

# TI-Nspire™ CX

## Opslagsvejledning

## ***Vigtige oplysninger***

Med mindre andet er anført i den Licens, som følger med programmet, stiller Texas Instruments ingen garantier, udtrykte eller underforståede, herunder, men ikke begrænset til enhver underforstået garanti om salgbarhed og egnethed til et bestemt formål for nogen programmer eller bøger, og stiller udelukkende disse materialer til rådighed "som de forefindes." Texas Instruments kan under ingen omstændigheder holdes ansvarlige for nogen særlige, indirekte, påløbne eller følgeskader i forbindelse med eller som følge af købet eller anvendelsen af disse materialer, og det eneste erstatningsansvar, Texas Instruments kan pådrage sig, uanset handlingen, kan ikke overstige købsprisen for dette produkt, som den er angivet i licensen. Texas Instruments kan endvidere ikke holdes ansvarlig for nogen form for krav i forbindelse med nogen tredjeparts anvendelse af disse materialer.

© 2024 Texas Instruments Incorporated

De faktiske produkter kan variere let fra de viste billeder.

# Indholdsfortegnelse

<b>Udtryksskabeloner</b> .....	<b>1</b>
<b>Alfabetisk oversigt</b> .....	<b>7</b>
A .....	7
B .....	16
C .....	20
D .....	36
E .....	45
F .....	53
G .....	61
I .....	71
L .....	78
M .....	94
N .....	104
O .....	113
P .....	115
Q .....	122
R .....	125
S .....	140
T .....	161
U .....	173
V .....	174
W .....	175
X .....	177
Z .....	178
<b>Symboler</b> .....	<b>185</b>
<b>TI-Nspire™ CX II - Tegn kommandoer</b> .....	<b>209</b>
Grafikprogrammering .....	209
Grafikskærm .....	209
Standardvisning og indstillinger .....	210
Fejlmeddelelser på grafikskærmen .....	211
Ugyldige kommandoer i grafiktilstand .....	211
C .....	212
D .....	213
F .....	216
G .....	218
P .....	219
S .....	221
U .....	223

<b>Tomme (ugyldige) elementer</b> .....	<b>224</b>
<b>Genveje til indtastning af matematiske udtryk</b> .....	<b>226</b>
<b>Hierarkiet i EOS™ ligningsoperativsystemet (Equation Operating System)</b>	<b>228</b>
<b>TI-Nspire CX II - TI-Grundlæggende programmeringsfunktioner</b> .....	<b>230</b>
Automatisk indrykning i programmeringseditoren .....	230
Forbedrede fejlmeddelelser til TI-Basic .....	230
<b>Konstanter og værdier</b> .....	<b>233</b>
<b>Fejlkoder og fejlmeddelelser</b> .....	<b>234</b>
<b>Fejlkoder- og meddelelser</b> .....	<b>242</b>
<b>Generelle oplysninger</b> .....	<b>244</b>
<b>Indeks</b> .....	<b>245</b>

# Udtryksskabeloner

Udtryksskabeloner er en nem metode til at indsætte matematiske udtryk i matematisk standardnotation. Når du indsætter en skabelon, optræder den i indtastningslinjen med små blokke på positioner, hvor du kan indsætte elementer. En markør viser, hvilket element, du kan indsætte.

Anvend piletasten eller tryk på **tab** for at flytte markøren til hvert elements position, og skriv en værdi eller et udtryk for hvert element. Tryk på **enter** eller **ctrl enter** for at beregne udtrykket.

## Brøkskabelon

**ctrl** **÷** -taster



**Bemærk:** Se også / (divider), side 187.

Eksempel:

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

## Eksponentskabelon

**^** -tast



**Bemærk:** Skriv første værdi, tryk på **^**, og skriv derefter eksponenten. Tryk på højrepilen (**►**) for at hente markøren tilbage til basislinjen.

**Bemærk:** Se også ^ (potens), side 188.

Eksempel:

$$2^3 = 8$$

## Kvadratrodsskabelon

**ctrl** **x<sup>2</sup>** -taster



**Bemærk:** Se også  $\sqrt{\quad}$  (kvadratrod), side 197.

Eksempel:

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

## Nte rod-skabelon

**ctrl** **^** -taster



**Bemærk:** Se også root(), side 137.

Eksempel:

## Nte rod-skabelon

**ctrl** **^** -taster

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

## e ekponentskabelon

**e<sup>x</sup>** -taster

e

Den naturlige eksponentialfunktion  $e$  opløftet til en potens

**Bemærk:** Se også  $e^{()}$ , side 45.

Eksempel:

$$e^1 \quad 2.71828182846$$

## Log-skabelon

**ctrl** **10<sup>x</sup>** -tasten

log  (  )

Beregner logaritmen med et angivet grundtal. Ved 10-talslogaritmen, der er standard, udelades grundtallet.

**Bemærk:** Se også  $\log()$ , side 90.

Eksempel:

$$\log_{10}(2) \quad 0.5$$

## Stykkevis-skabelon (2 stykker)

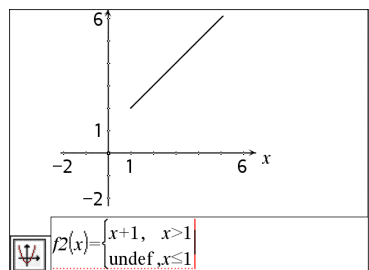
**Katalog** >

{

Gør det muligt at oprette udtryk og betingelser for en stykkevis funktion med to stykker.- Du kan tilføje et stykke ved at klikke på skabelonen og gentage skabelonen.

**Bemærk:** Se også  $\text{piecewise}()$ , side 117.

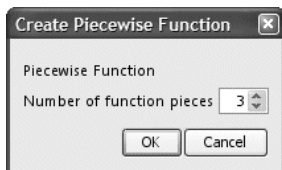
Eksempel:



## Stykkevis-skabelon (N stykker)

Katalog > 

Gør det muligt at oprette udtryk og betingelser for en stykkevis funktion med  $N$ -stykker. Beder om  $N$ .



**Bemærk:** Se også `piecewise()`, side 117.

Eksempel:

Se eksemplet med stykkevis-skabelonen (2 stykker).

## Skabelon til system med 2 ligninger

Katalog > 



Opretter et ligningssystem med to lineære ligninger. Du kan tilføje en række i et eksisterende system ved at klikke i skabelonen og gentage skabelonen.

**Bemærk:** Se også `system()`, side 160.

Eksempel:

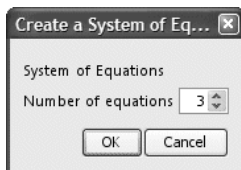
$$\text{solve} \left( \begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y \right) \quad x = \frac{5}{2} \text{ and } y = -\frac{5}{2}$$

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

## Skabelon til system med N ligninger

Katalog > 

Gør det muligt at oprette et system af  $N$  lineære ligninger. Beder om  $N$ .



**Bemærk:** Se også `system()`, side 160.

Eksempel:

Se eksemplet med ligningssystemskabelonen (2-ligninger).

## Absolut værdi-skabelon

Katalog > 



**Bemærk:** Se også `abs()`, side 7.

Eksempel:

## Absolut værdi-skabelon

Katalog > 

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

## dd°mm'ss.ss"-skabelon

Katalog > 

Eksempel:

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

Her kan du indtaste vinkler i **gg°mm'ss.ss''** format, hvor **gg** er antallet af decimalgrader, **mm** er antallet af minutter, og **ss.ss** antallet af sekunder.

## Matrix-skabelon (2 x 2)

Katalog > 

Eksempel:

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

Opretter en matrix 2 x 2.

## Matrix-skabelon (1 x 2)

Katalog > 

Eksempel:

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

## Matrix-skabelon (2 x 1)

Katalog > 

Eksempel:

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

## Matrix-skabelon (m x n)

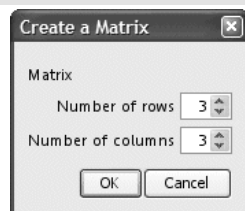
Katalog > 

Skabelonen vises, efter at du er blevet bedt om at angive antallet af rækker og kolonner.

Eksempel:

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





**Bemærk:** Hvis du opretter en matrix med mange rækker og kolonner, kan det tage et øjeblik, før den kommer frem.

Sum-skabelon ( $\Sigma$ )

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

Eksempel:

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

**Bemærk:** Se også  $\Sigma()$  (`sumSeq`), side 198.

Produkt-skabelon ( $\Pi$ )

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

Eksempel:

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

**Bemærk:** Se også  $\Pi()$  (`prodSeq`), side 198.

## Skabelon til differentialkvotient af første orden

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

Eksempel:

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

Skabelonerne til differentialkvotienter af første orden kan anvendes til at beregne differentialkvotienten af første orden i et punkt numerisk med automatiske differentiationsmetoder.

## Skabelon til differentialkvotient af første orden

Katalog > 

**Bemærk:** Se også `d()` (differentialkvotient), side 196.

## Skabelon til differentialkvotient af anden orden

Katalog > 

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

Eksempel:

---

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

---

Skabelonerne til differentialkvotienter af anden orden kan anvendes til at beregne differentialkvotienten af første orden i et punkt numerisk med automatiske differentiationsmetoder.

**Bemærk:** Se også `d()` (differentialkvotient), side 196.

## Bestemt integral skabelon

Katalog > 

$$\int_a^b \square dx$$

Eksempel:

---

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

---

Skabelonen til bestemte integraler kan anvendes til at beregne den bestemte integral numerisk efter samme metode som `nInt()`.

**Bemærk:** Se også `nInt()`, side 107.

# Alfabetisk oversigt

Elementer, hvis navne ikke er alfabetiske (som f.eks. +, ! og >), er anført sidst i dette afsnit, startende (side 185). Medmindre andet er angivet, udføres alle eksempler i dette afsnit i standard nulstillingstilstand, og alle variable antages at være ikke-defineret.

## A

### abs()

Katalog > 

**abs**(*Værdi*) $\Rightarrow$ *værdi*

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

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

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

**abs**(*Matrix*) $\Rightarrow$ *matrix*

Returnerer den absolutte værdi af argumentet.

**Bemærk:** Se også **Absolut værdi-skabelon**, side 3.

Hvis argumentet er et komplekst tal, returneres tallets modulus.

**Bemærk:** Alle udefinerede variable behandles som reelle variable.

### amortTbl()

Katalog > 

**amortTbl**(*NPmt*,*N*,*I*,*PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*afrundVærdi*]) $\Rightarrow$ *matrix*

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

Amortiseringsfunktion, der returnerer en matrix som en amortiseringstabel for et sæt af TVM-argumenter.

*NPmt* er antallet af betalinger, der skal inkluderes i tabellen. Tabellen starter med den første betaling.

*N*, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter (side 171).

- Hvis du udelader *Pmt*, sættes den som standard til  $Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ .
- Hvis du udelader *FV*, sættes  $FV = 0$  som standard.

- Standardværdierne for  $PpY$ ,  $CpY$  og  $PmtAt$  er de samme som for TVM-funktionerne.

$afrundVærdi$  angiver antallet af decimaler til afrunding.  
Standardværdi=2.

Kolonnerne i resultatmatricen er i denne rækkefølge: Betalingsnummer, beløb betalt til renter, beløb betalt til hovedstol og saldo.

Saldoen, der vises i række  $n$ , er saldoen efter betaling  $n$ .

Du kan bruge outputmatricen som input for de andre amortiseringsfunktioner  $\Sigma Int()$  og  $\Sigma Prn()$ , side 199 og  $bal()$ , side 16.

## and

*Boolsk Udtr1 and Boolsk Udtr2*  
 $\Rightarrow$ Boolsk udtryk

*Boolsk Liste1 and Boolsk Liste2*  
 $\Rightarrow$ Boolsk liste

*Boolsk Matrix1 and Boolsk Matrix2*  
 $\Rightarrow$ Boolsk matrix

Returnerer true eller false eller en forenklet form af den oprindelige indtastning.

*Heltal1 and Heltal2* $\Rightarrow$ *heltal*

Sammenligner to heltal bit for bit med en **and**-operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarende bits sammenlignes, er resultatet 1, hvis begge bits er 1. Ellers er resultatet 0. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemtilstand.

I hexadecimal tilstand:

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

Vigtigt: Tallet nul, ikke bogstavet O.

I binær tilstand:

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

I decimal tilstand:

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimal indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talsystem).

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulooperation til at bringe værdien ind i det korrekte område.

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

**Bemærk:** En binær indtastning kan have op til 64 cifre (præfikset 0b ikke medregnet). En hexadecimal indtastning kan have op til 16 cifre.

## angle()

**angle(*Værdi*)** ⇒ *værdi*

Returnerer vinklen på argumentet og fortolker argumentet som et komplekst tal.

I vinkeltilstanden Grader:

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

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

**angle(*Liste1*)** ⇒ *liste*

**angle(*Matrix1*)** ⇒ *matrix*

Returnerer en liste eller matrix med vinkler af elementerne i *Liste1* eller *matrix1*, hvor hvert element fortolkes som et komplekst tal, der repræsenterer et todimensionalt rektangulært koordinatpunkt.

## ANOVA

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

Udfører envejsanalyse af varians til sammenligning af middelværdier for to til 20 populationer. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

*Flag=0* for data, *Flag=1* for statistik

Output-variabel	Beskrivelse
stat.F	Værdien for F-statistik
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader i grupperne
stat.SS	Kvadratsum i grupperne
stat.MS	Middelkvadrat for grupperne
stat.dfError	Frihedsgrader for fejl
stat.SSError	Kvadratsum for fejlene
stat.MSError	Middelkvadrat for fejlene
stat.sp	Puljet standardafvigelse
stat.xbarlist	Gennemsnit af input for listerne
stat.CLowerList	95% konfidensintervaller for middelværdien for hver inputliste
stat.CUpperList	95% konfidensintervaller for middelværdien for hver inputliste

## ANOVA2-way

**ANOVA2way** *Liste1, Liste2*  
*[, Liste3..., Liste10], LevRow*

Beregner en tovejsanalyse af varians til sammenligning af middelværdier for to til ti populationer. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

*LevRow=0* for Blok

*LevRow=2,3,...,Len-1*, for to-faktor, hvor  
*Len=length(List1)=length(List2) = ... = length*  
*(List10)* og *Len / LevRow*  $\in \{2,3,\dots\}$

Output: Blokdesign

Output-variabel	Beskrivelse
stat.F	F statistik for kolonnefaktor
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader i kolonnefaktoren
stat.SS	Kvadratsum for kolonnefaktoren
stat.MS	Middelkvadrat for kolonnefaktoren
stat.FBlok	F statistik for faktor
stat.PValBlock	Mindste sandsynlighed, ved hvilken nul-hypotesen kan forkastes
stat.dfBlock	Frihedsgrader for faktoren
stat.SSBlock	Kvadratsum for faktoren
stat.MSBlock	Middelkvadrat for faktoren
stat.dfError	Frihedsgrader for fejl
stat.SSError	Kvadratsum for fejlene
stat.MSError	Middelkvadrat for fejlene
stat.s	Standardafvigelse for fejlen

#### KOLONNEFAKTOR Output

Output-variabel	Beskrivelse
stat.Fcol	F statistik for kolonnefaktor
stat.PValCol	Sandsynlighedsværdi for kolonnefaktoren
stat.dfCol	Frihedsgrader i kolonnefaktoren
stat.SSCol	Kvadratsum for kolonnefaktoren
stat.MSCol	Middelkvadrat for kolonnefaktoren

#### RÆKKEFAKTOR Output

Output-variabel	Beskrivelse
stat.Frow	F statistik for rækkefaktoren
stat.PValRow	Sandsynlighedsværdi for rækkefaktoren
stat.dfRow	Frihedsgrader for rækkefaktoren

Output-variabel	Beskrivelse
stat.SSRow	Kvadratsum for rækkefaktoren
stat.MSRow	Kvadraternes middelværdi for rækkefaktoren

#### INTERAKTION-output

Output-variabel	Beskrivelse
stat.FInteract	F statistik for interaktionen
stat.PVallInteract	Sandsynlighedsværdi for interaktionen
stat.dflInteract	Frihedsgrader for interaktionen
stat.SSInteract	Kvadratsum for interaktionen
stat.MSInteract	Middelkvadrat for interaktionen

#### FEJL-output

Output-variabel	Beskrivelse
stat.dfError	Frihedsgrader for fejl
stat.SSError	Kvadratsum for fejlene
stat.MSError	Middelkvadrat for fejlene
s	Standardafvigelse for fejlen

#### Ans

ctrl (-) -tasten

Ans⇒værdi

56 56

Returnerer resultatet af de sidst beregnede udtryk.

