94&130\end{bmatrix}$

$\begin{bmatrix}2&4&6\\3&5&7\\4&6&8\end{bmatrix}.\hat{}\,2$ → $\begin{bmatrix}4&16&36\\9&25&49\\16&36&64\end{bmatrix}$

x^2 (kvadrat) [x²] tast

Returnerer matrisens kvadrat av *kvadratMatrise1*. Dette er ikke det samme som å beregne kvadratet av hvert element. Bruk .^2 for å beregne kvadratet av hvert element.

.+ (prikk adder) [.][+] taster

Matrise1 **.+** *Matrise2* ⇒ *matrise*

Verdi **.+** *Matrise1* ⇒ *matrise*

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*.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} .+ \begin{bmatrix}10 & 30\\20 & 40\end{bmatrix} \qquad \begin{bmatrix}11 & 32\\23 & 44\end{bmatrix}$$

$$5 .+ \begin{bmatrix}10 & 30\\20 & 40\end{bmatrix} \qquad \begin{bmatrix}15 & 35\\25 & 45\end{bmatrix}$$

.– (prikk subt.) [.][–] taster

Matrise1 **.–** *Matrise2* ⇒ *matrise*

Verdi **.–** *Matrise1* ⇒ *matrise*

Matrise1 **.–** *Matrise2* returnerer en matrise som er differansen mellom hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

Verdi **.–** *Matrise1* returnerer en matrise som ere differansen av *Verdi* og hvert element i *Matrise1*.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} .- \begin{bmatrix}10 & 20\\30 & 40\end{bmatrix} \qquad \begin{bmatrix}-9 & -18\\-27 & -36\end{bmatrix}$$

$$5 .- \begin{bmatrix}10 & 20\\30 & 40\end{bmatrix} \qquad \begin{bmatrix}-5 & -15\\-25 & -35\end{bmatrix}$$

.• (prikk mult.) [.][×] taster

Matrise1 **.•** *Matrise2* ⇒ *matrise*

Verdi **.•** *Matrise1*⇒ *matrise*

Matrise1 **.•** *Matrise2* returnerer en matrise som er produktet av hvert par av samsvarende elementer i *Matrise1* og *Matrise2*.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} .\cdot \begin{bmatrix}10 & 20\\30 & 40\end{bmatrix} \qquad \begin{bmatrix}10 & 40\\90 & 160\end{bmatrix}$$

$$5 .\cdot \begin{bmatrix}10 & 20\\30 & 40\end{bmatrix} \qquad \begin{bmatrix}50 & 100\\150 & 200\end{bmatrix}$$

.• (prikk mult.) [.][×] taster

Verdi **.•** *Matrise1* returnerer en matrise med produktene av *Verdi* og hvert element i *Matrise1*.

. / (prikk divider) [.][÷] taster

Matrise1 **./** *Matrise2* ⇒ *matrise*

Verdi **./** *Matrise1*⇒ *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*.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} ./ \left(\begin{bmatrix}10 & 20\\30 & 40\end{bmatrix}\right) \qquad \begin{bmatrix}\frac{1}{10} & \frac{1}{10}\\\frac{1}{10} & \frac{1}{10}\end{bmatrix}$$

$$5 ./ \left(\begin{bmatrix}10 & 20\\30 & 40\end{bmatrix}\right) \qquad \begin{bmatrix}\frac{1}{2} & \frac{1}{4}\\\frac{1}{6} & \frac{1}{8}\end{bmatrix}$$

.^ (prikk potens) [.][^] taster

Matrise1 **.^** *Matrise2*⇒ *matrise*

Verdi **. ^** *Matrise1*⇒ *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*.

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} .^ \begin{bmatrix}0 & 2\\3 & -1\end{bmatrix} \qquad \begin{bmatrix}1 & 4\\27 & \frac{1}{4}\end{bmatrix}$$

$$5 .^ \begin{bmatrix}0 & 2\\3 & -1\end{bmatrix} \qquad \begin{bmatrix}1 & 25\\125 & \frac{1}{5}\end{bmatrix}$$

⁻(negere) [(−)] tast

⁻*Verdi1* ⇒ *verdi*

⁻*Liste1* ⇒ *liste*

⁻*Matrise1* ⇒ *matrise*

Returnerer argumentets negasjon.

For en liste eller matrise returneres alle elementene negert.

-2.43	-2.43
-{-1,0.4,1.2E19}	{1.,-0.4,-1.2E19}

`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 **%** ⇒ *verdi*

Liste1 **%** ⇒ *liste*

Matrise1 **%** ⇒ *matrise*

Returnerer $\frac{argument}{100}$

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*⇒*Boolsk uttrykk*

Liste1 **=** *Liste2*⇒ *Boolsk liste*

Matrise1 **=** *Matrise2*⇒ *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.

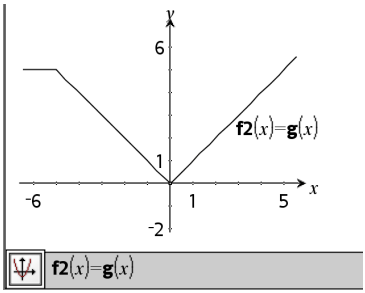
Eksempel på funksjon som bruker matematiske testsymboler: =, ≠, <, ≤, >, ≥

```
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)

[ctrl] [=] taster

Uttr1 ≠ *Uttr2* ⇒ *Boolsk uttrykk*

Liste1 ≠ *Liste2* ⇒ *Boolsk liste*

Matrise1 ≠ *Matrise2* ⇒ *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 **/=**

Se "=" (er lik) eksempel.

< (mindre enn)

[ctrl] [=] taster

Uttr1 < *Uttr2* ⇒ *Boolsk uttrykk*

Liste1 < *Liste2* ⇒ *Boolsk liste*

Matrise1 < *Matrise2* ⇒ *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*.

Se "=" (er lik) eksempel.

< (mindre enn)

ctrl = taster

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

≤ (mindre enn eller lik)

ctrl = taster

$Uttr1 \leq Uttr2 \Rightarrow Boolsk\ uttrykk$

$Liste1 \leq Liste2 \Rightarrow Boolsk\ liste$

$Matrise1 \leq Matrise2 \Rightarrow Boolsk\ matrise$

Returnerer sann hvis $Uttr1$ er bestemt å være mindre enn eller lik $Uttr2$.

Returnerer usann hvis $Uttr1$ er bestemt å være større enn $Uttr2$.

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive **<=**

Se "=" (er lik) eksempel.

> (større enn)

ctrl = taster

$Uttr1 > Uttr2 \Rightarrow Boolsk\ uttrykk$

$Liste1 > Liste2 \Rightarrow Boolsk\ liste$

$Matrise1 > Matrise2 \Rightarrow Boolsk\ matrise$

Returnerer sann hvis $Uttr1$ er bestemt å være større enn $Uttr2$.

Returnerer usann hvis $Uttr1$ er bestemt å være mindre enn eller lik $Uttr2$.

Alt annet returnerer en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Se "=" (er lik) eksempel.

≥ (større enn eller lik med)

ctrl = taster

Uttr1 ≥ *Uttr2* ⇒ *Boolsk uttrykk*

Liste1 ≥ *Liste2* ⇒ *Boolsk liste*

Matrise1 ≥ *Matrise2* ⇒ *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 **>=**

Se "=" (er lik) eksempel.

⇒ (logisk implikasjon)

ctrl = -taster

BoolskUttr1 ⇒ *BoolskUttr2* returnerer *Boolsk uttrykk*

BoolskListe1 ⇒ *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrise1 ⇒ *BoolskMatrise2* returnerer *Boolsk matrise*

Heltall1 ⇒ *Heltall2* returnerer *Heltall*

5>3 or 3>5	true
5>3 ⇒ 3>5	false
3 or 4	7
3 ⇒ 4	-4
{1,2,3} or {3,2,1}	{3,2,3}
{1,2,3} ⇒ {3,2,1}	{-1,-1,-3}

Behandler uttrykket **not <argument1> or <argument2>** og returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive **=>**

⇔ (logisk dobbel implikasjon, XNOR)

ctrl = -taster

BoolskUttr1 ⇔ *BoolskUttr2* returnerer *Boolsk uttrykk*

BoolskListe1 ⇔ *BoolskListe2* returnerer *Boolsk liste*

BoolskMatrise1 ⇔ *BoolskMatrise2* returnerer *Boolsk matrise*

Heltall1 ⇔ *Heltall2* returnerer *Heltall*

5>3 xor 3>5	true
5>3 ⇔ 3>5	false
3 xor 4	7
3 ⇔ 4	-8
{1,2,3} xor {3,2,1}	{2,0,2}
{1,2,3} ⇔ {3,2,1}	{-3,-1,-3}

Returnerer negasjon av en **XOR** Boolsk handling på de to argumentene. Returnerer sann, usann eller en forenklet form av ligningen.

For lister og matriser, returneres sammenlikninger element for element.

Merk: Du kan sette inn denne operatoren fra tastaturet ved å skrive **<=>**

! (fakultet)

?!▸ tast

Verdi1! ⇒ *verdi*

*Liste1***!** ⇒ *liste*

*Matrise1***!** ⇒ *matrise*

5!	120
{{5,4,3}}!	{120,24,6}
$\left(\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}\right)!$	$\begin{bmatrix}1 & 2\\6 & 24\end{bmatrix}$

Returnerer argumentets fakultet.

For en liste eller matrise, returneres en liste eller matrise av elementenes fakulteter.

& (legg til)

ctrl 📖 taster

Streng1 **&** *Streng2* ⇒ *streng*

"Hello "&"Nick"	"Hello Nick"

Returnerer en tekststreng som er *Streng2* lagt til *Streng1*.

d(*Uttr1*, *Var*[, *Orden*]) | *Var=Verdi⇒verdi*

d(*Uttr1*, *Var*[, *Orden*])⇒*verdi*

d(*Liste1*, *Var*[, *Orden*])⇒*liste*

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.

Studer eksemplet til høyre. Den første deriverte av x·(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()**.

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

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

$x:=3:\frac{d}{dx}(\{x^2,x^3,x^4\})$ $\{6,27,108\}$

$\frac{d}{dx}\left(x\cdot(x^2+x)^{\frac{1}{3}}\right)|x=0$ undef

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

∫() (integral) Katalog >

∫(*Uttr1*, *Var*, *Nedre*, *Øvre*) ⇒ *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 avsamme metode som nInt().

$$\int_0^1 x^2\,dx \qquad 0.333333$$

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

Merk: Se også **nInt()**, side 105 og **Bestemt integral-sjablon**, side 6.

√() (kvadratrot) ctrl x² taster

√(*Verdi1*)⇒*verdi*

√(*Liste1*)⇒*liste*

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

$$\sqrt{\{9,2,4\}} \qquad \{3,1.41421,2\}$$

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 `sqrt(...)`

Merk: Se også **Kvadratrot-sjablon**, side 1.

Π() (prodSeq) Katalog >

Π(*Uttr1*, *Var*, *Nedre*, *Øvre*)⇒*uttrykk*

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

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

Finner *Uttrykk1* for hver verdi av *Var* fra *Nedre* til *Øvre*, og returnerer produktet av resultatene.

$$\prod_{n=1}^{5}\left(\left\{\frac{1}{n},n,2\right\}\right) \qquad \left\{\frac{1}{120},120,32\right\}$$

Merk: Se også **Produkt-sjablon** (Π), side 5.

Π(*Uttr1*, *Var*, *Nedre*, *Nedre*−1)⇒1

$$\prod_{k=4}^{3}(k) \qquad 1$$

Π(*Uttr1*, *Var*, *Nedre*, *Øvre*) ⇒**1/**Π(*Uttr1*, *Var*, *Øvre*+*1*, *Nedre*−*1*) hvis *Øvre* < *Nedre* *1*

Π() (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.

$$\prod_{k=4}^{1}\left(\frac{1}{k}\right) \qquad 6$$

$$\prod_{k=4}^{1}\left(\frac{1}{k}\right)\cdot\prod_{k=2}^{4}\left(\frac{1}{k}\right) \qquad \frac{1}{4}$$

Σ() (sumSeq) Katalog >

Σ(*Uttr1***,** *Var***,** *Nedre***,** *Øvre***)⇒***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.

$$\sum_{n=1}^{5}\left(\frac{1}{n}\right) \qquad \frac{137}{60}$$

Σ(*Uttr1***,** *Var***,** *Nedre***,** *Nedre−1***)⇒**0

Σ(*Uttr1***,** *Var***,** *Nedre***,** *Øvre***)⇒-Σ(***Uttr1***,** *Var***,** *Øvre+1***,** *Nedre***−1)** hvis *Øvre* < *Nedre−1*

$$\sum_{k=4}^{3}(k) \qquad 0$$

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.

$$\sum_{k=4}^{1}(k) \qquad -5$$

$$\sum_{k=4}^{1}(k)+\sum_{k=2}^{4}(k) \qquad 4$$

ΣInt() Katalog >

ΣInt(*NPmt1***,** *NPmt2***,** *N***,** *I***,** *PV***,[***Pmt***],** [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*avrundVerdi*]**)** ⇒*verdi*

ΣInt(*NPmt1***,***NPmt2***,***amortTabell***)⇒***verdi*

ΣInt(1,3,12,4.75,20000,,12,12) −213.48

ΣInt() Katalog >

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 166.

- 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() Katalog >

ΣPrn(*NPmt1***,** *NPmt2***,** *N***,** *I***,** *PV***,** [*Pmt*]**,** [*FV*]**,** [*PpY*]**,** [*CpY*]**,** [*PmtAt*]**,** [*avrundVerdi*]**)⇒***verdi*

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.

ΣPrn(1,3,12,4.75,20000,,12,12) -4916.28

ΣPrn() Katalog >

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen med TVM-argumenter, side 166.