56+4 60

60+4 64



**approx()**

Katalog &gt;

**approx(Værdi)** ⇒ tal

Returnerer beregningen af argumentet som et udtryk med decimale værdier, når det er muligt, uanset den aktuelle indstilling af **Auto eller tilnærmet**.

Dette svarer til at indtaste argumentet og trykke på .

**approx(Liste)** ⇒ liste**approx(Matrix)** ⇒ matrix

Returnerer en liste eller *matrix*, hvor hvert element er beregnet til en decimalværdi, hvor det er muligt.

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

**►approxFraction()**

Katalog &gt;

*Value* ► **approxFraction(Tol)** ⇒ værdi*List* ► **approxFraction(Tol)** ⇒ liste*Matrix* ► **approxFraction(Tol)** ⇒ matrix

Returnere inputtet som en brøk med en tolerance på *Tol*. Hvis *Tol* udelades, anvendes en tolerance på 5.E-14.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive @>**approxFraction**(...).

$\frac{1}{2} + \frac{1}{3} + \tan(\pi)$	0.833333
$0.8333333333333333 \blacktriangleright \text{approxFraction}(5.E-14)$	$\frac{5}{6}$
$\{\pi, 1.5\} \blacktriangleright \text{approxFraction}(5.E-14)$	$\left\{\frac{5419351}{1725033}, \frac{3}{2}\right\}$

**approxRational()**

Katalog &gt;

**approxRational(Værdi, Tol)** ⇒ værdi**approxRational(Liste, Tol)** ⇒ liste**approxRational(Matrix, Tol)** ⇒ matrix

Returnerer argumentet som en brøk med en tolerance på *Tol*. Hvis *Tol* udelades, anvendes en tolerance på 5.E-14.

$\text{approxRational}(0.333, 5 \cdot 10^{-5})$	$\frac{333}{1000}$
$\text{approxRational}(\{0.2, 0.33, 4.125\}, 5.E-14)$	$\left\{\frac{1}{5}, \frac{33}{100}, \frac{33}{8}\right\}$

**arccos()** Se  $\cos^{-1}()$ , side 27.

**arccosh()** Se  $\cosh^{-1}()$ , side 29.

**arccot()** Se  $\cot^{-1}()$ , side 30.

**arccoth()** Se  $\coth^{-1}()$ , side 30.

**arccsc()** Se  $\csc^{-1}()$ , side 33.

**arccsch()** Se  $\operatorname{csch}^{-1}()$ , side 34.

**arcsec()** Se  $\sec^{-1}()$ , side 141.

**arcsech()** Se  $\operatorname{sech}^{-1}()$ , side 141.

**arcsin()** Se  $\sin^{-1}()$ , side 150.

**arcsinh()** Se  $\sinh^{-1}()$ , side 151.

## augment()

Katalog > **augment(Liste1, Liste2)**⇒liste

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

Returnerer en ny liste, der er *liste2* føjet til enden af *Liste1*.

**augment(Matrix1, Matrix2)**⇒matrix

Returnerer en ny matrix, der er *Matrix2* føjet til *Matrix1*. Når tegnet “,” anvendes, skal matricerne have lige store rækkedimensioner, og *Matrix2* føjes til *Matrix1* som nye kolonner. Ændrer ikke *Matrix1* eller *Matrix2*.

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

## avgRC()

Katalog > **avgRC(Udtryk1, Var [=Værdi] [, Trin])**⇒udtryk

x:=2	2
------	---

**avgRC(Udtryk1, Var [=Værdi] [, Liste1])**⇒liste

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

**avgRC(Liste1, Var [=Værdi] [, Trin])**⇒liste

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

**avgRC(Matrix1, Var [=Værdi] [, Trin])**⇒matrix

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

Returnerer den fremadrettede differenskvotient (gennemsnitlig ændringshastighed).

*Udtr1* kan være et brugerdefineret funktionsnavn (se **Func**).

Når *Værdi* er angivet, tilsidesætter den alle forudgående variabeltildelinger og alle nuværende “|” substitutioner for variabelen.

*Trin* er trinværdien. Hvis *Trin* udelades, er standardværdien 0.001.

Bemærk, at den lignende funktion **centralDiff()** anvender den centrale differenskvotient.

## B

### bal()

**bal**(*NPmt*,*N*,*I*,*PV*,*Pmt*), [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*afrundVærdi*]) $\Rightarrow$ værdi

**bal**(*NPmt*,*amortTabel*) $\Rightarrow$ værdi

Amortiseringsfunktion, der beregner saldo efter en angivet betaling.

*N*, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter (side 171).

*NPmt* angiver betalingsnummeret, hvorefter du vil have dataene beregnet.

*N*, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter (side 171).

- Hvis du udelader *Pmt*, bliver den som standard  $Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ .
- Hvis du udelader *FV*, bliver den som standard  $FV = 0$ .
- Standardværdierne for *PpY*, *CpY* og *PmtAt* er de samme som for TVM-funktionerne.

*afrundVærdi* angiver antallet af decimaler til afrunding.  
Standardværdi=2.

**bal**(*NPmt*,*amortTabel*) beregner saldoen efter betaling nummer *NPmt*, baseret på amortiseringstabel *amortTabel*.  
*amortTabel*-argumentet skal være en matrix i formen beskrevet under **amortTbl()**, side 7.

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

**Bemærk:** Se også  $\Sigma\text{Int}()$  og  $\Sigma\text{Prn}()$ , side 199.

## ►Base2

*Heltal* ►Base2⇒*heltal*

256►Base2

0b100000000

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @►Base2.

0h1F►Base2

0b11111

Konverterer *Heltal* til et binært tal. Binære eller hexadecimalt tal har altid henholdsvis 0b eller 0h som præfiks. Tallet nul, ikke bogstavet O, efterfulgt af b eller h.

0b *binærtTal*

0h *hexadecimaltTal*

Et binært tal kan have op til 64 cifre. Et hexadecimalt tal kan have op til 16.

Uden præfiks behandles *Heltal* som decimaltal (10- talsystem). Resultatet vises som binært uanset tilstanden for talsystem.

Negative tal vises på "2-komplement" form. For eksempel:

-1 vises som

0hFFFFFFFFFFFFFF i det hexadecimaltalsystem 0b111...111 (64 1-taller) i det binære talsystem

-2<sup>63</sup> vises som

0h8000000000000000 i det hexadecimaltalsystem 0b100...000 (63 nuller) i det binære talsystem

Hvis du indtaster et decimalt heltal, der ligger uden området for en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Undersøg følgende eksempler på værdier uden for området.

$2^{63}$  bliver  $-2^{63}$  og vises som

0h8000000000000000 på hexadecimal form  
0b100...000 (63 nuller) . På binær form bliver

$2^{64}$  til 0 og vises som

0h0 på hexadecimal form

0b0 på binær form.

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

0h7FFFFFFFFFFFFFFF hexadecimal form  
0b111...111 (64 1's) på binær form

## ►Base10

*Heltal* ►Base10⇒*heltal*

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

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @►Base10.

0h1F►Base10	31
-------------	----

Konverterer *Heltal* til et decimaltal (i titalssystemet). Binære eller hexadecimal indtastninger skal altid have hhv. 0b eller 0h som præfiks.

0b *binærtTal*

0h *hexadecimalTal*

Tallet nul, ikke bogstavet O, efterfulgt af b eller h.

Et binært tal kan have op til 64 cifre. Et hexadecimal tal kan have op til 16.

Uden præfiks behandles *Heltal* som decimaltal. Resultatet vises som decimaltal uanset tilstanden for talsystem.

*Heltal* ►Base16⇒*heltal*

256►Base16

0h100

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @►Base16.

0b111100001111►Base16

0hFOF

Konverterer *Heltal* til et hexadecimalt tal. Binære eller hexadecimalt tal har altid henholdsvis 0b eller 0h som præfiks.

0b *binærtTal*

0h *hexadecimaltTal*

Tallet nul, ikke bogstavet O, efterfulgt af b eller h.

Et binært tal kan have op til 64 cifre. Et hexadecimalt tal kan have op til 16.

Uden præfiks behandles *Heltal* som decimaltal (10-talssystem). Resultatet vises som hexadecimalt uanset tilstanden for talsystem.

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Yderligere oplysninger findes under ►Base2, side 17.

## binomCdf()

**binomCdf**(*n,p*)⇒*liste*

### binomCdf

(*n,p,nedreGrænse,øvreGrænse*)⇒*tal* hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* if *nedreGrænse* og *øvreGrænse* er lister

**binomCdf**(*n,p,øvreGrænse*)for  $P(0 \leq X \leq \textit{øvreGrænse}) \Rightarrow \textit{tal}$  hvis *øvreGrænse* er et tal, *liste* hvis *øvreGrænse* er en liste

Beregner den kumulerede sandsynlighed for den diskrete binomialfordeling med *n* antal forsøg og sandsynligheden *p* for succes ved hvert forsøg.

**binomCdf()**

Katalog &gt;

For  $P(X \leq \text{øvreGrænse})$ , sæt  $\text{nedreGrænse}=0$ **binomPdf()**

Katalog &gt;

**binomPdf( $n,p$ )** $\Rightarrow$ liste**binomPdf( $n,p$ , $XVærdi$ )** $\Rightarrow$ tal hvis  $XVærdi$  er et tal, liste hvis  $XVærdi$  er en listeBeregner en sandsynlighed ved  $XVærdi$  for den diskrete binomialfordeling med  $n$  antal forsøg og sandsynligheden  $p$  for succes ved hvert forsøg.**C****ceiling()**

Katalog &gt;

**ceiling( $Værdi$ )** $\Rightarrow$ værdi

$\text{ceiling}(.456)$	1.
------------------------	----

Returnerer det nærmeste heltal, der er  $\geq$  argumentet.

Argumentet kan være et reelt eller komplekst tal.

**Bemærk:** Se også **floor()**.**ceiling(Liste)** $\Rightarrow$ liste

$\text{ceiling}(\{-3.1, 1.2, 5\})$	$\{-3., 1, 3\}$
------------------------------------	-----------------

**ceiling(MatrixI)** $\Rightarrow$ matrix

$\text{ceiling}\left(\begin{matrix} 0 & -3.2 \cdot i \\ 1.3 & 4 \end{matrix}\right)$	$\begin{matrix} 0 & -3. \cdot i \\ 2. & 4 \end{matrix}$
--	---

Returnerer en liste eller matrix med oprunding anvendt på hvert element.

**centralDiff()**

Katalog &gt;

**centralDiff( $Udtr1, Var [=Værdi]$ ,  $Trin$ )** $\Rightarrow$ udtryk

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

**centralDiff( $Udtr1, Var$ ,  $Trin$ )** |  $Var=Værdi$  $\Rightarrow$ udtryk**centralDiff( $Udtr1, Var [=Værdi]$ ,  $Trin$ )** $\Rightarrow$ liste**centralDiff( $I, Var [=Værdi]$ ,  $Trin$ )** $\Rightarrow$ liste



**centralDiff**(*Matrix1*, *Var* [= *Værdi*]  
[, *Trin*])  $\Rightarrow$  *matrix*

Returnerer den numeriske differentialkvotient udregnet med formelen for den centrale differenskvotient.

Når *Værdi* er angivet, tilsidesætter den alle forudgående variabeltildelinger og alle nuværende "|" substitutioner for variabelen.

*Trin* er trinværdien. Hvis *Trin* udelades, er standardværdien 0,001.

Ved anvendelse af *Liste1* eller *Matrix1* bliver operationen mappet på tværs af værdierne i listen eller på tværs af matrixelementerne.

**Bemærk:** Se også .

## char()

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

Returnerer en tegnstreng med tegnet nummereret *Heltal* fra grafregnerens tegnsæt. Det gyldige område for *Heltal* er 0–65535.

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

## $\chi^2$ way

**$\chi^2$ way** *obsMatrix*

**chi22way** *obsMatrix*

Beregner en  $\chi^2$  test til association på tovejstabelen med tællinger i den observerede matrix *obsMatrix*. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

Oplysninger om effekten af tomme elementer i en matrix findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat. $\chi^2$	Chi-kvadrat stat: $\text{sum}(\text{observeret} - \text{forventet})^2/\text{forventet}$
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader for Chi-kvadrat stat
stat.ExpMat	Matrix med forventet elementtællingstabel, der antager nulhypotese
stat.CompMat	Matrix med bidrag til chi-kvadrat elementbidrag

## $\chi^2$ Cdf()

Katalog > 

$\chi^2$ Cdf(*nedreGrænse*,*øvreGrænse*,*df*) $\Rightarrow$ *tal*  
 hvis *nedreGrænse* og *øvreGrænse* er tal,  
*liste* hvis *nedreGrænse* og *øvreGrænse* er  
 lister

chi2Cdf(*nedreGrænse*,*øvreGrænse*,*df*) $\Rightarrow$ *tal*  
 hvis *nedreGrænse* og *øvreGrænse* er tal,  
*liste* hvis *nedreGrænse* og *øvreGrænse* er  
 lister

Beregn  $\chi^2$  sandsynlighedsfordelingen mellem *nedreGrænse* og *øvreGrænse* for de angivne frihedsgrader *df*.

For  $P(X \leq \textit{øvreGrænse})$ , sæt *nedreGrænse*=0.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

## $\chi^2$ GOF

Katalog > 

$\chi^2$ GOF *obsListe*,*forvListe*,*fg*

chi2GOF *obsListe*,*forvListe*,*fg*

Udfører en test for at bekræfte, at måledataene er fra en population, der er i overensstemmelse med en angivet distribution. *obsListe* er en liste med antal, og skal indeholde heltal. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat. $\chi^2$	Chi-kvadrat stat: $\text{sum}((\text{observeret} - \text{forventet})^2 / \text{forventet})$
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader for Chi-kvadrat stat
stat.CompList	Bidrag til chi-kvadrat elementbidrag

$\chi^2$ Pdf(*XVal*,*df*) $\Rightarrow$ tal hvis *XVal* er et tal, liste  
hvis *XVal* er en liste

chi2Pdf(*XVal*,*df*) $\Rightarrow$ tal hvis *XVal* er et tal,  
liste, hvis *XVal* er en liste

Beregner tæthedsfunktionen (pdf) for  $\chi^2$  fordelingen ved en angivet *XVal*-værdi for den angivne frihedsgrad *df*.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

**ClearAZ**

$5 \rightarrow b$	5
<i>b</i>	5
ClearAZ	Done
<i>b</i>	"Error: Variable is not defined"

Sletter alle enkelttegnvariable i det aktuelle opgaverum.

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock**, side 174.

**ClrErr**

Se et eksempel på **ClrErr**, i Eksempel 2 under **Try**-kommandoen, side 167.

Sletter fejlstatus og indstiller systemvariabel *errCode* til nul.

**Else** betingelsen i **Try...Else...EndTry**-blokken bør anvende **ClrErr** eller **PassErr**. Brug **ClrErr**, hvis fejlen skal behandles eller ignoreres. Brug **PassErr**, hvis det ikke er kendt, hvad der skal gøres ved fejlen, for at sende den til den næste fejlhåndtering. Hvis der ikke er flere ventende **Try...Else...EndTry**-fejlhåndteringer, vises fejldialogboksen som normalt.

**Bemærk:** Se også **PassErr**, side 116, og **Try**, side 167.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

## colAugment()

Katalog > 

**colAugment**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

Returnerer en ny matrix, der er *Matrix2* føjet til *Matrix1*. Matricerne skal have lige store kolonnedimensioner, og *Matrix2* føjes til *Matrix1* som nye rækker. Ændrer ikke *Matrix1* eller *Matrix2*.

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

## colDim()

Katalog > 

**colDim**(*Matrix*) $\Rightarrow$ *udtryk*

Returnerer antallet af kolonner i *Matrix*.

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

**Bemærk:** Se også **rowDim()**.

## colNorm()

Katalog > 

**colNorm**(*Matrix*) $\Rightarrow$ *udtryk*

Returnerer maksimum for summerne af de absolutte værdier for elementerne i kolonnerne i *Matrix*.

$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$
<b>colNorm</b> ( <i>mat</i> )	9

**Bemærk:** Udefinerede matricielementer er ikke tilladt. Se også **rowNorm()**.

**conj()**

Katalog &gt;

**conj**(*Værdi1*) $\Rightarrow$ *værdi*conj( $1+2\cdot i$ )  $1-2\cdot i$ **conj**(*Liste1*) $\Rightarrow$ *liste*conj( $\begin{pmatrix} 2 & 1-3\cdot i \\ -i & -7 \end{pmatrix}$ )  $\begin{pmatrix} 2 & 1+3\cdot i \\ i & -7 \end{pmatrix}$ **conj**(*Matrix1*) $\Rightarrow$ *matrix*

Returnerer kompleks konjugerede af argumentet.

**Bemærk:** Alle udefinerede variable behandles som reelle variable.**constructMat()**

Katalog &gt;

**constructMat****(**  
*Udtr, Var1, Var2, antalRækker, antalKol*  
**)**  
 $\Rightarrow$ *matrix*constructMat( $\frac{1}{i+j}, i, j, 3, 4$ )  $\begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{pmatrix}$ 

Returnerer en matrix baseret på argumenter.

*Udtr* er et udtryk i variablerne *Var1* og *Var2*. Elementer i den resulterende matrix er dannet ved beregning af *Udtr* for hver forøget værdi af *Var1* og *Var2*.*Var1* er automatisk forøget fra **1** til *antalRækker*. Inden for hver række, *Var2* er forøget fra **1** til *antalKol*.**CopyVar**

Katalog &gt;

**CopyVar** *Var1, Var2*Define  $a(x)=\frac{1}{x}$  *Done***CopyVar** *Var1., Var2.*Define  $b(x)=x^2$  *Done***CopyVar** *Var1, Var2* kopierer værdien af variabelen *Var1* til variabelen *Var2*, og opretter *Var2* hvis nødvendigt. Variablen *Var1* skal have en værdiCopyVar *a, c: c(4)*  $\frac{1}{4}$ CopyVar *b, c: c(4)* 16Hvis *Var1* er navnet på en eksisterende brugerdefineret funktion, kopieres definitionen af denne funktion til funktionen *Var2*. Funktionen *Var1* skal defineres.



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

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

**cos(kvadratMatrix1) ⇒ kvadratMatrix**

Returnerer matrixcosinus af kvadratMatrix1. Dette er ikke det samme som at beregne cosinus for hvert element.

Når en skalær funktion f(A) opererer på kvadratMatrix1 (A), beregnes resultatet efter algoritmen:

Beregn egenverdierne (li) og egenvektorer (Vi) af A.

KvadratMatrix1 skal være diagonaliserbar. Den må heller ikke have symbolske variable, der ikke er tildelt en værdi.

Dan matricerne:

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

Derefter  $A = X B X^{-1}$  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 beregninger udføres aritmetisk med flydende komma.

I vinkeltilstanden Radian:

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

**cos<sup>-1</sup>(Værdi1) ⇒ værdi**

I vinkeltilstanden Grader:

**cos<sup>-1</sup>(Liste1) ⇒ liste**

$\cos^{-1}(\text{Værdi})$  returnerer den vinkel, hvis cosinus er *Værdi*.

$\cos^{-1}(\text{Liste})$  returnerer en liste med de inverse cosinusværdier for hvert element af *Liste*.

**Bemærk:** Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive `arccos (...)`.

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

Returnerer den matrixinverse cosinus af *kvadratMatrix*. Dette er ikke det samme som at beregne den inverse cosinus for hvert element. Oplysninger om beregningsmetoden findes i `cos()`.

*KvadratMatrix* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

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

I vinkeltilstanden radian og rektangulært komplekst format:

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

Du kan se hele resultatet ved at trykke på  $\blacktriangle$  og derefter bruge  $\blacktriangleleft$  og  $\blacktriangleright$  til at bevæge markøren.

$\cosh(\text{Værdi}) \Rightarrow \text{værdi}$

$\cosh(\text{Liste}) \Rightarrow \text{liste}$

$\cosh(\text{Værdi})$  returnerer den hyperperbolske cosinus af argumentet.

$\cosh(\text{Liste})$  returnerer en liste med hyperbolsk cosinus for hvert element i *Liste*.

$\cosh(\text{kvadratMatrix}) \Rightarrow \text{kvadratMatrix}$

Returnerer matrix hyperbolsk cosinus af *kvadratMatrix*. Dette er ikke det samme som at beregne den hyperbolske cosinus for hvert element. Oplysninger om beregningsmetoden findes i `cos()`.

I vinkeltilstanden Grader:

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

I vinkeltilstanden Radian:



## cosh()

Katalog >

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

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

421.255	253.909	216.905
327.635	255.301	202.958
226.297	216.623	167.628

## cosh<sup>-1</sup>()

Katalog >

**cosh<sup>-1</sup>(Værdi)** ⇒ værdi

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

**cosh<sup>-1</sup>(Liste)** ⇒ liste

$$\cosh^{-1}\{1,2,1,3\} \quad \{0,1.37286,1.76275\}$$

**cosh<sup>-1</sup>(Værdi)** returnerer den inverse hyperbolske cosinus af argument.

**cosh<sup>-1</sup>(Liste)** returnerer en liste med de inverse hyperbolske cosinusværdier for hvert element i *Liste*.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccosh** (...).

**cosh<sup>-1</sup>(kvadratMatrix)** ⇒ kvadratMatrix

Returnerer den matrixinverse hyperbolske cosinus af *kvadratMatrix*. Dette er ikke det samme som at beregne den inverse hyperbolske cosinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

I vinkeltilstanden Radian og i rektangulært komplekst format:

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

2.52503+1.73485 <i>i</i>	-0.009241-1.49086 <i>i</i>
0.486969-0.725533 <i>i</i>	1.66262+0.623491 <i>i</i>
-0.322354-2.08316 <i>i</i>	1.26707+1.79018 <i>i</i>

Du kan se hele resultatet ved at trykke på **▶** og derefter bruge **◀** og **▶** til at bevæge markøren.

## cot()

-tast

**cot(Værdi)** ⇒ værdi

I vinkeltilstanden Grader:

**cot(Liste)** ⇒ liste

$$\cot(45) \quad 1$$

## cot()

-tast

Returnerer cotangens af *Værdi1* eller returnerer en liste med cotangens til alle elementer i *Liste1*.

**Bemærk:** Argumentet fortolkes som en vinkel målt i grader, nygrader eller radianer afhængigt af den aktuelt indstillede vinkeltilstand. Du kan bruge °, G eller r til midlertidigt at ignorere vinkeltilstanden.

I vinkeltilstanden Nygrader:

$$\frac{\text{cot}(50)}{\quad} = 1$$

I vinkeltilstanden Radian:

$$\frac{\text{cot}(\{1,2,1,3\})}{\quad} = \{0.642093, -0.584848, -7.01525\}$$

## cot<sup>-1</sup>()

-tast

$\text{cot}^{-1}(\text{Værdi1}) \Rightarrow \text{værdi}$

$\text{cot}^{-1}(\text{Liste1}) \Rightarrow \text{liste}$

Returnerer den vinkel, hvis cotangens er *Værdi1* eller returnerer en liste med den inverse cotangens til hvert element i *Liste1*.

**Bemærk:** Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccot** (...).

I vinkeltilstanden Grader:

$$\frac{\text{cot}^{-1}(1)}{\quad} = 45.$$

I vinkeltilstanden Nygrader:

$$\frac{\text{cot}^{-1}(1)}{\quad} = 50.$$

I vinkeltilstanden Radian:

$$\frac{\text{cot}^{-1}(1)}{\quad} = 0.785398$$

## coth()

Katalog > 

$\text{coth}(\text{Værdi1}) \Rightarrow \text{værdi}$

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

Returnerer den hyperbolske cotangens til *Udtr1* eller returnerer en liste med den hyperbolske cotangens til alle elementer i *Liste1*.

$$\frac{\text{coth}(1.2)}{\quad} = 1.19954$$

$$\frac{\text{coth}(\{1,3,2\})}{\quad} = \{1.31304, 1.00333\}$$

## coth<sup>-1</sup>()

Katalog > 

$\text{coth}^{-1}(\text{Værdi1}) \Rightarrow \text{værdi}$

$\text{coth}^{-1}(\text{Liste1}) \Rightarrow \text{liste}$

$$\frac{\text{coth}^{-1}(3.5)}{\quad} = 0.293893$$

$$\frac{\text{coth}^{-1}(\{-2,2,1,6\})}{\quad} = \{-0.549306, 0.518046, 0.168236\}$$

Returnerer den inverse hyperbolske cotangens til *Værdi1* eller returnerer en liste med den inverse hyperbolske cotangens til hvert element i *Liste1*.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive `arccoth(...)`.

## count()

**count**(*Værdi1* eller *Liste1* [, *Værdi2* eller *Liste2* [...]]) ⇒ *værdi*

Returnerer det akkumulerede antal af alle elementer i argumenterne, der evalueres til numeriske værdier.

Hvert argument kan være et udtryk, en værdi, en liste eller en matrix. Du kan blande datatyper og anvende argumenter med forskellige dimensioner.

For lister, matricer eller celleområder evalueres hvert element for at bestemme, om det skal inkluderes i tællingen.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for ethvert argument.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

<code>count(2,4,6)</code>	3				
<code>count({2,4,6})</code>	3				
<code>count(2, {4,6}, <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>8</td><td>10</td></tr><tr><td>12</td><td>14</td></tr></table>)</code>	8	10	12	14	7
8	10				
12	14				

## countif()

**countif**(*Liste*, *Kriterie*) ⇒ *værdi*

Returnerer det akkumulerede antal af alle elementer i *Liste*, der opfylder de angivne *Kriterie*.

*Kriterie* kan være:

- En værdi, et udtryk eller en streng. For eksempel tæller **3** kun de elementer i *Liste*, der reduceres til værdien 3.

<code>countIf({1,3,"abc",undef,3,1},3)</code>	2
Tæller antallet af elementer lig med 3.	
<code>countIf({"abc","def","abc",3},"def")</code>	1

Tæller antallet af elementer lig med "def."

## countIf()

Katalog > 

- Et Boolsk udtryk, der indeholder symbolet ? som pladsholder for hvert element. For eksempel ?<5 tæller kun de elementer i *Liste*, der er mindre end 5.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for *Liste*.

Tomme (ugyldige) elementer i listen ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

**Bemærk:** Se også `sumIf()`, side 160, og `frequency()`, side 58.

---

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

Tæller 1 og 3.

---

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

Tæller 3, 5 og 7.

---

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

Tæller 1, 3, 7 og 9.

## cPolyRoots()

Katalog > 

`cPolyRoots(Poly,Var)⇒liste`

`cPolyRoots(ListeAfKoeff)⇒liste`

Den første syntaks, `cPolyRoots(Poly,Var)`, returnerer en liste med komplekse rødder af polynomiet *Poly* med hensyn til variabelen *Var*.

*Poly* skal være et polynomium i udviklet form i en variabel. Anvend ikke uudviklede former som  $y^2 \cdot y + 1$  eller  $x \cdot x + 2 \cdot x + 1$

Den anden syntaks, `cPolyRoots(OfCoeffs)`, returnerer en liste med komplekse rødder for koefficienterne i *ListeAfKoeff*.

**Bemærk:** Se også `polyRoots()`, side 119.

---

<code>polyRoots(y<sup>3</sup>+1,y)</code>	{-1}
---	------

---

<code>cPolyRoots(y<sup>3</sup>+1,y)</code>	{-1, 0.5-0.866025 <i>i</i> , 0.5+0.866025 <i>i</i> }
--	--

---

<code>polyRoots(x<sup>2</sup>+2·x+1,x)</code>	{-1,-1}
---	---------

---

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

## crossP()

Katalog > 

`crossP(Liste1, Liste2)⇒liste`

Returnerer vektorproduktet af *Liste1* og *liste2* som en liste.

*Liste1* og *Liste2* skal have ens dimension, og dimensionen skal være 2 eller 3.

---

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

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

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

Returnerer en række eller kolonnevektor (afhængigt af argumenterne), der er vektorproduktet af *Vektor1* og *Vektor2*.

Både *Vektor1* og *Vektor2* skal være rækkevektorer, eller begge skal være kolonnevektorer. Begge vektorer skal have ens dimension, og dimensionen skal være enten 2 eller 3.

**csc()** -tast**csc**(*Værdi1*) ⇒ *værdi*

I vinkeltilstanden Grader:

csc(45)	1.41421
---------	---------

**csc**(*Liste1*) ⇒ *liste*

Returner cosecansen til *Værdi1* eller returnerer en liste med cosecansen til alle elementer i *Liste1*.

I vinkeltilstanden Nygrader:

csc(50)	1.41421
---------	---------

I vinkeltilstanden Radian:

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

**csc<sup>-1</sup>()** -tast**csc<sup>-1</sup>**(*Værdi1*) ⇒ *værdi*

I vinkeltilstanden Grader:

csc <sup>-1</sup> (1)	90.
-----------------------	-----

**csc<sup>-1</sup>**(*Liste1*) ⇒ *liste*

Returnerer den vinkel, hvis cosecans er *Værdi1* eller returnerer en liste med den inverse cosecans til hvert element i *Liste1*.

I vinkeltilstanden Nygrader:

csc <sup>-1</sup> (1)	100.
-----------------------	------

**Bemærk:** Resultatet returneres som en vinkel i nygrader eller radianer afhængigt af den aktuelle vinkeltilstand.

I vinkeltilstanden Radian:

csc <sup>-1</sup> ({1,4,6})	{1.5708, 0.25268, 0.167448}
-----------------------------	-----------------------------

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccsc (...)**.

**csch()**Katalog > **csch**(*Værdi*) ⇒ *værdi*

csch(3) 0.099822

**csch**(*Liste*) ⇒ *liste*csch({1,2,1,4})  
{0.850918,0.248641,0.036644}

Returnerer den hyperbolic cosecans til *Værdi* eller returnerer en liste med de hyperbolske cosecanser til alle elementer i *Liste*.

**csch<sup>-1</sup>()**Katalog > **csch<sup>-1</sup>**(*Tal*) ⇒ *værdi*csch<sup>-1</sup>(1) 0.881374**csch<sup>-1</sup>**(*Liste*) ⇒ *liste*csch<sup>-1</sup>({1,2,1,3})  
{0.881374,0.459815,0.32745}

Returnerer den inverse hyperbolske cosecans til *Værdi* eller returnerer en liste med den inverse hyperbolske cosecans til hvert element i *Liste*.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arccsch (...)**.

**CubicReg**Katalog > **CubicReg** *X*, *Y*, [*Frekv*] [, *Kategori*, *Medtag*]

Beregner polynomiell tredjegradsregression  $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$  på listerne *X* og *Y* med frekvens *Frekv*. En sammenfatning af resultaterne lagres i *stat.resultat* variable. (side 155.)

Alle lister skal have samme dimension med undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hvert tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for *X* og *Y* data.

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
stat.a, stat.b, stat.c, stat.d	Regressionskoefficienter
stat.r <sup>2</sup>	Forklaringsgraden
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> , der faktisk bruges i regressionen ud fra begrænsninger af <i>Hypighed</i> , <i>Kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> , der faktisk bruges i regressionen ud fra begrænsninger af <i>Hypighed</i> , <i>Kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

## cumulativeSum()

**cumulativeSum(Liste1) ⇒ liste**

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

Returnerer en liste med de kumulerede summer af elementerne i *Liste1*, startende ved element 1.

**cumulativeSum(Matrix1) ⇒ matrix**

Returnerer en matrix af de kumulerede summer af elementerne i *Matrix1*. Hvert element er den kumulerede sum af kolonnen fra top til bund.

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

Et tomt (ugyldigt) element i *Liste1* eller *Matrix1* giver et ugyldigt element i den resulterende liste eller matrix. Yderligere oplysninger om tomme elementer findes på side 224.

**Cyklus**

Overfører kontrol direkte til næste iteration i den aktuelle løkke (**For**, **While** eller **Loop**).

**Cycle** må ikke ikke benyttes uden for (**For**, **While** eller **Loop**).

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Funktionsliste, der adderer heltallene fra 1 til 100 og udelader 50.

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

**►Cylind**

*Vektor* ►Cylind

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**Cylind**.

Viser række- eller kolonnevektoren i cylindrisk form [ $r \angle \theta, z$ ].

*Vektor* skal have nøjagtig tre elementer. Det kan være en række eller en kolonne.

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

**D****dbd()**

**dbd(dato1,dato2)**⇒*værdi*

Returnerer antallet af dage mellem *dato1* og *dato2* med tælling af faktiske dage.

*dato1* og *dato2* kan være tal eller lister med tal inden for området af datoer i en standardkalender. Hvis både *dato1* og *dato2* er lister, skal de have samme længde.

*dato1* og *dato2* skal ligge mellem årene 1950 til 2049.

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



Du kan indtaste datoerne i to formater. Placeringen af decimaler er forskellen mellem datoformaterne.