- 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.

ΣPrn(*NPmt1,NPmt2,amortTabell***)** beregner summen av hovedbetalinger basert på amortiseringstabell *amortTabell*. Argumentet *amortTabell* må være en matrise i den form som er beskrevet under **amortTbl()**, side 7.

Merk: Se også ΣInt(), over, og **Bal()**, side 15.

tbl:=amortTbl(12,12,4.75,20000,,12,12)

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

ΣPrn(1,3,*tbl*) -4916.28

(Indir.ref) ctrl taster

*varNavnStreng*

Refererer til variabelen med navnet *varNavnStreng*. På denne måten kan du bruke strenger for å opprette variabelnavn "innenfra" en funksjon.

xyz:=12 12

#("x"&"y"&"z") 12

Oppretter eller refererer til variabelen xyz.

10→*r* 10

"r"→*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*

Legger inn et tall i vitenskapelig fremstilling. Tallet blir tolket som en *mantissa* × $10^{\text{eksponent}}$.

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.

23000.	23000.
2300000000.+4.1E15	4.1E15
$3\cdot 10^4$	30000

g (gradian) π▸ tast

*Uttr1*g ⇒*Uttrykk*

*Uttr1*g ⇒*Uttrykk*

*Liste1*g ⇒*liste*

*Matrise1*g ⇒*matrise*

Denne funksjonen gir deg en mulighet til å spesifisere en vinkel i gradianer mens du er i grader- eller gradian-modus.

I radian-vinkelmodus, multipliseres *Uttr1* med π/200.

I grader-vinkelmodus, multipliseres *Uttr1* med g/100.

I gradian modus, returneres *Uttr1* uendret.

Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive `@g`.

I grader, gradian eller radian modus:

$\cos(50^g)$	0.707107
$\cos(\{0,100^g,200^g\})$	{1,0.,-1.}

r (radian) π▸ tast

*Verdi1*r ⇒*verdi*

*Liste1*r ⇒*liste*

*Matrise1*r ⇒*matrise*

I grader, gradian eller radian modus:

$\cos\left(\frac{\pi}{4^r}\right)$	0.707107
$\cos\left(\left\{0^r,\left(\frac{\pi}{12}\right)^r,-(\pi)^r\right\}\right)$	{1,0.965926,-1.}

ʳ (radian) π▸ 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/ π.

I radian-vinkelmodus, returneres argumentet uendret.

I gradian modus, multipliseres argumentet med 200/ π.

Tips: Bruk ʳhvis du vil tvinge radianer inn en funksjonsdefinisjon uavhengig av hvilken modus som er i bruk når du bruker funksjonen.

Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive `@r`.

° (grader) π▸ tast

Verdi1°⇒verdi

Liste1°⇒liste

Matrise1°⇒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 π/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:

$\cos(45°)$ 0.707107

I radian modus:

$\cos\left(\left\{0,\frac{\pi}{4},90°,30.12°\right\}\right)$

$\{1.,0.707107,0.,0.864976\}$

°, ', '' (grader/minutter/sekunder) ctrl 📖 taster

gg°mm'ss.ss''⇒Uttrykk

gg Et positivt eller negativt tall

I Grader-vinkelmodus:

°, ', " (grader/minutter/sekunder)

ctrl 📖 taster

mm Et ikke-negativt tall

ss.ss Et ikke-negativt tall

Returnerer *gg*+(*mm*/60)+(*ss.ss*/3600).

Dette grunntall -60-formatet lar deg:

- Sette inn en vinkel i grader/minutter/sekunder uten hensyn til aktuell vinkelmodus.
- Sette inn tid, som timer/minutter/sekunder.

Merk: Sett to apostrofer ('') etter ss.ss, ikke et anførselstegn ("").

25°13'17.5"	25.2215
25°30'	$\frac{51}{2}$

∠ (vinkel)

ctrl 📖 taster

[*Radius*,∠θ_*Vinkel*]⇒*vektor*

(polar inndata)

[*Radius*,∠θ_*Vinkel*, *Z_Koordinat*] ⇒*vektor*

(sylindrisk inndata)

[*Radius*,∠θ_*Vinkel*,∠θ_*Vinkel*]⇒*vektor*

(sfærisk inndata)

Returnerer koordinater som en vektor, avhengig av vektorformatets modusinnstilling: rektangulær, sylindrisk eller sfærisk.

Merk: Du kan sette inn dette symbolet fra datamaskintastaturet ved å skrive **@<**.

I radian modus og vektorformat innstilt på: rektangulær

$[5 \ \angle 60° \ \angle 45°]$

$[1.76777 \quad 3.06186 \quad 3.53553]$

sylindrisk

$[5 \ \angle 60° \ \angle 45°]$

$[3.53553 \quad \angle 1.0472 \quad 3.53553]$

sfærisk

$[5 \ \angle 60° \ \angle 45°]$

$[5. \quad \angle 1.0472 \quad \angle 0.785398]$

(*Størrelse* ∠ *Vinkel*)⇒*kompleksVerdi*

(polar inndata)

Setter inn en kompleks verdi i (r∠θ) polar form. *Vinkelen* tolkes avhengig av aktuell vinkelmodus-innstilling.

I radian-vinkelmodus og rektangulært, komplekst format:

$5+3\cdot i-\left(10 \angle \frac{\pi}{4}\right)$ $\quad -2.07107-4.07107\cdot i$

10^()

Katalog >

10^ (*Verdi1***)⇒***verdi*

10^ (*Liste1***)⇒***liste*

Returnerer 10 opphøyd i argumentets potens.

For en liste, returneres 10 opphøyd i elementenes potens i *Liste1*.

$10^{1.5}$ 31.6228

10^(*kvadratMatrise***)⇒***kvadratMatrise*

Returnerer 10 opphøyd i potensen av *kvadratMatrise1*. Dette er ikke det samme som å beregne 10 opphøyd i potens av hvert element. For mer informasjon om beregningsmetode, se under **cos()**.

kvadratMatrise1 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 >

Verdi1 **^⁻¹⇒***verdi*

Liste1 **^⁻¹⇒***liste*

Returnerer den inverse verdien av et argument.

For en liste, returneres den inverse verdien av elementene i *Liste1*.

$(3.1)^{-1}$ 0.322581

kvadratMatrise1 **^⁻¹⇒***kvadratMatrise*

Returnerer den inverse verdien av *kvadratMatrise1*.

kvadratMatrise1 må være en ikke-singulær kvadratisk matrise.

$\begin{bmatrix}1&2\\3&4\end{bmatrix}^{-1}$

$$\begin{bmatrix}-2 & 1\\ \frac{3}{2} & \frac{-1}{2}\end{bmatrix}$$

| (begrensningsoperator)

ctrl -taster

Uttr | *BoolskUttr1* [**and** *BoolskUttr2*]...

Uttr | *BoolskUttr1* [**or** *BoolskUttr2*]...

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 sammenføyes av logiske "**and**" eller "**or**" operatorer.

$x+1\|x=3$	4
$x+55\|x=\sin(55)$	54.0002

Med begrensningsoperatoren har du tre utgangstyper av funksjonalitet:

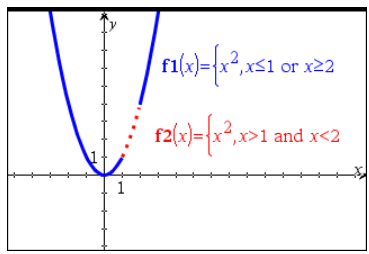
- Erstatninger
- Intervallbegrensninger
- Eksklusjoner

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**

Verdi → *Var*

Liste → *Var*

Matrise → *Var*

Uttr → *Funksjon*(*Param1*,...)

Liste → *Funksjon*(*Param1*,...)

Matrise → *Funksjon*(*Param1*,...)

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**.

$\frac{\pi}{4} \rightarrow myvar$	0.785398
$2\cdot\cos(x) \rightarrow y1(x)$	*Done*
$\{1,2,3,4\} \rightarrow lst5$	$\{1,2,3,4\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow matg$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
"Hello" → *str1*	"Hello"

:= (tildele)

ctrl **taster**

Var := *Verdi*

Var := *Liste*

Var := *Matrise*

Funksjon(*Param1*,...) := *Uttr*

Funksjon(*Param1*,...) := *Liste*

Funksjon(*Param1*,...) := *Matrise*

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*.

$myvar := \frac{\pi}{4}$	.785398
$y1(x) := 2\cdot\cos(x)$	*Done*
$lst5 := \{1,2,3,4\}$	$\{1,2,3,4\}$
$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
str1:="Hello"	"Hello"

© (kommentar)

ctrl 📖 taster

© [*tekst*]

© fremstiller *tekst* som en kommentarlinje som lar deg kommentere funksjoner og programmer som du oppretter.

© kan plasseres ved begynnelsen eller hvor som helst på linjen. Alt som er til høyre for ©, til slutten av linjen, er kommentaren.

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