MM.DDÅÅ (almindeligt format i USA)

DDMM.ÅÅ (almindeligt format i Europa)

## ►DD

Tal ►DD⇒værdi

I vinkeltilstanden Grader:

Liste l ►DD⇒liste

$$\overline{(1.5^\circ)\text{►DD} \quad 1.5^\circ}$$

Matrix l ►DD⇒matrix

$$\overline{(45^\circ 22' 14.3'')\text{►DD} \quad 45.3706^\circ}$$

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @>DD.

$$\overline{\left\{ \{ 45^\circ 22' 14.3'', 60^\circ 0' 0'' \} \right\}\text{►DD} \quad \{ 45.3706^\circ, 60^\circ \}}$$

Returnerer den decimale ækvivalent til argumentet udtrykt i grader. Argumentet er et tal, en liste eller matrix, som efter den indstillede tilstand af Vinkel tolkes i grader, nygrader eller radianer.

I vinkeltilstanden Nygrader:

$$\overline{1\text{►DD} \quad \frac{9}{10}^\circ}$$

I vinkeltilstanden Radian:

$$\overline{(1.5)\text{►DD} \quad 85.9437^\circ}$$

## ►Decimal

Værdi l ►Decimal⇒værdi

$$\overline{\frac{1}{3}\text{►Decimal} \quad 0.333333}$$

Liste l ►Decimal⇒værdi

Matrix l ►Decimal⇒værdi

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @>Decimal.

Viser argumentet i decimal form. Denne operator kan kun anvendes ved slutningen af indtastningslinjen.

**Define** *Var* = *Udtryk*

**Define** *Funktion*(*Param1*, *Param2*, ...) = *Udtryk*

Definerer variabelen *Var* eller den brugerdefinerede funktion *Funktion*.

Parametre som *Param1* er pladsholdere til at sætte argumenter ind i funktionen. Ved kald af en brugerdefineret funktion skal du angive argumenter (for eksempel værdier eller variable), der svarer til parametrene. Når den kaldes, evaluerer funktionen *Udtryk* med de angivne argumenter.

*Var* og *Funktion* kan ikke være navnet på en systemvariabel eller en integreret funktion eller kommando.

**Bemærk:** Denne form for **Define** svarer til at eksekvere udtrykket: *udtryk* → *Funktion*(*Param1*,*Param2*).

**Define** *Funktion*(*Param1*, *Param2*, ...) = **Func**  
*Blok*  
**EndFunc**

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

I denne form kan den brugerdefinerede funktion eller programmet eksekvere en blok med flere sætninger.

*Blok* kan være en enkelt sætning eller en række sætninger på separate linjer. *Blok* kan også rumme udtryk og kommandoer (som f.eks. **If**, **Then**, **Else** og **For**).

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

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

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

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

**Bemærk:** Se også **Define LibPriv**, side 39 og **Define LibPub**, side 39.

## Define LibPriv

**Define LibPriv** *Var* = *Udtryk*

**Define LibPriv** *Funktion*(*Param1*, *Param2*, ...) = *Udtryk*

**Define LibPriv** *Funktion*(*Param1*, *Param2*, ...) = **Func**  
*Blok*  
**EndFunc**

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

Fungerer på samme måde som **Define** med den undtagelse, at den definerer en privat biblioteksvariabel, funktion, eller et program. Private funktioner og programmer optræder ikke i Katalog.

**Bemærk:** Se også **Define**, side 38, og **Define LibPub**, side 39.

## Define LibPub

**Define LibPub** *Var* = *Udtryk*

**Define LibPub** *Funktion*(*Param1*, *Param2*, ...) = *Udtryk*

**Define LibPub** *Funktion*(*Param1*, *Param2*, ...) = **Func**  
*Blok*  
**EndFunc**

**Define LibPub** *Program*(*Param1*, *Param2*,

...) = Prgm

Blok

EndPrgm

Fungerer på samme måde som **Define** med den undtagelse, at den definerer en offentlig biblioteksvariabel, funktion, eller et program. Offentlige funktioner og programmer optræder i Katalog, når biblioteket er gemt eller opdateret.

**Bemærk:** Se også **Define**, side 38 og **Define LibPriv**, side 39.

## deltaList()

Se  $\Delta$ List(), side 86.

## DelVar

Katalog > 

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

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

**DelVar** *Var*.

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

Sletter de angivne variable, eller variabelgruppe fra hukommelse.

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

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock**, side 174

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

**DelVar** *Var*. sletter alle elementer i *Var*. variabelgruppe (så som statistikken *stat.nn* resultater, eller variable dannet ved brug af **LibShortcut()**-funktionen).

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

Punktummet (.) i denne form af **DelVar** -kommandoen begrænser den til at slette en variabelgruppe: den simple variabel *Var* berøres ikke.

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

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

getVarInfo()	<table border="1"> <tr> <td><i>aa.a</i></td> <td>"NUM"</td> <td>"0"</td> </tr> <tr> <td><i>aa.b</i></td> <td>"NUM"</td> <td>"0"</td> </tr> <tr> <td><i>aa.c</i></td> <td>"NUM"</td> <td>"0"</td> </tr> </table>	<i>aa.a</i>	"NUM"	"0"	<i>aa.b</i>	"NUM"	"0"	<i>aa.c</i>	"NUM"	"0"
<i>aa.a</i>	"NUM"	"0"								
<i>aa.b</i>	"NUM"	"0"								
<i>aa.c</i>	"NUM"	"0"								

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

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

## delVoid()

Catalog > 

**delVoid(Liste1)** ⇒ *liste*

$\text{delVoid}(\{1, \text{void}, 3\})$   $\{1, 3\}$

Returnerer en liste med indholdet i *Liste1* med alle tomme (ugyldige) elementer fjernet.

Yderligere oplysninger om tomme elementer findes på side 224.

## det()

Catalog > 

**det(kvadratMatrix[, Tolerance])** ⇒ *udtryk*

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



Returnerer determinanten af *kvadratMatrix*.

$\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \text{mat1}$   $\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$

Ethvert matricelement kan valgfrit behandles som nul, hvis den absolutte værdi er mindre end *Tolerance*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers, *Tolerance* ignoreres.

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

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

- Hvis du anvender   eller indstiller **Auto eller tilnærmet**-tilstanden til Approximate, foretages beregningerne med aritmetik med flydende komma.
- Hvis *Tolerance* udelades eller ikke anvendes, beregnes standardtolerancen som:

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

## diag()

Catalog > 

**diag(List)** ⇒ *matrix*

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

**diag(rækkeMatrix)** ⇒ *matrix*

**diag(kolonneMatrix)** ⇒ *matrix*

## diag()

Katalog &gt;

Returnerer en matrix med værdierne i argumentlisten eller matricen i hoveddiagonalen.

**diag(kvadratMatrix)**⇒*rækkeMatrix*

Returnerer en rækkematrix, der indeholder elementerne fra hoveddiagonalen i *kvadratMatrix*.

*kvadratMatrix* skal være kvadratisk.

4	6	8		4	6	8
1	2	3		1	2	3
5	7	9		5	7	9
diag(Ans)				4 2 9		

## dim()

Katalog &gt;

**dim(Liste)**⇒*heltal*

Returnerer dimensionen af *liste*.

**dim(matrix)**⇒*liste*

Returnerer dimensionerne af matricen som en liste med to elementer {rækker, kolonner}.

**dim(Streng)**⇒*heltal*

Returnerer det antal tegn, der er indeholdt i tegnstrengen *Streng*.

dim({0,1,2}) 3

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

dim("Hello") 5

dim("Hello "&"there") 11

## Disp

Katalog &gt;

**Disp udtrykEllerStreng1 [, udtrykEllerStreng2] ...**

Viser argumenterne i *Calculator* historikken. Argumenterne vises efter hinanden med små mellemrum som separator.

Anvendes hovedsagelig i programmer og funktioner til at sikre at mellemregninger vises.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
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** tillader dig at angive den linje, hvor det specificerede udtryk eller streng vil blive vist på skærmen.

Linjeantallet kan angives som et udtryk.

Vær opmærksom på, at linjenummeret ikke gælder hele skærmen, men for området umiddelbart efter kommando/program.

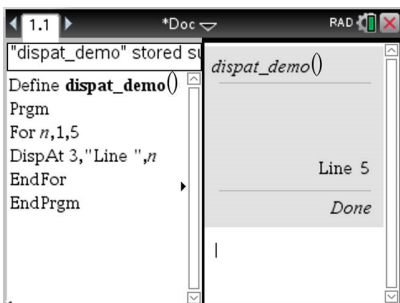
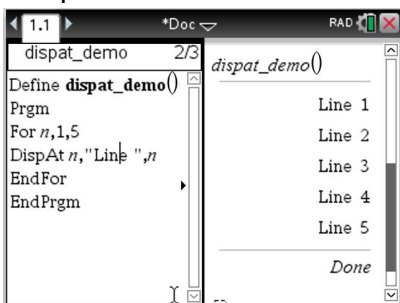
Denne kommando tillader dashboard-lignende output fra programmer, hvor værdien af et udtryk eller fra en sensor-af læsning bliver opdateret på den samme linje.

**DispAt** og **Disp** kan bruges indenfor det samme program.

**Bemærk:** Det maksimale antal er sat til 8, idet det passer til en hel skærm af linjer på den håndholdte skærm - sålænge linjerne ikke har 2D-matematiske udtryk. Det præcise antal linjer afhænger af indholdet af den viste information.

DispAt

### Eksempel



### Konkrete eksempler:

Define z()=	Output
Prgm	z()
For n,1,3	Iteration 1:
DispAt 1,"N: ",n	Linje 1: N:1
Disp "Hallo"	Linje 2: Hallo
EndFor	Iteration 2:
EndPrgm	Linje 1: N:2
	Linje 2: Hallo
	Linje 3: Hallo
	Iteration 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

## Fejlbetingelser:

Fejlmeddelelse	Beskrivelse
DispAt linjeantal skal være mellem 1 og 8	Udtryk evaluerer linjeantallet udenfor rækken 1-8 (inklusive)
For få argumenter	Funktionen eller kommandoen mangler et eller flere argumenter.
Ingen argumenter	Samme som aktuel syntaksfejl-dialog
For mange argumenter	Begræns argument. Samme fejl som Disp.
Ugyldig datatype	Første argument skal være et tal.
Ugyldig: DispAt ugyldig	"Hallo verden" datatypefejl er fundet ugyldig (hvis tilbagekald er defineret)

## DMS

Tal ▶DMS

I vinkeltilstanden Grader:

List ▶DMS

 $\{45.371\}$  ▶DMS  $45^{\circ}22'15.6''$ 

Matrix ▶DMS

 $\{\{45.371,60\}\}$  ▶DMS  $\{45^{\circ}22'15.6'',60^{\circ}\}$ 

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @>DMS.



Tolker argumentet som en vinkel og viser tilsvarende tal for grader (D), minutter (M) og sekunder (S/s) (DDDDDD°MM'SS.ss"). Se mere om DMS-formatet for grader, minutter og sekunder °, ', " på side 203.

**Bemærk:** ►DMS konverterer fra radianer til grader ved anvendelse i radiantilstanden. Hvis inputtet følges af et grader-symbol °, sker der ingen konvertering. Du kan kun anvende ►DMS ved slutningen af en indtastningslinje.

**dotP()**

**dotP(Liste1, Liste2)⇒udtryk**

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

Returnerer "prik" produktet af to lister.

**dotP(Vektor1, Vektor2)⇒udtryk**

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

Returner "prik" produktet af to vektorer.

Begge skal være rækkevektorer, eller begge skal være kolonnevektorer.

**E**

**e^()**

**e^(Værdi1)⇒værdi**

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

Returnerer e opløftet til potensen Værdi1.

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

**Bemærk:** Se også **e Eksponentskabelon**, side 2.

**Bemærk:** At trykke for at vise e^ (er ikke det samme som at trykke på tegnet på tastaturet.

Du kan indtaste et komplekst tal i rei θ polær form. Anvend dog kun denne form i vinkeltilstanden Radian. Den forårsager en domænefejl i vinkeltilstandene Grader eller Nygrader.

**e^()****e^x**-tast**e^(Liste1)**⇒*liste* $e^{\{1,1,0.5\}}$      $\{2.71828,2.71828,1.64872\}$ Returnerer **e** opløftet til potensen af hvert element i *Liste1*.**e^(kvadratMatrix1)**⇒*kvadratMatrix*

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

Returnerer matrix eksponentialfunktion af *kvadratMatrix1*. Dette er ikke det samme som at beregne **e** opløftet til potensen af hvert element. Oplysninger om beregningsmetoden findes i **cos()**.*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.**eff()**

Katalog &gt;

**eff(nominelRente, CpY)**⇒*værdi* $eff(5.75,12)$     5.90398Finansfunktion, der omregner den nominelle rente *nominelRente* til en effektiv årlig rente, hvor *CpY* er antallet af rentetilskrivninger per år.*nominelRente* skal være et reelt tal, og *CpY* skal være et reelt tal > 0.**Bemærk:** Se også **nom()**, side 108.**eigVc()**

Katalog &gt;

**eigVc(kvadratMatrix)**⇒*matrix*

I rektangulært komplekst format:

Returnerer en matrix med egenvektorerne for en reel eller kompleks *kvadratMatrix*, hvor hver kolonne i resultatet svarer til en egenværdi. Bemærk, at en egenvektor ikke er unik. Den kan skaleres af enhver konstantfaktor. Egenvektorerne er normaliseret, dvs. at hvis  $V = [x_1, x_2, \dots, x_n]$ , så:

-1	2	5	→ <i>m1</i>	-1	2	5
3	-6	9		3	-6	9
2	-5	7		2	-5	7

eigVc( <i>m1</i> )		
-0.800906	0.767947	(
0.484029	0.573804+0.052258·i	0.573804-0.052258·i
0.352512	0.262687+0.096286·i	0.262687-0.096286·i

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

Du kan se hele resultatet ved at trykke på **▲** og derefter bruge **◀** og **▶** til at bevæge markøren.

*kvadratMatrix* balanceres først med similaritetstransformationer, til række- og kolonnenormer er så tæt som muligt på samme værdi. *KvadratMatrix* reduceres derefter til øvre Hessenberg form, og egenvektorerne beregnes via en Schur faktorisering.

## eigVI()

**eigVI(*kvadratMatrix*)** ⇒ *liste*

Returnerer en liste med egenværdier af en reel eller kompleks *kvadratMatrix*.

*kvadratMatrix* balanceres først med similaritetstransformationer, til række- og kolonnenormer er så tæt som muligt på samme værdi. *KvadratMatrix* reduceres derefter til øvre Hessenberg form, og egenværdierne beregnes fra øvre Hessenberg-matricen.

I rektangulær kompleks formattilstand:

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

eigVI(*m1*)

{-4.40941,2.20471+0.763006·i,2.20471-0.}

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

## Else

Se If, side 71.

## Elseif

```
If Boolsk Udtr1 Then
  Blok1
Elseif Boolsk Udtr2 Then
  Blok2
  :
Elseif Boolsk UdtrN Then
  BlokN
EndIf
  :
```

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
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 56.

**EndFunc** Se Func, side 60.

**EndIf** Se If, side 71.

**EndLoop** Se Loop, side 93.

**EndPrgm** Se Prgm, side 120.

**EndTry** Se Try, side 167.

**EndWhile** Se While, side 177.

**euler ()** Katalog > 

**euler**(*Expr*, *Var*, *depVar*, {*Var0*,  
*VarMax*}, *depVar0*, *VarStep* [,  
*eulerStep*]) ⇒*matrix*

**euler**(*SystemOfExpr*, *Var*,  
*ListOfDepVars*, {*Var0*, *VarMax*},  
*ListOfDepVars0*, *VarStep* [,  
*eulerStep*]) ⇒*matrix*

**euler**(*ListOfExpr*, *Var*, *ListOfDepVars*,  
{*Var0*, *VarMax*}, *ListOfDepVars0*,  
*VarStep* [, *eulerStep*]) ⇒*matrix*

Anvender Euler-metoden til at løse systemet

Differentialligning:

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

$$\text{euler}(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1) \\ \left[ \begin{array}{cccccc} 0. & 1. & 2. & 3. & 4. & \dots \\ 10. & 10.9 & 11.8712 & 12.9174 & 14.042 & \dots \end{array} \right]$$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

System af ligninger:

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

med  $\text{depVar}(\text{Var}0)=\text{depVar}0$  i intervallet  $[\text{Var}0, \text{VarMax}]$ . Returnerer en matrix, hvor første række definerer  $\text{Var}$ -outputværdierne, og anden række definerer værdien af første løsningskomponent ved de tilsvarende  $\text{Var}$ -værdier, osv.

*Expr* er højresiden, som definerer den ordinære differentialligning (ODE - ordinary differential equation).

*SystemOfExpr* er systemet af højresider, der definerer ODE'erne (svarende til rækkefølgen af afhængige variable i *ListOfDepVars*).

*ListOfExpr* er en liste af højresider, der definerer systemet af ODE'er (svarende til rækkefølgen af afhængige variable i *ListOfDepVars*).

*Var* er den uafhængige variable.

*ListOfDepVars* er en liste med afhængige variable.

$\{\text{Var}0, \text{VarMax}\}$  er en liste med to elementer, der informerer funktionen om, at integrere fra  $\text{Var}0$  til  $\text{VarMax}$ .

*ListOfDepVars0* er en liste med startværdier for afhængige variable.

*VarStep* er et tal forskelligt fra nul, således at  $\text{sign}(\text{VarStep}) = \text{sign}(\text{VarMax}-\text{Var}0)$  og løsninger returneres i  $\text{Var}0+i \cdot \text{VarStep}$  for alle  $i=0,1,2,\dots$  således at  $\text{Var}0+i \cdot \text{VarStep}$  er i  $[\text{var}0, \text{VarMax}]$  (der vil muligvis ikke være en løsningsværdi ved  $\text{VarMax}$ ).

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

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

---


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


---

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

---

*eulerStep* er et positivt heltal (som standard er 1), der definerer antallet af euler-trin mellem outputværdier. Den faktiske trin størrelse, som euler-metoden anvender, er  $VarStep/eulerStep$ .

**eval ()****Hub-menu**

**eval**(*Expr*) ⇒ *string*

**eval()** er kun gyldig i TI-Innovator™ Hub Kommandoargument for programmeringskommandoerne **Get**, **GetStr** og **Send**. Softwaren evaluerer udtrykket *Expr* og udskifter **eval()**-udsagnet med resultatet som en tekststreng.

Argumentet *Expr* skal blot være et reelt tal.

Sæt det blå element i RGB LED til halv intensitet.

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

Nulstil det blå element til OFF.

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

**eval()**-argument skal blot være et reelt tal.

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

Programmet vil langsomt indføre det røde element

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

Kør programmet.

<i>fadein()</i>	Done
-----------------	------

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

Selvom **eval()** ikke viser et resultat, kan du se den resulterende Hub-kommandostreng, efter at du har udført kommandoen ved at inspicere en af følgende specielle variable.

*iostr.SendAns*  
*iostr.GetAns*

*iostr.GetStrAns*

**Bemærk:** Se også **Get** (side 62), **GetStr** (side 68), og **Send** (side 142).

## Exit

Katalog > 

## Exit

Afslutter den aktuelle **For**, **While**, eller **Loop**-blok.

**Exit** er ikke tilladt uden for de tre løkkestrukturer (**For**, **While**, eller **Loop**).

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

## Funktionsliste:

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

## exp()

-tast

**exp**(*Værdi*) $\Rightarrow$ *værdi*

Returnerer **e** opløftet til potensen *Udtr1*.

Returnerer **e** opløftet til potensen *Værdi1*.

**Bemærk:** Se også **e** eksponentskabelon, side 2.

Du kan indtaste et komplekst tal i rel  $\theta$  polær form. Anvend dog kun denne form i vinkeltilstanden Radian. Den forårsager en domænefejl i vinkeltilstandene Grader eller Nygrader.

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

Returnerer **e** opløftet til potensen af hvert element i *Liste1*.

**exp**(*kvadratMatrix1*) $\Rightarrow$ *kvadratMatrix*

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

$e^{\{1,1,.05\}}$	$\{2.71828,2.71828,1.64872\}$
-------------------	-------------------------------

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

## exp()

-tast

Returnerer matrix eksponentialfunktion af *kvadratMatrix1*. Dette er ikke det samme som at beregne **e** opløftet til potensen af hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

## expr()

Katalog > 

**expr**(*Streng*)⇒*udtryk*

Returnerer den tegnstring, der er indeholdt i *Streng* som et udtryk og og eksekverer 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*, *Medtag*]]

Beregner polynomielle tredjegradsregression  $y = a \cdot (b)^x$  på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for *X* og *Y* data.

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer, hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.



Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot (b)^x$
stat.a, stat.b	Regressionskoefficienter
stat.r <sup>2</sup>	Koefficient af en lineær forklaringsgrad til transformerede data
stat.r	Korrelationskoefficient til transformerede data ( $x$ , $\ln(y)$ )
stat.Resid	Residualer af kurvetilpasningen $= y - a \cdot (b)^x$
stat.ResidTrans	Residualer associeret med lineær tilpasning af transformerede data
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> , der faktisk bruges i regressionen ud fra begrænsninger af <i>Hyppeghed</i> , <i>Kategoriliste</i> og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> , der faktisk bruges i regressionen ud fra begrænsninger af <i>Hyppeghed</i> , <i>Kategoriliste</i> og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>



## F

### factor()

**factor(*rationalTal*)** returnerer det rationale tal opløst i primtal. Ved sammensatte tal øges beregningstiden eksponentielt med antallet af cifre i den næststørste faktor. Opløsning af et 30-cifret heltal kan for eksempel vare længere end en dag, og opløsning af et 100-cifret tal kan vare længere end et århundrede.

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

Sådan stopper du en beregning manuelt,

- **Håndholdt:** Hold tasten  **on** nede, mens du gentagne gange trykker på .
- **Windows®:** Hold tasten **F12** nede, mens du gentagne gange trykker på **Enter**.
- **Macintosh®:** Hold tasten **F5** nede, mens

du gentagne gange trykker på **Enter**.

- **iPad®**: App'en viser en meddelelse. Du kan fortsat vente eller annullere.

Hvis du kun vil bestemme, om et tal er et primtal, skal du anvende **isPrime()** i stedet. Det er meget hurtigere, især hvis *RationalTal* ikke er et primtal, og den næststørste faktor har mere end fem cifre.

## FCdf()

**FCdf**

(  
*nedreGrænse*  
,*øvreGrænse*,*fgTæller*,*fgNævner*) $\Rightarrow$ *tal* hvis  
*nedreGrænse* og *øvreGrænse* er tal, *liste*  
hvis *nedreGrænse* og *øvreGrænse* er lister

**FCdf**

(  
*nedreGrænse*  
,*øvreGrænse*,*fgTæller*,*fgNævner*) $\Rightarrow$ *tal* hvis  
*nedreGrænse* og *øvreGrænse* er tal, *liste*  
hvis *nedreGrænse* og *øvreGrænse* er lister

Beregner sandsynligheden hørende til  $F$ -fordelingen for intervallet mellem *nedreGrænse* og *øvreGrænse* med de angivne frihedsgrader *dfTæller* og *dfNævner*.

For den kumulerede fordeling  $P(X \leq \textit{øvreGrænse})$  skal du sætte *nedreGrænse* = 0.

## Fill

**Fill** *Tal*, *matrixVar* $\Rightarrow$ *matrix*

Erstatter hvert element i variabelen *matrixVar* med *Udtr*.

*matrixVar* skal eksistere i forvejen.

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

Fill *Tal*, *listVar* ⇒ *liste*

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

Erstatter hvert element i variabelen *Listevar* med *Udtr.*

Fill 1.01, *alist* Done

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

*Listevar* skal eksistere i forvejen.

## FiveNumSammendrag

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

Frembringer en forkortet version af 1-variabelstatistikken på listen *X*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

*X* repræsenterer en liste med dataene.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hver tilsvarende *X* værdi. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste med numeriske kategorikoder for tilsvarende *X* værdier.

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer, hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Et tomt (ugyldigt) element i en af listerne *X*, *Freq* eller *Category* resulterer i at det tilsvarende element i alle disse lister bliver ugyldigt. Yderligere oplysninger om tomme elementer findes på side 224

Output-variabel	Beskrivelse
stat.MinX	Minimum af x-værdier
stat.Q <sub>1</sub> X	1. kvartil af x
stat.MedianX	Median af x
stat.Q <sub>3</sub> X	3. kvartil af x
stat.MaxX	Maksimum af x-værdier

**floor()**Katalog > **floor**(*Værdi*) ⇒ *heltal*

floor(-2.14) -3.

Returnerer det største heltal, der er ≤ argumentet. Denne funktion er identisk med **int()**.

Argumentet kan være et reelt eller komplekst tal.

**floor**(*Liste1*) ⇒ *liste*

$$\text{floor}\left(\left\{\frac{3}{2}, 0, -5, 3\right\}\right) \quad \{1, 0, -6\}$$
**floor**(*Matrix1*) ⇒ *matrix*

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

Returnerer en liste eller matrix med nedrunding af hvert element.

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

**For**Katalog > **For** *Var*, *Lav*, *Høj* [, *Trin*]  
*Blok***EndFor**

Eksekverer sætningerne i *blok* iterativt for hver værdi af *Var* fra *Lav* til *Høj* i intervaller på *Trin*.

*Var* må ikke være en systemvariabel.

*Trin* kan være positiv eller negativ. Standardværdien er 1.

*Blok* kan enten være en enkelt sætning eller en serie sætninger adskilt med kolon.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

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

## format()

Katalog > 

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

Returnerer *Tal* som en tegnstring baseret på formatskabelonen.

*formatStreng* er en streng og skal være på formen: "F[n]", "S[n]", "E[n]", "G[n][c]", hvor [ ] angiver valgfrie dele.

F[n]: Fast format. n er det antal cifre, der vises efter decimalpunktet.

S[n]: Videnskabeligt format. n er det antal cifre, der vises efter decimalpunktet.

E[n]: Teknisk format. n er antallet af cifre efter det betydende ciffer. Eksponenter er justeret til et multiplum af tre, og decimalpunktet flyttes til højre med nul, en eller to pladser.

G[n][c]: Samme som fast format men skiller også cifrene til venstre for decimalpunktet i grupper på tre. c angiver gruppeskilletegnet og er som standard et komma. Hvis c er et punktum, vises grundtallet som et komma.

[Rc]: Alle ovennævnte angivelser kan udvides med Rc-grundtalflaget, hvor c er et enkelt tegn, der angiver, hvad der skal substitueres for grundtalspunktet.

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

## fPart()

Katalog > 

**fPart**(*Udtr1*) ⇒ *udtryk*

**fPart**(*Liste1*) ⇒ *liste*

**fPart**(*Matrix1*) ⇒ *matrix*

Returnerer decimaldelen af argumentet.

For en liste eller matrix returneres decimaldelen af elementerne.

Argumentet kan være et reelt eller komplekst tal.

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

**F Pdf**(*XVærdi*,*dfTæller*,*dfNævner*)

**FPdf**(*XVærdi*,*dfTæller*,*dfNævner*)

Beregner  $F$  sandsynlighedsfordelingen på *XVal* for den angivne *dfTæller* (frihedsgrader i tælleren) og *dfNævner* (frihedsgrader i nævneren).

## freqTable►list()

**freqTable►liste**

(*Liste1*,*frekvHeltalListe*)⇒*liste*

Returnerer en liste indeholdende elementerne fra *Liste1* udvidet i henhold til hyppighederne i *frekvHeltalListe*. Denne funktion kan anvendes til at danne en frekvenstabel for Data- & Statistikapplikationerne.

*Liste1* kan være enhver gyldig liste.

*frekvHeltalListe* skal have den samme dimension som *Liste1* og må kun indeholde ikke-negative heltalselementer. Hvert element angiver det antal gange det tilsvarende *Liste1* element vil blive gentaget i resultatlisten. En nul-værdi udelukker det tilsvarende *Liste1* element.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **freqTable@>list(...)**.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

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

## frequency()

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

Returnerer en liste, der indeholder optælling af elementerne i *Liste1*. Antallene er baseret på områder (bins), som du definerer i *binsListe*.

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

Hvis *binsListe* er  $\{b(1), b(2), \dots, b(n)\}$ , er de specificerede områder  $\{? \leq b(1), b(1) < ? \leq b(2), \dots, b(n-1) < ? \leq b(n), b(n) > ?\}$ . Den resulterende liste er et element længere end *binsListe*.

Hvert element af resultatet svarer til antallet af elementer fra *Liste1*, der er i dette område. Udtrykt med **countIf()**-funktionen er resultatet  $\{\text{countIf}(\text{liste}, ? \leq b(1)), \text{countIf}(\text{liste}, b(1) < ? \leq b(2)), \dots, \text{countIf}(\text{liste}, b(n-1) < ? \leq b(n)), \text{countIf}(\text{liste}, b(n) > ?)\}$ .

Elementer i *Liste1*, der ikke kan "placeres i en størrelse" ignoreres. Tomme (ugyldige) elementer ignoreres også. Yderligere oplysninger om tomme elementer findes på side 224.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for begge argumenter.

**Bemærk:** Se også **countIf()**, side 31.

Forklaring af resultatet:

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

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

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

Elementet "hello" er en streng og kan ikke placeres i nogen af de definerede områder.

## FTest\_2Samp

**FTest\_2Samp** *Liste1, Liste2[, Hyppighed1 [, Hyppighed2[, Hypot]]]*

**FTest\_2Samp** *Liste1, Liste2[, Hyppighed1 [, Hyppighed2[, Hypot]]]*

(Datalisteinput)

**FTest\_2Samp** *sx1, n1, sx2, n2[, Hypot]*

**FTest\_2Samp** *sx1, n1, sx2, n2[, Hypot]*

(Sammenfatning, stat input)

Udfører en F test med målinger. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

eller  $H_a: \sigma_1 > \sigma_2$ , sæt *Hypot* > 0

Til  $H_a: \sigma_1 \neq \sigma_2$  (standard), sæt *Hypot* = 0

Til  $H_a: \sigma_1 < \sigma_2$ , sæt  $Hypot < 0$

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.F	Beregnet $\hat{U}$ statistik for datasekvensen
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.dfTæller	frihedsgrader for tæller = $n_1 - 1$
stat.dfNævner	tæller, frihedsgrader = $n_2 - 1$
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.x1_bar stat.x2_bar	Middelværdi af stikprøver for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Størrelse på stikprøverne

## Func

## Func

*Blok*

## EndFunc

Skabelon til oprettelse af en brugerdefineret funktion.

*Blok* kan være en enkelt sætning, en serie sætninger adskilt med kolon eller en serie sætninger på separate linjer. Funktionen kan anvende **Return**-instruktionen til at returnere et specifikt resultat.

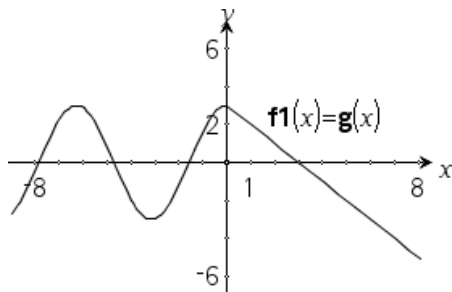
## Bemærk indtastning af

**eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Definition af en stykvis funktion:

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

Tegnet resultat af grafen  $g(x)$





**gcd()**Katalog > **gcd**(*Værdi1*, *Værdi2*) $\Rightarrow$ *udtryk* $\text{gcd}(18,33)$ 

3

Returnerer den største fælles divisor af to argumenter. **Gcd** for to brøker er **gcd** af deres tællere divideret med **lcm** af deres nævnere.

I **Auto** eller **tilnærmet**-tilstand er **gcd** af flydende decimalbrøker 1.0.

**gcd**(*Liste1*, *Liste2*) $\Rightarrow$ *liste* $\text{gcd}(\{12,14,16\},\{9,7,5\})$  $\{3,7,1\}$ 

Returnerer de største fælles divisorer af de tilsvarende elementer i *Liste1* og *Liste2*.

**gcd**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

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

Returnerer de største fælles divisorer af de tilsvarende elementer i *Matrix1* og *Matrix2*.

**geomCdf()**Katalog > **geomCdf**

(*p*,*nedreGrænse*,*øvreGrænse*) $\Rightarrow$ *tal*, hvis *nedreGrænse* og *øvreGrænse* er tal, *liste*, hvis *nedreGrænse* og *øvreGrænse* er lister

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

Beregner den kumulerede geometriske sandsynlighed fra *nedreGrænse* til *øvreGrænse* med den angivne sandsynlighed for succes *p*.

For  $P(X \leq \textit{øvreGrænse})$ , sæt *nedreGrænse*=1.

**geomPdf()**Katalog > 

**geomPdf**(*p*,*XVærdi*) $\Rightarrow$ *tal* hvis *XVærdi* er et tal, *liste* hvis *XVærdi* er en liste

Beregner sandsynligheden i  $XVærdi$ , nummeret på den forsøgsgang hvor den første succes forekommer, for den diskrete geometrisk distribution med den angivne sandsynlighed for succes,  $p$ .

**Get****Hub-menu**

**Get**[*promptString*,] *var*[, *statusVar*]

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

Programmeringskommando: Henter en værdi fra en tilsluttet TI-Innovator™ Hub og tildeler værdien til den variable *var*.