```
Define g(n)=Func
        © Declare variables
        Local i,result
        result:=0
        For i,1,n,1 ©Loop n times
        result:=result+i^2
        EndFor
        Return result
        EndFunc
                                Done
g(3)                            14
```

0b, 0h

0 B taster, 0 H taster

0b *binærTall*

0h *heksadesimalTall*

Markerer hhv. et binært eller heksadesimalt tall. For å sette inn et binært eller heksadesimalt tall må du sette inn prefikset 0b eller 0h uavhengig av grunninnstillingsmodus. Uten prefiks blir et tall behandlet som et desimaltall (grunntall 10).

Resultatene vises i forhold til grunntall-modusen.

I desimalt grunntall-modus:

```
0b10+0hF+10                     27
```

I binær grunntall-modus:

```
0b10+0hF+10                     0b11011
```

I heksades grunntall-modus:

```
0b10+0hF+10                     0h1B
```

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 77.

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 [ctrl] [␣].

Beregninger som involverer åpne elementer

De fleste beregninger som involverer et åpent (tomt) innlegg, vil produsere et åpent (tomt) resultat. Se spesialtilfeller nedenfor.

$\lvert _ \rvert$	_
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

sum({2,_,3,5,6.6})	16.6
median({1,2,_,_,_,3})	2
cumulativeSum({1,2,_,4,5})	{1,3,_,7,12}
cumulativeSum([1 2; 3 _; 5 6])	[1 2; 4 _; 9 8]

SortA og **SortD** flytter alle åpne (tomme) elementer innenfor det første utsagnet til bunnen.

{5,4,3,_,1}→*list1*	{5,4,3,_,1}
{5,4,3,2,1}→*list2*	{5,4,3,2,1}
SortA *list1*,*list2*	*Done*
list1	{1,3,4,5,_}
list2	{1,3,4,5,2}

Listeutsagn som inneholder åpne (tomme) elementer

{1,2,3,_,5} → *list1*	{1,2,3,_,5}
{1,2,3,4,5} → *list2*	{1,2,3,4,5}
SortD *list1*,*list2*	*Done*
list1	{5,3,2,1,_}
list2	{5,3,2,1,4}

I regresjoner introduserer en åpning i en X- eller Y-liste en åpning for det tilsvarende elementet i en rest.

l1:={1,2,3,4,5}: *l2*:={2,_,3,5,6.6}	{2,_,3,5,6.6}
LinRegMx *l1*,*l2*	*Done*
stat.Resid	{0.434286,_,-0.862857,-0.011429,0.44}
stat.XReg	{1.,_,3.,4.,5.}
stat.YReg	{2.,_,3.,5.,6.6}
stat.FreqReg	{1.,_,1.,1.,1.}

En utelatt kategori i en regresjon introduserer en åpning (tomt element) for det tilsvarende elementet i en rest.

l1:={1,3,4,5}: *l2*:={2,3,5,6.6}	{2,3,5,6.6}
cat:={"M","M","F","F"}: *incl*:={"F"	{"F"
LinRegMx *l1*,*l2*,1,*cat*,*incl*	*Done*
stat.Resid	{_,_,0.,0.}
stat.XReg	{_,_,4.,5.}
stat.YReg	{_,_,5.,6.6}
stat.FreqReg	{_,_,1.,1.}

En frekvens på 0 i en regresjon introduserer en åpning (tomt element) for det tilsvarende elementet i en rest.

l1:={1,3,4,5}: *l2*:={2,3,5,6.6}	{2,3,5,6.6}
LinRegMx *l1*,*l2*,{1,0,1,1}	*Done*
stat.Resid	{0.069231,_,-0.276923,0.207692}
stat.XReg	{1.,_,4.,5.}
stat.YReg	{2.,_,5.,6.6}
stat.FreqReg	{1.,_,1.,1.}

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(...)**, ...
ΔList()	**deltaList(...)**

Fra tastaturet på datamaskinen

Hvis du skal legge inn dette:	Skriv dette:
i (imaginær konstant)	**@i**
e (naturlig log-grunntall e)	**@e**
E (vitenskapelig notasjon)	**@E**
^{T}(transponert)	**@t**
^{r} (radianer)	**@r**

Hvis du skal legge inn dette:	Skriv dette:
° (grader)	`@d`
^{g} (gradianer)	`@g`
∠ (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 (ʳ), senket skrift ([ ]), transponert (ᵀ)
5	Eksponensiering, potens-operator (^)
6	Negasjon (L)
7	Sett sammen streng (&)
8	Multiplikasjon (•), divisjon (/)
9	Addisjon (+), subtraksjon (-)
10	Likhetsrelasjoner: lik (=), ulik (≠ eller /=), mindre enn (<), mindre enn eller lik (≤ eller <=), større enn (>), større enn eller lik (≥ eller >=)
11	Logisk **not**
12	Logisk **and**
13	Logiske **or**
14	**enten...eller**, **verken ...eller**, **ikke både ...og**
15	Logisk implikasjon (⇒)
16	Logisk dobbel implikasjon, XNOR (⇔)
17	Begrensningsoperator (“|”)
18	Lagre (→)

Parenteser, hakeparenteser, buede parenteser

Først behandles alle beregninger som står i parentes, hakeparentes eller buet parentes. I uttrykket 4(1+2) behandler EOS™ -programvaren for eksempel først den delen av uttrykket som står i parenteser, 1+2, og multipliserer deretter resultatet, 3, med 4.

Antallet åpne- og lukkeparenteser, åpne- og lukke-hakeparenteser og buede åpne- og lukkeparenteser må være det samme innenfor ett uttrykk eller én ligning. Hvis ikke, vises en feilmelding, som angir det manglende elementet. For eksempel vil (1+2)/(3+4 vise feilmeldingen "Mangler)."

Merk: Siden TI-Nspire™ -programvaren gjør at du kan definere dine egne funksjoner, blir et variabelnavn fulgt av et uttrykk i parentes betraktet som en "oppkalling av funksjon" istedenfor halvveis skjult multiplikasjon. For eksempel i a(b+c) blir funksjonen a beregnet for verdien b+c (av den variable). For å multiplisere uttrykket b+c med variabelen a, må du bruke eksplisitt multiplikasjon: a*(b+c).

Omregning

Omregnings-operatoren (#) omregner en streng til et variabel- eller funksjonsnavn. For eksempel oppretter #("x"&"y"&"z") variabelnavnet xyz. Omregning lar deg også opprette og modifisere en variabel mens du er inne i et program. Hvis for eksempel 10→r og "r"→s1, så er #s1=10.

Postoperatorer

En postoperator er en operator som kommer direkte etter et argument, som f.eks. 5!, 25%, eller 60°15' 45". Et argument som er fulgt av en postoperator blir behandlet ved fjerde prioritetsnivå. I uttrykket 4^3! blir blir for eksempel 3! behandlet først. Resultatet, 6, blir så eksponenten av 4 for å oppnå 4096.

Eksponensiering

Eksponensiering (^) og element-for-element-eksponensiering (.^) blir behandlet fra høyre til venstre. Uttrykket 2^3^2 blir for eksempel behandlet som det samme som 2^(3^2) for å produsere 512. Dette er forskjellig fra (2^3)^2, som er 64.

Negasjon

For å legge inn et negativt tall, trykk på [(−)] fulgt av tallet. Postoperasjoner og eksponensiering utføres før negasjon. Resultatet av $^{-}x^2$ er for eksempel et negativt tall, og $^{-}9^2 = {}^{-}81$. Bruk parenteser for å opphøye et negativt tall i annen potens, som f.eks. $(^{-}9)^2$ for å produsere 81.

Begrensning ("|")

Argumentet som følger etter ("|")-operator gir et sett av begrensninger som påvirker hvordan argumentet som står foran operatoren blir behandlet.

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 162.

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 esc eller on 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->a, der a er en udefinert variabel, forårsake denne feilen.
200	Ugyldig begrensningsuttrykk For eksempel vil løs(3x^2-4=0,x) | 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 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 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	Reservert navn eller systemvariabel
900	Argumentfeil Median-median-modell kunne ikke brukes på datasettet.
910	Syntaksfeil
920	Fant ikke teksten
930	For få argumenter Funksjonen eller kommandoen mangler et eller flere argumenter.
940	For mange argumenter Uttrykket eller ligningen inneholder for mange argumenter og kan ikke behandles.
950	For mange indekser
955	For mange udefinerte variabler
960	Variabelen er ikke definert Variabelen er ikke tildelt noen verdi. Bruk en av følgende kommandoer: • sto → • := • **Define** for å tildele variablene verdi.
965	Ulisensiert OS
970	Variabel er i bruk, så referanser eller endringer er ikke tillatt
980	Variabel er beskyttet
990	Ugyldig variabelnavn Pass på at navnet ikke overgår tillatt lengde
1000	Grunnmengde for vindusvariabel
1010	Zoom
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 √(-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 $xxx\backslash yyy$, 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 opplister hver numeriske varselkode og dens assosierte melding.

Se **warnCodes()** på side 171 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 udef^0 erstattet av 1
10014	udef^0 erstattet av 1
10015	1^∞ eller 1^udef erstattet av 1
10016	1^udef erstattet av 1
10017	Overflyt erstattet av ∞ eller ⁻∞
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.

Stikkordregister

'

', fremstilling minutter 196

–

–, subtrahere[*] 179

!

!, fakultet 189

"

", fremstilling sekunder 196

#

#, Indir.ref 194
#, omregnings-operator 207

%

%, prosent 185

&

&, legg til 189

*

*, multiplisere 180

.

.-, prikk subtraksjon 183
.*, prikk multiplikasjon 183
./ , prikk divisjon 184
.^, prikk potens 184
.+, prikk addisjon 183

:

:=, tildele 200

^

^, potens 182

|

|, begrensningsoperator 199

+

+, addere 179

/

/, dividere[*] 181

≠

≠ , ulik[*] 186

=

=, er lik 185

>

>, større enn 187

∏

∏, produkt, sjablon for 5
∏, produkt[*] 191

∑

∑( ), sum[*] 192
∑Int( ) 192
∑Prn( ) 193

√

√, kvadratrot[*] 191

∫

∫, integral[*] 191

≤

≤ , mindre enn eller lik 187

≥

≥ , større enn eller lik med 188

▸

▸Cylind, vise som sylindrisk vektor [Sylind] 35
▸Polar, vise som polar vektor[Polar] 115
▸, omregner til gradian vinkel[Grad] . 69
▸approxFraction( ) 13
▸DD, vises som desimalvinkel[DD] .. 36
▸Desimal, vise resultat som desimal [Decimal] 37
▸DMS, vise som grader/minutter/sekunder [DMS (GMS)] 43
▸Grunntall10, vise som desimalt heltall[Grunntall10] 18
▸Grunntall16, vise som heksadesimal [Grunntall16] 18
▸Grunntall2, vise som binær [Grunntall2] 16
▸Rad, omregne til radian vinkel 122
▸Rect, vise som rektangulær vektor . 125
▸Sphere, vise som sfærisk vektor [Sfære (kule)] 149

→

→, lagre 200

⇒

⇒ , logisk implikasjon[*] 188, 204

⇔

⇔ , logisk dobbel implikasjon[*] 189

©

©, kommentar 201

°

°, grader fremstilling[*] 196
°, grader/minutter/sekunder[*] 196

0

0b, binær indikator 201
0h, heksadesimal indikator 201

1

10^( ), tier-potens 198

2

2-delers stykkevis funksjon sjablon for 2
2-utvalg F test 58

A

abs( ), absoluttverdi 7
Absoluttverdi sjablon for 3-4
addere, + ... 179
amortiseringstabell, amortTbl( ) 7, 15
amortTbl( ), amortiseringstabell 7, 15
and, Boolsk operator 8
andrederivert sjablon for 6
angle( ), vinkel 9
ANOVA, enveis varians-analyse 9
ANOVA2-veis, toveis varians-analyse 10
Ans, siste svar 12
antall betingede elementer i en liste , tellIf( ) 31
antall elementer i en liste, antall( ) ... 30
antall( ), antall elementer i en liste ... 30
approx( ), tilnærmet 12
approxRational( ) 13
arccos() 13
arccosh() 13
arccot() 13
arccoth() 13
arccsc() 13
arccsch() 14
arcsec() 14
arcsech() 14
arcsin() 14
arcsinh() 14
arctan() 14
arctanh() 14
argumenter i TVM-funksjoner 166
augment( ), utvid/sett sammen 14
avgRC( ), gjennomsnittlig 15

endringshastighet .....
avrund, round( ) 134
avslutt
funksjon, EndFunc 59
avslutt, Exit 49

B

begrensningsoperator "|" 199
begrensningsoperator, rekkefølge av beregning 206
behandle polynom, polyEval( ) 115
behandling, rekkefølge av 206
bestemt integral
sjablon for 6
bibliotek
lage snarveier til objekter 78
BiblOff 38
BiblPriv 38
binomCdf( ) 19, 74-75
binomPdf( ) 19
binær
indikator, 0b 201
vise, ▸Grunntall2 16
Boolske operatorer
⇒ 188, 204
⇔ 189
and 8
eller 111
enten ... eller 172
ikke 107
ikke både...og 101
verken ... eller 105
brøker
ekteBrøk 118
sjablon for 1

C

c 2 2-veis 21
c 2 Pdf( ) 22
Cdf( ) 52
ceiling( ), øvre 19
centralDiff( ) 20
char( ), tegnstreng 20
χ^2Cdf( ) 21
χ^2GOF 22
ClearAZ 23
colDim( ), matrisens kolonnedimensjon 24
colNorm( ), matrisens kolonnenorm 24
conj( ), kompleks konjugert 24
constructMat( ), konstruer matrise . 24
$\cos^{-1}$, invers cosinus 27
cos( ), cosinus 26
$\cosh^{-1}$( ), hyperbolsk, invers cosinus 28
cosh( ), hyperbolsk cosinus 28
cosinus, cos( ) 26
$\cot^{-1}$( ), invers cotangens 29
cot( ), cotangens 29
cotangens, cot( ) 29
$\coth^{-1}$( ), hyperbolsk invers cotangens 30
coth( ), hyperbolsk cotangens 30
cPolyRoots() 31
crossP( ), kryssprodukt 32
$\csc^{-1}$( ), invers cosekans 32
csc( ), cosekans 32
csch^{-1}( ), invers hyperbolsk cosekans 33
csch( ), hyperbolsk cosekans 33
CubicReg, kubisk regresjon 33
cumulativeSum( ), kumulativ sum .. 34
cycle, Løkke 35
Cycle, løkke 35

D

d ( ), første deriverte 190
dager mellom datoer, dbd( ) 36
dbd( ), dager mellom datoer 36
Define, definer 37
Definer 37
Definer BiblOff 38
Definer BiblPriv 38
definer, Define 37
definere
felles (offentlig) funksjon eller program 38
privat funksjon eller program .. 38

deltaList() 39
DelVar, slett variabel 39
delVoid( ), fjern gamle elementer 40
deriverte
    første deriverte, d( ) 190
    numerisk derivert, nDeriv( ) 104