Der skal anmodes om værdien:

- i forvejen gennem en **Send "READ ..."** kommando.  
— eller —
- ved at indlejre en **"READ ..."** anmodning som det valgfrie *promptString*-argument. Med denne metode kan du bruge en enkelt kommando til at anmode om værdien og modtage den.

Implicit reduktion finder sted. For eksempel fortolkes en modtaget streng på "123" som en numerisk værdi. Brug **GetStr** i stedet for **Get** for at bevare strengen.

Hvis du inkluderer det valgfrie argument *statusVar*, tildeles det en værdi baseret på operationens succes. En værdi på nul betyder, at ingen data blev modtaget.

I den anden syntaks lader argumentet *func*() et program gemme den modtagne streng som en funktionsdefinition. Denne syntaks virker, ligesom hvis programmet udførte kommandoen:

Define *func*(*arg1*, ...*argn*) = *modtaget streng*

Eksempel: Anmod om den nuværende værdi for hubbens indbyggede lysniveausensor. Brug **Get** for at modtage værdien, og tildel den til den variable *lightval*.

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

Indlejrer READ-anmodningen med **Get**-kommandoen.

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

Så kan programmet bruge den definerede funktion *func()*.

**Bemærk:** Du kan bruge kommandoen **Get** i et brugerdefineret program, men ikke i en funktion.

**Bemærk:** Se også **GetStr**, side 68 og **Send**, side 142.

## getDenom()

Katalog >

**getDenom**(*Brøkl*) ⇒ *værdi*

Transformerer argumentet til et udtryk med en forkortet fællesnævner og returnerer derefter dens nævner.

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

## getKey()

Katalog >

**getKey**([0|1]) ⇒ **returnString**

**Beskrivelse:** **getKey()** - tillader et TI-Basic-program at få tastaturinput - håndholdt, stationær og emuleret på stationær.

### Eksempel:

- tasttrykket := **getKey()** vil returnere en tast eller en tom streng, hvis ingen tast blev trykket. Denne kommando vil straks returnere.
- tasttrykket := **getKey(1)** venter til en tast bliver trykt ned. Denne kommando vil sætte udførelsen af programmet på pause, indtil en tast er trykket ned.

**getKey()**

Eksempel:

```

Define getKey_demo()
Prgm
Local key
key:=" "
While key!="esc"
key:=getKey(1)
Disp "Key: ",key
EndWhile
EndPrgm
  
```

### Håndtering af tasttryk:

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
Esc	Esc	"esc"
Touchpad - klik øverst	n/a	"op"
Til	n/a	"til top"
Scratchapps	n/a	"kladde"
Touchpad - venstreklik	n/a	"venstre"
Touchpad - klik i midten	n/a	"midten"
Touchpad - højreklik	n/a	"højre"
Dok	n/a	"dok"
Tabulator	Tabulator	"tabulator"
Touchpad - klik nederst	Ned pil	"ned"
menu	n/a	"menu"
Ctrl	Ctrl	ingen retur
Skift	Skift	ingen retur
Var	n/a	"var"
Del	n/a	"del"
=	=	"="
trigonometri	n/a	"trigonometri"
0 til 9	0-9	"0" ... "9"
Skabeloner	n/a	"skabelon"
Katalog	n/a	"kat"
^	^	"^"
X <sup>2</sup>	n/a	"kvadrat"
/ (divisionstast)	/	"/"
* (gangetast)	*	"*"
e <sup>x</sup>	n/a	"eksp"
10 <sup>x</sup>	n/a	"10potens"
+	+	"+"

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
-	-	"_"
(	(	"("
)	)	")"
.	.	". "
(-)	n/a	"-" (negativ fortegn)
Indtast	Indtast	"indtast"
ee	n/a	"E" (eksponentiel notation E)
a - z	a - z	alpha = bogstav trykket ned (lille bogstav) ("a" - "z")
skift a - z	skift a - z	alpha = bogstav trykket ned "A" - "Z"
		Bemærk: ctrl-shift fungerer som lock-caps
?!	n/a	"?!"
pi	n/a	"pi"
Flag	n/a	ingen retur
,	,	" , "
Return	n/a	"retur"
mellemrum	mellemrum	" " (mellemrum)
Utilgængelig	Specielle tegntaster som @, !^, osv.	Tegnet returneres
n/a	Funktionstaster	Ingen returnerede tegn
n/a	Specielle skrivebordskontrolltaster	Ingen returnerede tegn
Utilgængelig	Andre skrivebordstaster, som ikke er tilgængelige på beregneren, når getkey () venter på et tastetryk. {, }, ,, ;, ...)	Samme tegn du får i noter (ikke i et matematik-felt)

**Bemærk:** Det er vigtigt at være opmærksom på, at tilstedeværelsen af `getKey()` i et program ændrer, hvordan visse hændelser, bliver håndteret af systemet. Nogle af disse er beskrevet nedenfor.

**Afslut program og håndter hændelse** - Præcis som hvis en bruger ville afbryde et program ved at trykke på tasten **ON**

"**Support**" nedenfor betyder - system fungerer som forventet - program kører stadig.

Hændelse	Enhed	Skrivebord - TI-Nspire™ Student Software
Hurtig svar	Afslut program, håndter hændelse	Samme som den håndholdte (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-kun)
Fjern fil mgmt  (Inkl. afsendelse af "exit tryk 2 test" fil fra en anden håndholdt eller skrivebordshåndholdt)	Afslut program, håndter hændelse	Samme som den håndholdte.  (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-kun)
Afslut klasse	Afslut program, håndter hændelse	Support  (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-kun)

Hændelse	Enhed	Skrivebords - TI-Nspire™ alle versioner
TI-Innovator™ Hub forbind/afbryd	Support - kan med held udstede kommandoer til TI-Innovator™ Hub. Når du går ud af programmet, arbejder TI-Innovator™ Hub stadig med den håndholdte.	Samme som den håndholdte

## getLangInfo()

Katalog > 

`getLangInfo()` ⇒ streng

`getLangInfo()`

"en"

Returnerer en streng, som svarer til det korte navn af det aktuelle aktive sprog. Man kan, for eksempel, bruge det i et program eller funktion til at bestemme det aktuelle sprog.

Engelsk = "en"  
 Dansk = "da"  
 Tysk = "de"  
 Finsk = "fi"  
 Fransk = "fr"  
 Italiensk = "it"  
 Hollandsk = "nl"  
 Belgisk Hollandsk = "nl\_BE"  
 Norsk = "no"  
 Portugisisk = "pt"  
 Spansk = "es"  
 Svensk = "sv"

## getLockInfo()

**getLockInfo**(*Var*)⇒*værdi*

Returnerer den aktuelle låste/oplåste tilstand på variabelen *Var*.

*værdi* = 0: *Var* er ulåst eller findes ikke.

*værdi* = 1: *Var* er låst og kan ikke ændres eller slettes.

Se **Lock**, side 89, og **unLock**, side 174.

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

## getMode()

**getMode**(*TilstandNavnHeltal*)⇒*værdi*

**getMode**(0)⇒*liste*

**getMode**(*TilstandNavnHeltal*) returnerer en værdi, der repræsenterer den aktuelle indstilling for tilstanden *TilstandNavnHeltal*.

**getMode**(0) returnerer en liste, der indeholder talpar. Hvert par består af et tilstandsheltal og et indstillingsheltal.

Se tabellen nedenfor for en oversigt over tilstande og deres indstillinger.

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

Hvis du gemmer indstillingerne med **getMode(0)** → *var*, kan du anvende **setMode(*var*)** i en funktion eller et program for midlertidigt at gendanne indstillingerne under eksekveringen af funktionen eller programmet. Se **setMode()**, side 145.

Tilstandsnavn	Tilstandsheltal	Indstillingsheltal
Viste cifre	1	1=Float, 2=Float1, 3=Float2, 4=Float3, 5=Float4, 6=Float5, 7=Float6, 8=Float7, 9=Float8, 10=Float9, 11=Float10, 12=Float11, 13=Float12, 14=Fix0, 15=Fix1, 16=Fix2, 17=Fix3, 18=Fix4, 19=Fix5, 20=Fix6, 21=Fix7, 22=Fix8, 23=Fix9, 24=Fix10, 25=Fix11, 26=Fix12
Vinkel	2	1=Radian, 2=Grader, 3=Gradian
Eksponentielt format	3	1=Normal, 2=Videnskabelig, 3=Teknisk
Reel eller komplekse	4	1=Reel, 2=Rektangulær, 3=Polær
Auto eller tilnærmet	5	1=Auto, 2=Tilnærmet
Vektorformat	6	1=Rektangulær, 2=Cylindrisk, 3=Sfærisk
Talsystem	7	1=Decimal, 2=Hex, 3=Binær

## getNum()

**getNum(*Brøk1*)** ⇒ *værdi*

Transformerer argumentet til et udtryk med en forkortet fællesnævner og returnerer derefter dens tæller.

$x:=5; y:=6$	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

**GetStr**[*promptStreng*,] *var*[, *statusVar*]

Se **Get** for eksempler.

**GetStr**[*promptStreng*,] *func*(*arg1*, ...*argn*)



[, *statusVar*]

Programmeringskommando: Fungerer identisk med kommandoen **Get**, bortset fra at den hentede værdi altid fortolkes som en streng. I modsætning hertil fortolker kommandoen **Get** svarene som et udtryk, medmindre det er omsluttet af citationstegn ("").

**Bemærk:** Se også **Get**, side 62 og **Send**, side 142.

## getType()

Katalog > 

**getType**(*var*) ⇒ *streng*

Returnerer en streng, som angiver datatypen for variabelen *var*.

Hvis *var* ikke er defineret, returneres strengen "NONE".

$\{1,2,3\} \rightarrow temp$	$\{1,2,3\}$
<code>getType(temp)</code>	"LIST"
$3 \cdot i \rightarrow temp$	$3 \cdot i$
<code>getType(temp)</code>	"EXPR"
<code>DelVar temp</code>	<i>Done</i>
<code>getType(temp)</code>	"NONE"

## getVarInfo()

Katalog > 

**getVarInfo**() ⇒ *matrix* eller *streng*

**getVarInfo**(*BibNavnStreng*) ⇒ *matrix* eller *streng*

**getVarInfo**() returnerer en matrix med informationer (variabelnavn, type, bibliotekets tilgængelighed og låst/ulåst-status) for alle variable og biblioteksobjekter defineret i den aktuelle opgave.

Hvis der ikke er defineret nogen variable, returnerer **getVarInfo**() strengen "NONE"

<code>getVarInfo()</code>	"NONE"												
Define $x=5$	<i>Done</i>												
Lock $x$	<i>Done</i>												
Define LibPriv $y=\{1,2,3\}$	<i>Done</i>												
Define LibPub $z(x)=3 \cdot x^2 - x$	<i>Done</i>												
<code>getVarInfo()</code>	<table border="1"> <tbody> <tr> <td><math>x</math></td> <td>"NUM"</td> <td>"{ }"</td> <td>1</td> </tr> <tr> <td><math>y</math></td> <td>"LIST"</td> <td>"LibPriv"</td> <td>0</td> </tr> <tr> <td><math>z</math></td> <td>"FUNC"</td> <td>"LibPub"</td> <td>0</td> </tr> </tbody> </table>	$x$	"NUM"	"{ }"	1	$y$	"LIST"	"LibPriv"	0	$z$	"FUNC"	"LibPub"	0
$x$	"NUM"	"{ }"	1										
$y$	"LIST"	"LibPriv"	0										
$z$	"FUNC"	"LibPub"	0										
<code>getVarInfo(tmp3)</code>	"Error: Argument must be a string"												
<code>getVarInfo("tmp3")</code>	<table border="1"> <tbody> <tr> <td><i>volcyL2</i></td> <td>"NONE"</td> <td>"LibPub"</td> <td>0</td> </tr> </tbody> </table>	<i>volcyL2</i>	"NONE"	"LibPub"	0								
<i>volcyL2</i>	"NONE"	"LibPub"	0										

**getVarInfo**

(*Biblioteksnavnstreng*) returnerer en matrix med oplysninger for alle biblioteksobjekter, der er defineret i biblioteket *BibNavnStreng*. *BibNavnStreng* skal være en streng (tekst omsluttet af citationstegn) eller en strengvariabel.

Hvis biblioteket *BibNavnStreng* ikke findes, opstår der en fejl.

Bemærk eksemplet til venstre, i hvilket resultatet af **getVarInfo()** er tilknyttet til variabel *vs*. Forsøg på at vise række 2 eller række 3 af *vs* returnerer en "ugyldig liste eller matrix" fejl, fordi mindst et af elementerne i disse rækker (variable *b*, f.eks) reevaluerer til en matrix.

Denne fejl kan også opstå, når *Ans* bruges til at evaluere et **getVarInfo()** resultat.

Systemet giver den ovenfor nævnte fejl, fordi den aktuelle version af softwaren ikke understøtter en generaliseret matrixstruktur, hvor et element i en matrix enten kan være en matrix eller en liste.

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

**Goto****Goto etiketnavn**

Overfører kontrol til etiketten *etiketnavn*.

*Etiketnavn* skal defineres i den samme funktion med en **Lbl**-kommando.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

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

*Udtr1* ► **Grad** ⇒ *udtryk*

I vinkeltilstanden Grader:

Konverterer *Udtr1* til vinkelmål i nygrader.

(1.5) ► Grad (1.66667)<sup>g</sup>

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @>Grad.

I vinkeltilstanden Radian:

(1.5) ► Grad (95.493)<sup>g</sup>

/

**identitet()**

**identitet**(*heltal*) ⇒ *matrix*

Returnerer identitetsmatrixen med en dimension af *heltal*.

identity(4) 

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

*Heltal* skal være et positivt heltal.

**If**

**If** *BooleanExpr*  
*Statement*

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

**If** *BooleanExpr* **Then**  
*Block*

**Endif**

$g(-2)$  4

Hvis *BooleanExpr* evalueres som sand, eksekveres enkelt sætningen *Statement* eller sætningsblokken *Block*, før eksekveringen fortsættes.

Hvis *BooleanExpr* evalueres som falsk, fortsættes eksekveringen uden eksekvering af sætningen eller sætningsblokken.

*Block* kan enten være en enkelt sætning eller en række sætninger adskilt af ":"-tegn

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

**If BooleanExpr Then***Block1***Else***Block2***EndIf**

Hvis *BooleanExpr* evalueres som sand, eksekveres *Block1*, og *Block2* springes over.

Hvis *BooleanExpr* evalueres som falsk, springes over *Block1*, men *Block2* eksekveres.

*Block1* og *Block2* kan være en enkelt sætning.

**If BooleanExpr1 Then***Block1***ElseIf BooleanExpr2 Then***Block2*

:

**ElseIf BooleanExprN Then***BlockN***EndIf**

Muliggør en forgrening. Hvis *BooleanExpr1* evalueres som sand, eksekveres *Block1*. Hvis *BooleanExpr1* evalueres som falsk, evalueres *BooleanExpr2*, og så videre.

Define  $g(x)=\text{Func}$ 

Done

If  $x < 0$  ThenReturn  $-x$ 

Else

Return  $x$ 

EndIf

EndFunc

 $g(12)$ 

12

 $g(-12)$ 

12

Define  $g(x)=\text{Func}$ If  $x < 5$  Then

Return 5

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

Return 3

EndIf

EndFunc

Done

 $g(-4)$ 

4

 $g(10)$ 

3

**ifFn()**

**ifFn( BooleanExpr, Value\_If\_true [, Value\_If\_false [, Value\_If\_unknown]])**  
 $\Rightarrow$  udtryk, liste eller matrix

Beregner det boolske udtryk *BooleanExpr* (eller hvert element i *BooleanExpr*) og giver et resultat baseret på følgende regler:

- *BooleanExpr* kan teste en enkelt værdi, en liste eller en matrix.
- Hvis et element i *BooleanExpr* evaluerer sandt, returneres det tilsvarende element fra *Value\_If\_true*.

ifFn({1,2,3}&lt;2,5,{5,6,7},{8,9,10})

{5,6,10}

Testværdien for **1** er mindre end 2,5, så dens tilsvarende

*Value\_If\_True*-element på 5 kopieres til resultatlisten.

Testværdien for **2** er mindre end 2,5, så dens tilsvarende

- Hvis et element i *BooleanExpr* evaluerer falsk, returneres det tilsvarende element fra *Value\_If\_false*. Hvis du udelader *Value\_If\_false*, returneres undef.
- Hvis et element i *BooleanExpr* hverken er sandt eller falsk, returneres det tilsvarende element *Value\_If\_unknown*. Hvis du udelader *Value\_If\_unknown*, returneres undef.
- Hvis det andet, tredje eller fjerde argument i **ifFn()**-funktionen er et enkelt udtryk, udføres den Boolske test på hver position i *BooleanExpr*.