    numerisk derivert, nDerivative( ) 103
desimal
    heltall vise, 4Grunntall10 18
    vinkel-visning, ▸DD 36
det( ), matrisedeterminant 40
diag( ), matrisediagonal 41
dim( ), dimensjon 41
dimensjon, dim( ) 41
DispAt 42
dividere heltall, intDiv( ) 73
dividere, / 181
dotP( ), prikk produkt 44

E

e eksponent
    sjablon for 2
e i en potens, e^( ) 44, 50
E, eksponent 195
e^( ), e i en potens 44
eff), omregn nominell til effektiv
        rente 45
effektiv rente, eff( ) 45
egendefinerte funksjoner 37
egendefinerte funksjoner og
        programmer 38
egenvektor, eigVc( ) 45
egenverdi, eigVl( ) 46
eigVc( ), egenvektor 45
eigVl( ), egenverdi 46
ekskludering med "|" operator 199
eksponensiell regresjon, ExpReg 51
eksponent, E 195
eksponenter
    sjablon for 1
ekte brøk, propFrac 118
eliminasjonsform, ref( ) 126
eller (Boolsk), eller 111
eller, Boolsk operator 111
else if, ElseIf 46
else, Else 70
ElseIf, else if 46
en-variabel-statistikk, OneVar 110
end
    For...EndFor 55
    if, EndIf 70
    stigningstall, EndLoop 91
    while, EndWhile 172
end if, EndIf 70
end stigningstall, EndLoop 91
end while, EndWhile 172
endfunksjon, EndFunc 59
Endret internrente av retur, mirr( ), 97
EndTry, avslutt prøv 162
EndWhile, end while 172
enhetsvektor, unitV( ) 168
enten ... eller ..., Boolsk eksklusiv eller 172
EOS (Equation Operating System) 206
Equation Operating System
        (Ligningsoperativsystem)
        (EOS) 206
er lik, = 185
erstatning med "|" operator 199
etikettNavn, Lbl 77
euler( ), Euler function 47
Exit, avslutt 49
exp( ), e i en potens 50
expr( ), streng til uttrykk 51
ExpReg, eksponensiell regresjon 51

F

factor( ), faktor 52
faktor, factor( ) 52
fakultet, ! 189
feil og problemløsing
    send feil, SendFeil 113
    slett feil, SlettFeil 23
Fill, matrise fylle 53
finansielle funksjoner, tvmFV( ) 164
finansielle funksjoner, tvmI( ) 165
finansielle funksjoner, tvmN( ) 165
finansielle funksjoner, tvmPmt( ) 165
finansielle funksjoner, tvmPV( ) 166

FiveNumSummary 53
fjerdegrads regresjon, QuartReg ... 120
fjern
åpne elementer fra liste 40
floor( ), nedre 54
For 55
for, For 55
For, for 55
fordelingsfunksjoner
binomCdf( ) 19, 74-75
binomPdf( ) 19
c 2 Pdf( ) 22
c22-veis( ) 21
invNorm( ) 75
invt( ) 75
normCdf( ) 106
normPdf( ) 107
poissCdf( ) 114
poissPdf( ) 114
tCdf( ) 158
tPdf( ) 161
χ^2Cdf( ) 21
χ^2GOF( ) 22
fordelingssfunksjoner
Invχ^2( ) 74
Forespør 128
ForespørStr 130
format( ), formatstreng 55
formatstreng, format( ) 55
fortegn, sign( ) 143
fpart( ), funksjonsdel 56
frekvens( ) 57
fremstilling i
grader/minutter/sekunder 196
fremstilling minutter, ... 196
fremstilling sekunder, " 196
freqTable( ) 56
Frobenius-norm, norm( ) 106
Func, funksjon 59
Func, programfunksjon 59
funksjoner
del, fpart( ) 56
egendefinere 37
programfunksjon, Func 59
funksjoner og variabler
kopiere 25
første deriverte
sjablon for 5

G

G, gradianer 195
gcd( ), største felles divisor 59
geomCdf( ) 60
geomPdf( ) 60
getDenom( ), lesNevner 61
getKey() 62
getLangInfo( ), hent/returner
språkinformasjon 65
getLockInfo( ), tester låsestatus av
variabel eller variabelgruppe 66
getNum( ), les/returner teller 67
GetStr 67
getType( ), get type of variable 68
getVarInfo( ), les/returner
variabelinformasjon 68
gjennomsnitt, mean( ) 93
gjennomsnittlig endringshastighet,
avgRC( ) 15
Goto, gå til 69
grader fremstilling, ° 196
grader/minutter/sekunder-visning,
▸DMS 43
gradian fremstilling, G 195
grupper, låse og låse opp 88, 168
grupper, teste låsestatus 66
gå til, Goto 69

H

heksadesimal
indikator, 0h 201
vise, 4Grunntall16 18
heltall, int( ) 73
heltallsdel, iPart( ) 76
Hent 60
hent/returner
variabelinformasjon, getVarInfo(
) 65, 68
hurtigtaster 204

hvis, Hvis 70
hyperbolic (hyperbolsk)
invers tangens, $\tanh^{-1}$( ) 158
hyperbolsk
cosinus, cosh( ) 28
invers cosinus, $\cosh^{-1}$( ) 28
invers sinus, $\sinh^{-1}$( ) 146
sinus, sinh( ) 146
tangens, tanh( ) 157
høyre( ), høyre 131
høyre, høyre( ) 131
høyre, right( ) 73

I

identitet( ), identitetsmatrise 69
identitetsmatrise, identitet( ) 69
If, if 70
ifFn( ) 71
ikke både...og, Boolsk operator 101
ikke, Boolsk operator 107
imag( ), imaginær del 72
imaginær del, imag( ) 72
Indir.ref, # 194
innenfor streng, inString( ) 72
innstillinger, les aktuell 66
inString( ), innenfor streng 72
int( ), heltall 73
intDiv( ), dividere heltall 73
integral, ∫ 191
interpolere( ), interpolere 73
invers cosinus, $\cos^{-1}$( ) 27
invers kumulativ normalfordeling
(invNorm( ) 75
invers sinus, sin / ( ) 145
invers tangens, $\tan^{-1}$( ) 157
invers, x^{-1} 198
invF( ) 74
invNorm( ), invers kumulativ
normalfordeling) 75
invt( ) 75
Invχ^2( ) 74
iPart( ), heltallsdel 76
irr( ), internrente
internrente, irr( ) 76
isPrime( ), primtest 76
isVoid( ), test for tomrom 77

K

kolUtvid 23
kombinasjoner, nCr( ) 102
Kommandoen Wait 170
kommentar, © 201
kompleks
konjugert, conj( ) 24
konstruer matrise, constructMat( ) . 24
kopiere variabel eller funksjoner,
CopyVar 25
korrelasjonsmatrise, corrMat( ) 26
korrMat( ), korrelasjonsmatrise 26
kryssprodukt, crossP( ) 32
kubisk regresjon, CubicReg 33
kumulativ sum, cumulativeSum( ) .. 34
kvadratisk regresjon, QuadReg 119
kvadratrot
sjablon for 1
kvadratrot, √( ) 191
kvadratrot, ‡ ( ) 150

L

lagre
symbol, & 200
Lbl, etikettNavn 77
lcm, minste felles multiplum 78
left( ), venstre 78
legg til, & 189
lengde på streng 41
les
modus, lesModus( ) 140
les/returner
lesNevner, getDenom( ) 61
lesTeller, getNum( ) 67
lesModus( ), les modus 140
lesModus( ), les modus-innstillinger . 66
libShortcut( ), lage snarveier til
bibliotekobjekter 78
ligningssystemer (2-ligning)
sjablon for 3

ligningssystemer (N-ligning)
sjablon for 3
lineær regresjon, LinRegAx 80
lineær regresjon, LinRegBx 79, 81
LinRegBx, lineær regresjon 79
LinRegMx, lineær regresjon 80
LinRegtIntervals, lineær regresjon 81
LinRegtTest 83
linSolve() 84
Δlist( ), differensliste 85
list▸mat( ), liste til matrise 85
liste til matrise, list▸mat( ) 85
liste, antall betingede elementer i 31
liste, antall elementer i 30
lister
differens, @liste( ) 85
differenser i en liste, @list( ) 85
kryssprodukt, crossP( ) 32
kumulativ sum, cumulativeSum( ) 34