**Bemærk:** Hvis den reducerede *BooleanExpr*-sætning indeholder en liste eller matrix, skal alle andre liste- eller matrixargumenter have de samme dimensioner, og resultatet vil have de samme dimensioner.

*Value\_If\_True*-element på 6 kopieres til resultatlisten.

Testværdien for 3 er ikke mindre end 2,5, så det tilhørende *Value\_If\_False*-element på 10 kopieres til resultatlisten.

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

*Value\_If\_True* er en enkelt værdi og svarer til enhver valgt position.

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

*Value\_If\_False* er ikke specificeret. Undef anvendes.

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

Ét element valgt fra *Value\_If\_True*. Ét element valgt fra *Value\_If\_unknown*.

## imag()

**imag(ValueI)** ⇒ værdi

Returnerer imaginærdelen af argumentet.

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

**imag(ListI)** ⇒ liste

Returnerer en liste med imaginærdelen af elementerne.

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

**imag(MatrixI)** ⇒ matrix

Returnerer en matrix med imaginærdelene af elementerne.

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

## Henvisning

Se #(), side 201.

## inString()

Katalog > 

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

<code>inString("Hello there", "the")</code>	7
<code>inString("ABCEFG", "D")</code>	0

Returnerer tegnpositionen i strengen *srcString*, hvor første forekomst af strengen *subString* begynder.

*Start*, hvis medtaget, angiver den position i *srcString*, hvor søgningen begynder. Standard = 1 (første tegn i *srcString*).

Hvis *srcString* ikke indeholder *subString*, eller *Start* er > længden på *srcString*, returneres nul.

## int()

Katalog > 

**int**(*Value*) ⇒ *heltal*

**int**(*List1*) ⇒ *liste*

**int**(*Matrix1*) ⇒ *matrix*

<code>int(-2.5)</code>	-3.
<code>int([-1.234 0 0.37])</code>	[-2. 0 0.]

Returnerer det største heltal, der er mindre end eller lig med argumentet. Denne funktion identisk med **floor()**.

Argumentet kan være et reelt eller komplekst tal.

For lister og matrixer returneres det største heltal mindre end eller lig med hvert element.

## intDiv()

Katalog > 

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

**intDiv**(*List1*, *List2*) ⇒ *liste*

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

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

Returnerer den heltalsdel med fortegn, der er en del af (*Number1* ÷ *Number2*).

Returnerer for lister og matrixer heltalsdelen med fortegn, der er en del af (argument 1 ÷ argument 2), for hvert elementpar.

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

Denne funktion gør følgende:

Givet *xList*, *yList*=**f**(*xList*) og *yPrimeList*=**f'**(*xList*) for en ukendt funktion **f** anvendes en kubisk interpolation til at approksimere funktionen **f** ved *xValue*. Det antages, at *xList* er en liste med monotont voksende eller aftagende tal, men denne funktion kan returnere en værdi selvom det ikke er tilfældet. Denne funktion gennemløber *xList* i søgningen efter et interval [*xList*[*i*], *xList*[*i*+1]] der indeholder *xValue*. Hvis den finder et sådan interval, returnerer den en interpoleret værdi for **f**(*xValue*); i modsat fald returnerer den **undef**.

*xList*, *yList* og *yPrimeList* skal have samme dimension  $\geq 2$  og indeholde udtryk, der reducerer til tal.

*xValue* kan være et tal eller en liste af tal.

Differentialligning:

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

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

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

Brug funktionen **interpolate**() til at beregne funktionsværdierne for listen med *x*-værdier:

$xvalueList := seq(i, i, 0, 10, 0.5)$
{ 0, 0.5, 1., 1.5, 2., 2.5, 3., 3.5, 4., 4.5, 5., 5.5, 6., 6.5, 7., 7.5, 8., 8.5, 9., 9.5, 10 }
$xlist := mat▶list(rK[1])$
{ 0., 1., 2., 3., 4., 5., 6., 7., 8., 9., 10. }
$ylist := mat▶list(rK[2])$
{ 5., 3.19499, 5.00394, 6.99957, 9.00593, 10.9970 }
$yprimeList := -3 \cdot y + 6 \cdot t + 5   y = ylist \text{ og } t = xlist$
{ -10., 1.41503, 1.98819, 2.00129, 1.98221, 2.00671 }
$interpolate(xvalueList, xlist, ylist, yprimeList)$
{ 5., 2.67062, 3.19499, 4.02782, 5.00394, 6.00011 }

## invχ<sup>2</sup>()

**invχ<sup>2</sup>**(*Areal*, *df*)

**invChi2**(*Areal*, *df*)

Beregner den inverse kumulerede  $\chi^2$  (chi-kvadrat) sandsynlighedsfunktion angivet ved frihedsgrad, *fg* for et givet *Areal* under kurven.

## invF()

**invF**(*Areal*, *dfNumer*, *dfDenom*)

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

Beregner den inverse kumulerede F fordelingsfunktion angivet ved *dfNumer* og *dfDenom* for et givet *Areal* under kurven.

## invBinom()

Katalog > 

### invBinom

(*CumulativeProb*, *NumTrials*, *Prob*, *OutputForm*) ⇒ skalar eller matrix

Givet antallet af forsøg (*NumTrials*) og sandsynligheden for succes i hvert forsøg (*Prob*) vil denne funktion returnere det minimale antal succeser, *k*, således at den kumulerede sandsynlighed for *k* succeser er større end eller lig med den givne kumulerede sandsynlighed (*CumulativeProb*).

*OutputForm*=0, viser resultatet som en skalar (standard).

*OutputForm*=1, viser resultatet som en matrix.

Eksempel: Mary og Kevin spiller et spil terninger. Mary skal gætte det maksimale antal gange, der rulles en sekser i 30 slag. Hvis der rulles en sekser så mange eller færre gange, vinder Mary. Desuden vinder hun mere, jo mindre det tal, hun gætter på, er. Hvad er det mindste tal, Mary kan gætte på, hvis hun vil have en sandsynlighed for at vinde på over 77 %?

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

## invBinomN()

Katalog > 

invBinomN(*CumulativeProb*, *Prob*, *NumSuccess*, *OutputForm*) ⇒ skalar eller matrix

Givet sandsynligheden for succes i hvert forsøg (*Prob*) og antallet af succeser (*NumSuccess*) vil denne funktion returnere det minimale antal forsøg, *N*, således at den kumulerede sandsynlighed for *x* succeser er mindre end eller lig med den givne kumulerede sandsynlighed (*CumulativeProb*).

*OutputForm*=0, viser resultatet som en skalar (standard).

*OutputForm*=1, viser resultatet som en matrix.

Eksempel: Monique øver sig i målskud i kurvebold. Hun ved fra erfaring, at hendes chance for at ramme målet ved et kast er 70 %. Hun har tænkt sig at øve, til hun har fået 50 mål. Hvor mange skud skal hun forsøge for at være sikker på, at sandsynligheden for at få mindst 50 mål er større end 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(*Areal*[ $\mu$ ,  $\sigma$ ])

Beregner den inverse kumulerede normalfordelingsfunktion for et givet *Areal* under normalfordelingskurven angivet ved  $\mu$  og  $\sigma$ .



**invf**(*Area*,*df*)

Beregner den inverse kumulerede student-t sandsynlighedsfunktion angivet ved frihedsgrad *df* for et givet *Area* under kurven.

**iPart**()**iPart**(*Number*) ⇒ *heltal***iPart**(*List1*) ⇒ *liste***iPart**(*Matrix1*) ⇒ *matrix*

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

Returnerer heltalsdelen af argumentet.

Returnerer heltalsdelen af hvert element for lister og matrixer.

Argumentet kan være et reelt eller komplekst tal.

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

Finansfunktion, der beregner den interne rente af en investering.

*CF0* er startpengestrømmen på tidspunkt 0; den skal være et reelt tal.

*CFList* er en liste over pengestrømsbeløb efter startpengestrømmen *CF0*.

*CFFreq* er en valgfri liste, hvor hvert element angiver hyppigheden for et grupperet (fortløbende) pengestrømsbeløb, som er det tilsvarende element i *CFList*. Standardværdien er 1; Hvis du indtaster værdier, skal de være positive heltal < 10.000.

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

**Bemærk:** Se også **mirr()**, side 99.

## isPrime()

Katalog > 

**isPrime**(*Number*) ⇒ *boolsk konstantudtryk*

Returnerer sand eller falsk for at vise, om *Number* er et helt tal  $\geq 2$ , der kun kan divideres med sig selv og 1.

Hvis *Number* har flere end ca. 306 cifre og ikke har nogen faktorer  $\leq 1021$ , viser **isPrime**(*Number*) en fejlmeddelelse.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

isPrime(5)	true
isPrime(6)	false

Funktion til søgning af det næste primtal efter det angivne tal:

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

## isVoid()

Katalog > 

**isVoid**(*Var*) ⇒ *boolsk konstantudtryk*  
**isVoid**(*Expr*) ⇒ *boolsk konstantudtryk*  
**isVoid**(*List*) ⇒ *liste over boolske konstantudtryk*

Returnerer sand eller falsk for at angive, om argumentet er en ugyldig datatype.

For yderligere oplysninger om ugyldige elementer se side 224.

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

## L

### Lbl

Katalog > 

**Lbl** *etiketNavn*

Definerer en etiket med navnet *etiketNavn* i en funktion.

Du kan anvende en **Goto** *etiketNavn*-kommando til at videregive kontrollen til kommandoen lige efter etiketten.

*EtiketNavn* skal opfylde de samme navngivningskrav som et variabelnavn.

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

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

**lcm()**

**lcm**(*Værdi1*, *Værdi2*) $\Rightarrow$ *udtryk*

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

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

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

**lcm**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

Returnerer det mindste fælles multiplum af to argumenter. **lcm** af to brøker er **lcm** af deres tællere divideret med **gcd** af deres nævnere. **lcm** af brøker med flydende komma er deres produkt.

For to lister eller matricer returneres det mindste fælles multiplum af deres tilsvarende elementer.

**left()**

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

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

Returnerer *Antal*-tegn fra venstre i tegnstrengen *kildeStreng*.

Hvis du udelader *Antal*, returneres alle *kildeStreng*-variable.

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

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

Returnerer *Antal*-elementer til venstre i *Liste1*.

Hvis du udelader *Antal*, returneres hele *Liste1*.

**left**(*Sammenligning*) $\Rightarrow$ *udtryk*

Returnerer venstre af en ligning eller ulighed.

**libShortcut**(*BibNavneStreng*,  
*GenvejNavneStreng*

[, *BibPrivFlag*]) ⇒ *liste med variable*

Opretter en variabelgruppe i den aktuelle opgave, som indeholder referencer til alle objekter i det specificerede biblioteksdokument *bibNavneStreng*. Tilføjer også gruppe-medlemmerne til variabelmenuen. Du kan henvise til hvert objekt ved brug af *GenvejNavneStreng*.

Sæt *BibPrivFlag* = **0** for at udelukke private biblioteksobjekter (standard)

Sæt *BibPrivFlag* = **1** for at medtage private biblioteksobjekter

For at kopiere en variabelgruppe, se **CopyVar** på side 25.

For at slette en variabelgruppe, se **DelVar** på side 40.

Dette eksempel forudsætter et korrekt gemt og opdateret biblioteksdokument med navnet, **linalg2**, som indeholder objekter defineret som *clearmat*, *gauss1*, og *gauss2*.

```
getVarInfo("linalg2")
┌──────────┬──────────┬──────────┬──────────┐
│ clearmat  │ "FUNC"   │ "LibPub"  │ "        │
│ gauss1    │ "PRGM"   │ "LibPriv" │ "        │
│ gauss2    │ "FUNC"   │ "LibPub"  │ "        │
└──────────┴──────────┴──────────┴──────────┘
```

---

```
libShortcut("linalg2", "la")
┌──────────┬──────────┬──────────┬──────────┐
│ la.clearmat, la.gauss2 │
└──────────┴──────────┴──────────┴──────────┘
```

---

```
libShortcut("linalg2", "la", 1)
┌──────────┬──────────┬──────────┬──────────┐
│ la.clearmat, la.gauss1, la.gauss2 │
└──────────┴──────────┴──────────┴──────────┘
```

## LinRegBx

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

Beregner den lineære regression  $y = a + b \cdot x$  på listerne *X* og *Y* med hyppigheder *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for  $X$  og  $Y$  data.

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a+b \cdot x$
stat.a, stat.b	Regressionskoefficienter
stat.r <sup>2</sup>	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede $X$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede $Y$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

**LinRegMx**  $X, Y, [Frekv][Kategori, Medtag]$

Beregner den lineære regression  $y = m \cdot x + b$  på listerne  $X$  og  $Y$  med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af undtagelse af *Medtag*.

$X$  og  $Y$  er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende  $X$  og  $Y$  datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Category* er en liste, der indeholder numeriske eller strengkategorikoder for  $X$  og  $Y$  data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $m \cdot x + b$
stat.m, stat.b	Regressionskoefficienter
stat.r <sup>2</sup>	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede $X$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede $Y$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

## LinRegtIntervaller

**LinRegtIntervals**  $X, Y, F[, 0[, CNiveau]]]$

Til hældning. Beregner et niveau  $C$  konfidensinterval for hældningen.

**LinRegtIntervals**  $X, Y, F[, 1, Xværdi [, CNiveau]]]$

Åbent svar Beregner en forudset  $y$ -værdi, et niveau C forudsigelsesinterval for enkle observationer, og et niveau C konfidensinterval til gennemsnits-responsen.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155)

Alle lister skal have samme dimension.

$X$  og  $Y$  er lister med uafhængige og afhængige variable.

$F$  er en valgfri liste med hyppighedsværdier. Hvert element i  $F$  angiver hyppigheden af forekomster for hvert tilsvarende datapunkt for  $X$  og  $Y$ . Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a+b \cdot x$
stat.a, stat.b	Regressionskoefficienter
stat.fg	Frihedsgrader
stat.r <sup>2</sup>	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen

Kun for hældningstypen

Output-variabel	Beskrivelse
[stat.CLower, stat.CUpper]	Konfidensinterval for hældningen
stat.ME	Konfidensinterval, fejlmargen
stat.SESlope	Standardfejll for hældning
stat.s	Standardfejll for linjen

Kun for svartype

Output-variabel	Beskrivelse
[stat.CLower, stat.CUpper]	Konfidensinterval for en middelværdi
stat.ME	Konfidensinterval, fejlmargen
stat.SE	Standardfejl for middelværdi
[stat.LowerPred, stat.UpperPred]	Prædiktionsinterval for en enkelt observation
stat.MEPred	Prædiktionsintervallsmargin for fejl
stat.SEPred	standardfejl for prædiktion
Statistik.ŷ	$a + b \cdot XVærdi$

## LinRegtTest

katalog > 

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

Beregner en lineær regression ud fra  $X$  og  $Y$  listerne og en  $t$  test på værdien af hældningen  $\beta$  og korrelationskoefficienten  $\rho$  for ligningen  $y = \alpha + \beta x$ . Den tester nulhypotesen  $H_0: \beta = 0$  (ækvivalent,  $\rho = 0$ ) mod en af tre alternative hypoteser.

Alle lister skal have samme dimension.

$X$  og  $Y$  er lister med uafhængige og afhængige variable.

$Frekv$  er en valgfri liste med hyppigheder. Hvert element i  $Frekv$  angiver hyppigheden af hændelse for hver tilsvarende  $X$  og  $Y$  datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

$Hypot$  er en valgfri værdi, som angiver en af tre alternative hypoteser, mod hvilken nulhypotesen ( $H_0: \beta = \rho = 0$ ) vil blive testet.

Til  $H_a: \beta \neq 0$  og  $\rho \neq 0$  (standard), sæt  $Hypot = 0$

Til  $H_a: \beta < 0$  og  $\rho < 0$ , sæt  $Hypot < 0$

Til  $H_a: \beta > 0$  og  $\rho > 0$ , sæt  $Hypot > 0$

En sammenfatning af resultaterne lagres i  $stat.results$  variable. (side 155)



Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a + b \cdot x$
stat.t	$t$ -Statistik for signifikanstest
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.fg	Frihedsgrader
stat.a, stat.b	Regressionskoefficienter
stat.s	Standardfejl for linjen
stat.SESlope	Standardfejl for hældning
stat.r <sup>2</sup>	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen

## linSolve()

**linSolve**(SystemafLineæreLigninger, Var1, Var2, ...) ⇒ liste

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

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

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

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

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

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

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

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

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

Returnerer en løsning for variablene Var1, Var2, ...