liste til matrise, list▸mat( ) 85
maksimum, max( ) 93
matrise til liste, mat Δliste( ) 92
midtstreng, mid( ) 96
minimum, min( ) 96
nye, newList( ) 103
prikk produkt, dotP( ) 44
produkt, product( ) 118
sorter fallende, SortD 149
sorter stigende, SortA 148
summering, sum( ) 154
tomme elementer i 202
utvid/sett sammen, utvid( ) 14
ln( ), naturlig logaritme 85
LnReg, logaritmisk regresjon 86
Local, lokal variabel 87
Logaritme
sjablon for 2
logaritmer 85
logaritmisk regresjon, LnReg 86
logisk dobbel implikasjon, ⇔ 189
logisk implikasjon, ⇒ 188, 204
Logistic, logistisk regresjon 89
LogisticD, logistisk regresjon 90
logistisk regresjon, Logistic 89
logistisk regresjon, LogisticD 90
lokal variabel, Local 87
lokal, Local 87
Loop, stigningstall 91
LU (lower-upper), matrisens nedre-øvre dekomposisjon 92
Lås, lås variabel eller variabelgruppe 88
låse opp variabler og variabelgrupper 168
låse variabler og variabelgrupper 88

M

maksimum, max( ) 93
mat▸list( ), matrise til liste 92
matrices (matriser)
undermatrise, subMat( ) 155
matrise (1 × 2)
sjablon for 4
matrise (2 × 1)
sjablon for 4
matrise (2 × 2)
sjablon for 4
matrise (m × n)
sjablon for 4
matrise til liste, mat Δliste( ) 92
matriser
determinant, det( ) 40
diagonal, diag( ) 41
dimensjon, dim( ) 41
egenvektor, eigVc( ) 45
egenverdi, eigVl( ) 46
eliminasjonsform, ref( ) 126
fill, Fylle 53
identitet, identitet( ) 69
kolonnedimensjoner, colDim( ) 24
kolonnenorm, colNorm( ) 24
kumulativ sum, cumulativeSum( ) 34
liste til matrise, list▸mat( ) 85
maksimum, max( ) 93
matrise til liste, mat Δliste( ) 92
minimum, min( ) 96
nedre-øvre dekomposisjon, LU (lower-upper) 92

nye, newMat( ) 104
prikk addisjon, .+ 183
prikk divisjon, .P 184
prikk multiplikasjon, .* 183
prikk potens, .^ 184
prikk subtraksjon, .N 183
produkt, product( ) 118
QR faktorisering, QR 119
radaddisjon, rowAdd( ) 135
raddimensjon, rowDim( ) 135
radhandling, mRow( ) 98
radmultiplikasjon og addisjon, mRowAdd( ) 98
radnorm, rowNorm( ) 135
radskift, rowSwap( ) 135
redusert eliminasjonsform, rref( ) 135
summering, sum( ) 154
tilfeldig, tilfMat( ) 124
transponert, T 156
undermatrise, subMat( ) 154
utvid/sett sammen, augment( ) 14
max( ), maksimum 93
mean( ), gjennomsnitt 93
med (gitt at), | 199
median-median linjeregresjon, MedMed 94
median( ), median 94
median, median( ) 94
MedMed, median-median linjeregresjon 94
mid( ), midtstreng 96
midtstreng, mid( ) 96
min( ), minimum 96
mindre enn eller lik, { 187
minimum, min( ) 96
minste felles multiplum, lcm 78
mirr( ), endret internrente av retur 97
mod( ), modul 97
modul, mod( ) 97
modus-innstillinger, lesModus( ) 66
moduser
lesing, lesModus( ) 140
mRow( ), matrise radhandling 98
mRowAdd( ), matrise radmultiplikasjon og addisjon 98
multiplisere, * 180
Multippel lineær regresjon ttest 100
MultReg 98
MultRegIntervals( ) (MultRegIntervaller) 99
MultRegTests( ) (MultRegTester) 100

N

n-te rot
sjablon for 1
naturlig logaritme, ln( ) 85
nCr( ), kombinasjoner 102
nDerivative( ), numerisk derivert 103
nedre, floor( ) 54
negasjon, legge inn negative tall 207
netto nåverdi, npv( ) 108
newList( ), ny liste 103
newMat( ), ny matrise 104
nfMax( ), numerisk funksjonsmaksimum 104
nfMin( ), numerisk funksjonsminimum 104
nInt( ), numerisk integral 105
nom), omregn effektiv til nominell rente 105
nominell rente, nom( ) 105
norm( ), Frobenius-norm 106
normCdf( ) 106
normPdf( ) 107
nPr( ), permutasjoner 108
npv( ), netto nåverdi 108
nSolve( ), numerisk løsning 109
numerisk
derivert, nDeriv( ) 104
derivert, nDerivative( ) 103
integral, nInt( ) 105
løsning, nSolve( ) 109
ny
liste, newList( ) 103
matrise, newMat( ) 104
når, when( ) 171

O

objekter
lage snarveier til bibliotek 78
omregne
▸Rad 122
4Grad 69
omregnings-operator (#) 207
OneVar, en-variabel-statistikk 110
operatorer
rekkefølge av behandling 206
ord( ), numerisk tegnkode 112
Overfører øyeblikkelig kontroll til den neste iterasjonen i aktuell løkke (For, While, eller Loop). 35

P

P▸Rx( ), rektangulær x-koordinat ... 112
P▸Ry( ), rektangulær y-koordinat ... 113
Pdf( ) 56
permutasjoner, nPr( ) 108
poissCdf( ) 114
poissPdf( ) 114
polar
koordinat, R▸Pr( ) 122
koordinat, R▸Pθ( ) 122
vektor-visning, ▸Polar 115
polyEval( ), behandle polynom 115
polynomer
behandle, polyEval( ) 115
tilfeldig, tilfPoly( ) 124
PolyRoots() 116
potens, ^ 182
potensregresjon, PowerReg 116, 128, 130, 159
PowerReg, potensregresjon 116
Prgm, definer program 117
prikkP
addere, .+ 183
divisjon, ./ 184
multiplikasjon, .* 183
potens, .^ 184
produkt, dotP( ) 44
subtraksjon, .- 183
primtallstest, isPrime( ) 76
prodSeq() 118
product( ), produkt 118
produkt, P( ) 191
produkt, product( ) 118
programmer
definere felles (offentlig) bibliotek 38
definere privat bibliotek 38
programmer og programmering
slett feil, SlettFeil 23
try, Try 162
vis I/O-skjerm, Vis 137
vis I/O skjerm, Vis 41
programmering
definer program, Prgm 117
send feil, SendFeil 113
vis data, Vis 41, 137
propFrac, ekte brøk 118
prosent, % 185

Q

QR faktorisering, QR 119
QR, QR faktorisering 119
QuadReg, kvadratisk regresjon 119
QuartReg, fjerdegrads regresjon ... 120

R

R, radian 195
R▸Pr( ), polarkoordinat 122
R▸Pθ( ), polarkoordinat 122
radian, R 195
RandSeed, tilfeldig startverdi 125
real( ), reell 125
redusert eliminasjonsform, rref( ) .. 135
reell, real( ) 125
ref( ), eliminasjonsform 126
RefreshProbeVars 127
regresjon
fjerdegrads, QuartReg 120
kvadratisk, QuadReg 119
lineær regresjon, LinRegAx 80
lineær regresjon, LinRegBx 79

logaritmisk, LnReg 86
Logistisk 89
logistisk, Logistic 90
median-median-linje, MedMed 94
MultReg 98
potensregresjon, PowerReg 116
sinus, SinReg 147
Regresjon
lineær regresjon, LinRegBx 81
regresjoner
eksponensiell, ExpReg 51
kubisk, CubicReg 33
potensregresjon, PowerReg 116, 128, 130, 159
rektangulær vektor-visning, ▸Rect 125
rektangulær x-koordinat, P▸Rx( ) 112
remain( ), rest 128
resiprok, x^{-1} 198
rest, remain( ) 128
resultat verdier, statistikker 151
resultater, statistikk 150
Return, retur 131
right, right( ) 47, 171
rk23( ), Runge Kutta-funksjon 131
rotate( ), rotere 133
rotere, rotate( ) 133
round( ), avrund 134
rowAdd( ), radaddisjon i matrise 135
rowDim(), raddimensjon i matrise 135
rowNorm(), radnorm i matrise 135
rowSwap(), radskift i matrise 135
rref( ), redusert eliminasjonsform 135

S

sannsynlig normalfordeling, normCdf( ) 106
sannsynlig student- t -fordeling, tCdf( ) 158
sannsynlighetstetthet for student- t, tPdf() 161
sannsynlighetstetthet for student-t, tPdf( ) 161
sannsynlighetstetthet, normPdf( ) 107
sec^{-1}( ), invers sekans 136
sec( ), sekans 136
$sech^{-1}$( ), invers hyperbolsk sekans 137
sech( ), hyperbolsk secant 137
sekvens, seq( ) 138
send feil, SendFeil 113
SendFeil, send feil 113
seq( ), sekvens 138
seqGen( ) 138
seqn( ) 139
sequence, seq( ) 138-139
sfærisk vektor-visning, ▸Sphere 149
shift( ), skift 142
sign( ), fortegn 143
simult( ), simultane ligninger 143
simultane ligninger, simult( ) 143
sin^{-1}( ), invers sinus 145
sin( ), sinus 145
$sinh^{-1}$( ), hyperbolsk invers sinus 146
sinh( ), hyperbolsk sinus 146
SinReg, sinusregresjon 147
sinus, sin( ) 145
sinusregresjon, SinReg 147
sjabloner
(∏), produkt 5
2-delers stykkevis funksjon 2
Absoluttverdi 3-4
andrederivert 6
bestemt integral 6
brøk 1
e eksponent 2
eksponent 1
første deriverte 5
kvadratrot 1
ligningssystemer (2-ligning) 3
ligningssystemer (N-ligning) 3
Logaritme 2
matrise (1 × 2) 4
matrise (2 × 1) 4
matrise (2 × 2) 4
matrise (m × n) 4
n-te rot 1
stykkevis funksjon (N-delers) 2
sum (G) 5
skift, shift( ) 142

slett
feil, SlettFeil 23
åpne elementer fra liste 40
slettAZ 23
slette
variabel, DelVar 39
SlettFeil, slett feil 23
snarveier, tastatur 204
SortA, sorter stigende 148
SortD, sorter fallende 149
sorterer
fallende, SortD 149
stigende, SortA 148
språk
hente språkinformasjon 65
sqrt( ), kvadratrot 150
standardavvik, stdDev( ) 151-152, 169
stat.resultater 150
stat.verdier 151
statistikk
en-variabel-statistikk, OneVar 110
fakultet, ! 189
gjennomsnitt, mean( ) 93
kombinasjoner, nCr( ) 102
median, median( ) 94
permutasjoner, nPr( ) 108
standardavvik, stdDev( ) 151-152, 169
tilfdeldig startverdi, RandSeed 125
tilfeldig norm, tilfNorm( ) 124
to-variable resultater, TwoVar 166
varians, variance( ) 169
stdDevPop( ), populasjonens
standardavvik 151
stdDevSamp( ), utvalgets
standardavvik 152
stigningstall, Loop 91
Stoppkommando 153
streng
dimensjon, dim( ) 41
lengde 41
strenger
brukes for å opprette
variabelnavn 207
format, format( ) 55
formatering 55
høyre, høyre( ) 73, 131
Indir.ref, # 194
innenfor, inString 72
legg til, & 189
midtstreng, mid( ) 96
rotere, rotate( ) 133
skift, shift( ) 142
streng til uttrykk, expr( ) 51
tegnkode, ord( ) 112
tegnstreng, char( ) 20
uttrykk til streng, string( ) 153
venstre, left( ) 78
string( ), uttrykk til streng 153
strings
right, right( ) 47, 171
stykkevis funksjon (N-delers)
sjablon for 2
stykkevis( ) 114
større enn eller lik med, | 188
større enn, > 187
største felles divisor, gcd( ) 59
subMat( ), undermatrise 154-155
subtrahere, - 179
sum (G)
sjablon for 5
sum av hovedbetalinger 193
sum av rentebetalinger 192
sum( ), summering 154
sum, Σ( ) 192
sumIf( ) 154
summering, sum( ) 154
sumSeq() 155
svar (siste), Ans 12

T

T, transponert 156
t test, tTest 163
tan⁻¹( ), invers tangens 157
tan( ), tangens 156
tangens, tan( ) 156
tanh⁻¹( ), hyperbolsk invers tangens 158
tanh( ), hyperbolsk tangens 157
tCdf(), sannsynlig student t-fordeling 158

tegn
streng, char( ) 20
tegnkode, ord( ) 112
tegnstreng, char( ) 20
Tekstkommando 159
teller dager mellom datoer, dbd( ) 36
tellIf( ), antall betingede elementer i en liste 31
test for tomrom, isVoid( ) 77
Test_2S, 2_utvalg F test 58
tidsverdi for penger, antall betalinger 165
tidsverdi for penger, betalingsbeløp 165
tidsverdi for penger, Fremtidig verdi 164
tidsverdi for penger, nåverdi 166
tidsverdi for penger, Rente 165
tier-potens, 10^( ) 198
tilf( ), tilfeldig nummer 123
tilfBin, tilfeldig tall 123
tilfeldig
matrise, tilfMat( ) 124
norm, tilfNorm( ) 124
polynom, tilfPoly( ) 124
startverdi, RandSeed 125
tilfeldig utvalg 124
tilfInt( ), tilfeldig heltall 123
tilfNorm( ), tilfeldig norm 124
tilfPoly( ), tilfeldig polynom 124
tilfSamp( ) 124
tilMat( ), tilfeldig matrise 124
tilnærmet, approx( ) 12
tInterval, t konfidensintervall 159
tInterval_2Samp, toutvalg t konfidensintervall 160
to-variable resultater, TwoVar 166
tomme (åpne) elementer 202
tomrom, test for 77
trace( ) 161
transponert, T 156
Try, feil håndteringskommando 162
try, Try 162
Try, try 162
tTest, t test 163
tTest_2Samp, to-utvalgs t-test 163
TVM-argumenter 166
tvmFV( ) 164
tvmI( ) 165
tvmN( ) 165
tvmPmt( ) 165
tvmPV( ) 166
TwoVar, to-variable resultater 166

U

ulik, ≠ 186
undermatrise, subMat( ) 154-155
unitV( ), enhetsvektor 168
unLock, lås opp variabel eller variabelgruppe 168
uttrykk
streng til uttrykk, expr( ) 51
utvid/sett sammen, augment( ) 14

V

variabel
opprette navn fra en tegnstreng 207
variabler
lokal, Local 87
slette alle enkelttegn 23
slette, DelVar 39
variabler og funksjoner
kopiere 25
variabler, låse og låse opp 66, 88, 168
varians, variance( ) 169
varPop( ) 169
varSamp( ), utvalgets varians 169
varselkoder og meldinger 217
vektorer
enhet, unitV( ) 168
kryssprodukt, crossP( ) 32
Overfører øyeblikkelig kontroll til den neste iterasjonen i aktuell løkke (For, While, eller Loop). 35
prikk produkt, dotP( ) 44
venstre, left( ) 78
verken ... eller, Boolsk operator 105
vinkel, angle( ) 9
vis data, Vis 41, 137
Vis, vis data 41, 137

vise som
binær, 4Grunntall2 16
desimalt heltall, 4Grunntall10 .. 18
desimalvinkel, ▸DD 36
grader/minutter/sekunder, ▸DMS 43
heksadesimal, 4Grunntall16 18
polar vektor, ▸Polar 115
rektangulær vektor, ▸Rect 125
sfærisk vektor, ▸Sphere 149
sylindrisk vektor, ▸Cylind 35

W

warnCodes( ), Warning codes 171
when( ), når 171
while, While 172
While, while 172

X

x^{-1}, resiprok 198
x^2, kvadrat 182
XNOR 189

Z

zInterval, z konfidensintervall 173
zInterval_1Prop, en-proporsjons z konfidensintervall 174
zInterval_2Prop, to-proporsjons z konfidensintervall 174
zInterval_2Samp, to-utvalgs z konfidensintervall 175
zTest 176
zTest_1Prop, en-proporsjons z-test . 176
zTest_2Prop, to-proporsjons z-test . 177
zTest_2Samp, to-utvalgs z-test 178

Ø

øvre, ceiling( ) 19-20, 31

Å

åpne (tomme) elementer 202
åpne elementer, fjern 40