Det første argument skal kunne beregnes til et system af lineære ligninger eller en enkelt lineær ligning. Eller opstår der en argumentfejl.

For eksempel giver beregningen af **linSolve(x=1 og x=2,x)** resultatet "Argumentfejl".

**Δlist()**

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

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

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **deltaList (...)**.

Returnerer en liste med differenserne mellem konsekutive elementer i *Liste1*. Hvert element i *Liste1* er subtraheret fra det næste element i *Liste1*. Den resulterende liste er altid et element kortere end den oprindelige *Liste1*.

**list▶mat()**

$\text{list}\blacktriangleright\text{mat}(\text{Liste [, elementerPrRække]}) \Rightarrow \text{matrix}$

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

Returnerer en matrix fyldt rækkevis med elementerne fra *Liste*.

*ElementerPrRække* angiver antallet af elementer pr. række, hvis den er medtaget. Standard er antallet af elementer i *Liste* (en række).

Hvis *Liste* ikke udfylder den resulterende matrix, tilføjes nuller.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **list@>mat (...)**.

**ln()**

$\text{ln}(\text{Værdi}) \Rightarrow \text{værdi}$

$\text{ln}(2.)$	0.693147
-----------------	----------

**ln(Liste1)⇒liste**

Returnerer den naturlige logaritme til argumentet.

Til en liste returneres de naturlige logaritmer af elementerne.

Hvis kompleks formattilstand er reel:

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

"Error: Non-real calculation"

Hvis kompleks formattilstand er rektangulær:

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

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

**ln(kvadratMatrix1)⇒kvadratMatrix**

Returnerer den naturlige matrixlogaritme af *kvadratMatrix1*. Dette er ikke det samme som at beregne den naturlige logaritme af hvert element. Oplysninger om beregningsmetoden findes under **cos()**.

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

I vinkeltilstanden radian og rektangulært komplekst format:

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

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

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

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

Beregner den lineære regression  $y = a + b \cdot \ln(x)$  på liste *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for  $X$  og  $Y$  data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer, hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a+b \cdot \ln(x)$
stat.a, stat.b	Regressionskoefficienter
stat.r <sup>2</sup>	Koefficient af en lineær forklaringsgrad til transformerede data
stat.r	Korrelationskoefficient til transformerede data ( $\ln(x)$ , $y$ )
stat.Resid	Residualer forbundet med eksponentielmodellen
stat.ResidTrans	Residualer associeret med lineær tilpasning af transformerede data
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

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

Erklærer de angivne *var* som lokale variable. Disse variable eksisterer kun under beregning af en funktion og slettes, når eksekveringen af funktionen afsluttes.

**Bemærk:** Lokale variable sparer hukommelse, fordi kun eksisterer midlertidigt. De forstyrrer heller ikke de eksisterende globale variabelværdier. Lokale variable skal anvendes til **For**-løkker og midlertidig lagring af variabelværdier i en flerlinjefunktion da modifikationer af globale ikke er tilladt i en funktion.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
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

```

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

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

Låser de angivne variable eller den angivne variabelgruppe. Låste variable kan ikke redigeres eller slettes.

Du kan ikke låse eller oplåse systemvariablen *Ans*, og du kan ikke låse systemvariabelgrupperne *stat.* eller *tvm.*

**Bemærk:** Kommandoen **Lås (Lock)** rydder Fortryd/Annuller Fortryd-historikken, når den anvendes på ulåste variable.

Se **unLock**, side 174 og **getLockInfo()**, side 67.

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

## log()

ctrl 10<sup>x</sup>-taster

log(*Værdi1* [, *Værdi2*]) ⇒ *værdi*

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

log(*Liste1* [, *Værdi2*]) ⇒ *list*

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

Returner -*Udtr2*-talslogaritmen til argumentet.

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

**Bemærk:** Se også **Log-skabelon**, side 2.

Ved en liste returneres *Udtr2*-talslogaritmen til elementerne.

Hvis kompleks formattilstand er reel:

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

"Error: Non-real calculation"

Hvis *Udtr2* udelades, anvendes 10-talslogaritmen.

Hvis kompleks formattilstand er rektangulær:

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

log(*kvadratMatrix1* [, *Værdi2*]) ⇒ *kvadratMatrix*

Returnerer matrix *Værdi2*-logaritmen til *kvadratMatrix1*. Dette er ikke det samme som at beregne *Værdi2*-talslogaritmen til hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

I vinkeltilstanden radian og rektangulært komplekst format:

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

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

Hvis tal-argumentet udelades, anvendes 10-talslogaritmen.

Du kan se hele resultatet ved at trykke på **▲** og derefter bruge **◀** og **▶** til at bevæge markøren.

## Logistic

Katalog > 

Logistik *X*, *Y* [, [*Frekv*] [, *Kategori*, *Medtag*]]

Beregner den logistiske regression  $y = (c / (1 + a \cdot e^{-bx}))$  på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

$X$  og  $Y$  er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende  $X$  og  $Y$  datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for  $X$  og  $Y$  data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $c/(1+a \cdot e^{-bx})$
stat.a, stat.b, stat.c	Regressionskoefficienter
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede $X$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede $Y$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

**LogisticD**  $X, Y$  [, [*Iterationer*], [*Frekv*] [, *Kategori*, *Medtag*] ]

Beregner den logistiske regression  $y = (c / (1 + a \cdot e^{-bx}) + d)$  på listerne  $X$  og  $Y$  med hyppighed *Frekv*, ved brug af et angivet tal fra *Iterationer*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

$X$  og  $Y$  er lister med uafhængige og afhængige variable.

*Iterationer* er en valgfri værdi, som angiver det maksimale antal gange en løsning vil forsøges. Hvis udeladt, anvendes 64. Typisk resulterer større værdier i større nøjagtighed men længere eksekveringstider og omvendt.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende  $X$  og  $Y$  datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for  $X$  og  $Y$  data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $c / (1 + a \cdot e^{-bx}) + d$
stat.a, stat.b, stat.c, stat.d	Regressionskoefficienter
stat.Resid	Residualer fra regressionen



Output-variabel	Beskrivelse
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>Kategori liste</i> og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>Kategori liste</i> og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

## Loop

Katalog > 

### Loop

*Blok*

### EndLoop

Eksekverer gentagne gange sætningerne i *Blok*. Bemærk, at løkken eksekveres uendeligt, medmindre en **Goto** eller **Exit**-kommando eksekveres i *Blok*.

*Blok* er en sekvens af sætninger adskilt med kolon.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
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
```



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

**LU** *Matrix*, *lMatrix*, *uMatrix*, *pMatrix* [*Tol*]

Beregner Doolittle LU (nedre-øvre) opløsningen af en reel eller kompleks matrix. Den nedre triangulære matrix lagres i *lMatrix*, den øvre triangulære matrix i *uMatrix*, og permutationsmatricen (der beskriver de foretagne rækkeombbytninger under beregningen) i *pMatrix*.

$$lMatrix \cdot uMatrix = pMatrix \cdot matrix$$

Ethvert matricelement kan valgfrit behandles som nul, hvis dets absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers ignoreres *Tol*.

- Hvis du anvender   eller indstiller **Auto eller tilnærmet**-tilstanden til Approximate, foretages beregningerne med aritmetik med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:
- $5E-14 \cdot \max(\dim(Matrix)) \cdot \text{rækkeNorm}(Matrix)$

**LU**-algoritmen til faktoropløsning anvender partiel pivotering med rækkeombbytninger.

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

## M

### mat▶list()

**mat▶list**(*Matrix*) ⇒ *liste*

Returner en liste bestående af elementerne i *Matrix*.

Elementerne kopieres fra *Matrix* række for række.

<b>mat▶list</b> ( $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ )	$\{1,2,3\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
<b>mat▶list</b> ( <i>m1</i> )	$\{1,2,3,4,5,6\}$

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive `mat@>list(...)`.

## max()

**max**(*Værdi1*, *Værdi2*) $\Rightarrow$ udtryk

$\max\{2.3, 1.4\}$	2.3
--------------------	-----

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

$\max\{\{1,2\}, \{-4,3\}\}$	$\{1,3\}$
-----------------------------	-----------

**max**(*Matrix1*, *Matrix2*) $\Rightarrow$ matrix

Returnerer maksimum af de to argumenter. Hvis argumenterne er to lister eller matricer, returneres en liste eller matrix med maksimumsværdier for hvert sammenhørende elementpar.

**max**(*List*) $\Rightarrow$ udtryk

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

Returnerer det største element i *liste*.

**max**(*Matrix1*) $\Rightarrow$ matrix

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

Returnerer en rækkevektor med det største element i hver kolonne i *Matrix1*.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224

**Bemærk:** Se også `min()`.

## mean()

**mean**(*Liste*[, *hyppighedsliste*]) $\Rightarrow$ udtryk

$\text{mean}\{\{0.2,0.1,-0.3,0.4\}\}$	0.26
---------------------------------------	------

Returnerer middelværdien for elementerne i *Liste*.

$\text{mean}\{\{1,2,3\}, \{3,2,1\}\}$	$\frac{5}{3}$
---------------------------------------	---------------

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

**mean**(*Matrix1*[, *Hyppighedsmatrix*]) $\Rightarrow$ matrix

I rektangulært vektorformat:

Returnerer en rækkevektor af middelværdierne af alle kolonner i *Matrix1*.

## mean()

Katalog > 

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

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

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

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

$$\text{mean}\left(\begin{array}{cc|cc} 1 & 2 & 5 & 3 \\ 3 & 4 & 4 & 1 \\ 5 & 6 & 6 & 2 \end{array}\right) \quad \left[\frac{47}{15} \quad \frac{11}{3}\right]$$

## median()

Katalog > 

**median**(*Liste*[,  
*Hyppighedsliste*]) $\Rightarrow$ *udtryk*

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

Returnerer medianen af elementerne i *Liste*.

Hvert *Hyppighedsliste*-element tæller antallet af forekomster i ubrudt rækkefølge for de tilsvarende elementer i *Liste*.

**median**(*Matrix1*[,  
*Hyppighedsmatrix*]) $\Rightarrow$ *matrix*

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

Returnerer en rækkevektor med medianerne af kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af forekomster i en ubrudt rækkefølge af det tilsvarende element i *Matrix1*.

### Noter:

- Alle elementer i listen eller matricen skal kunne omregnes til tal.
- tomme (ugyldige) elementer i listen eller matricen ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

**MedMed**  $X, Y$  [, *Frekv*] [, *Kategori*, *Medtag*]

Beregner median-median linje  $y = (m \cdot x + b)$  på listerne  $X$  og  $Y$  med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

$X$  og  $Y$  er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende  $X$  og  $Y$  datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for  $X$  og  $Y$  data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer, hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Median-median-linjeligning: $m \cdot x + b$
stat.m, stat.b	Modelkoefficienter
stat.Resid	Residualer fra median-median-linjen
stat.XReg	Liste af datapunkter i den modificerede $X$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategori</i>
stat.YReg	Liste af datapunkter i den modificerede $Y$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategori</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

**mid()**Katalog > **mid(kildeStreng, Start[, Antal])**⇒streng

Returnerer *Antal* tegn fra tegnstrengen *kildeStreng*, startende med tegn nummer *Start*.

Hvis *Antal* udelades eller er større end dimension på *kildeStreng*, returneres alle tegn fra *kildeStreng*, begyndende med tegn nummer *Start*.

*Antal* skal være  $\geq 0$ . Hvis *Antal* = 0, returneres en tom streng.

**mid(kildeListe, Start [, Antal])**⇒liste

Returnerer *Antal* elementer fra *kildeListe*, begyndende med element nummer *Start*.

Hvis *Antal* udelades eller er større end dimensionen på *kildeListe*, returneres alle elementer fra *kildeListe*, begyndende med element nummer *Start*.

*Antal* skal være  $\geq 0$ . Hvis antal = 0, returneres en tom liste.

**mid(kildeStrengListe, Start[, Antal])**⇒liste

Returnerer *Antal* strenge fra listen med strenge *kildeStrengListe* begyndende med element nummer *Start*.

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

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

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

**min()**Katalog > **min(Værdi1, Værdi2)**⇒udtryk**min(Liste1, Liste2)**⇒liste**min(Matrix1, Matrix2)**⇒matrix

Returnerer minimum af de to argumenter. Hvis argumenterne er to lister eller matricer, returneres en liste eller matrix med minimumværdi af hvert sammenhørende elementpar.

**min(Liste)**⇒udtryk

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

min({0,1,-7,1,3,0,5})	-7
-----------------------	----

Returnerer det mindste element af *Liste*.

$\text{min}(\text{Matrix1}) \Rightarrow \text{matrix}$

Returnerer en rækkevektor med det mindste element i hver kolonne i *Matrix1*.

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

**Bemærk:** Se også `max()`.

**mirr**

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

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

Finansfunktion, der returnerer den modificerede interne rente af en investering.

*finansRente* er rentesatsen, du betaler for pengestrømsbeløbene.

*geninvestRente* er rentesatsen, som pengestrømmen geninvesteres til.

*CF0* er startpengestrømmen på tidspunkt 0. Den skal være et reelt tal.

*CFListe* er en liste over pengestrømsbeløb efter startpengestrømmen *CF0*.

*CFFrekv* er en valgfri liste, hvor hvert element angiver hyppigheden for et grupperet (fortløbende) pengestrømsbeløb, som er det tilsvarende element i *CFListe*. Standardværdien er 1. Hvis du indtaster værdier, skal de være positive heltal < 10.000.

**Bemærk:** Se også `irr()`, side 77.

**mod()**

Katalog &gt;

**mod**(*Værdi1*, *Værdi2*) $\Rightarrow$ *udtryk*

mod(7,0) 7

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

mod(7,3) 1

**mod**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

mod(-7,3) 2

Returnerer det første argument modulo andet argument som defineret efter definitionen:

mod(7,-3) -2

mod(-7,-3) -1

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

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

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

Når det andet argument er ikke-nul, er resultatet periodisk i det pågældende argument. Resultatet er enten nul eller har samme fortegn som det andet argument.

Hvis argumenterne er to lister eller to matricer, returneres en liste eller matrix med modulo af hvert par af sammenhørende elementer.

**Bemærk:** Se også **remain()**, side 132

**mRow()**

Katalog &gt;

**mRow**(*Tal*, *Matrix1*, *Index*) $\Rightarrow$ *matrix*

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks* af *Matrix1* ganget med *Tal*.

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

Katalog &gt;

**mRowAdd**(*Tal*, *Matrix1*, *Indeks1*, *Indeks2*) $\Rightarrow$ *matrix*

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks2* af *Matrix1* erstattet med:

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

$$\text{Tal} \times \text{række Indeks1} + \text{række Indeks2}$$

**MultReg**

Katalog &gt;

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



Beregner multiple lineære regressioner af listen  $Y$  på listerne  $X1, X2, \dots, X10$ . En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.b0, stat.b1, ...	Regressionskoefficienter
stat.r <sup>2</sup>	Koefficient af multipel forklaringsgrad
stati. $\hat{y}$ List	$\hat{y}$ Liste = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualer fra regressionen

## MultRegIntervals

**MultRegIntervals**  $Y, X1[,X2[,X3, \dots [,X10]]], XValListe[,CNiveau]$

Beregner en forudset  $y$ -værdi, et niveau  $C$  forudsigelsesinterval for enkle observationer, og et niveau  $C$  konfidensinterval til gennemsnits-responsen.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
Statistik. $\hat{y}$	Et punkttestimat: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ for <i>XValListe</i>
stat.dffejl (stat.dfeError)	Frihedsgrader for fejl
stat.CLower, stat.CUpper	Konfidensinterval for en middelværdi

Output-variabel	Beskrivelse
stat.ME	Konfidensinterval, fejlmargen
stat.SE	Standardfejl for middelværdi
stat.LowerPred, stat.UpperPred	Prædiktionsinterval for en enkelt observation
stat.MEPred	Prædiktionsintervallmargin for fejl
stat.SEPred	Standardfejl for prædiktion
stat.bList	Liste med regressionskoefficienter, {b0,b1,b2,...}
stat.Resid	Residualer fra regressionen

## MultRegTests

Katalog > 

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

Multipel lineær regressionstest beregner en multipel lineær regression fra de givne data, og danner den globale  $F$  teststatistik og  $t$  teststatistikker for koefficienterne.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)



Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $b_0+b_1 \cdot x_1+b_2 \cdot x_2+ \dots$
stat.F	Global $F$ teststatistik
stat.PVal	P-værdi tilknyttet global $F$ statistik
stat.r <sup>2</sup>	Koefficient af multipel forklaringsgrad
stat.AdjR <sup>2</sup>	Justeret koefficient af multipel forklaringsgrad
stat.s	Standardafvigelse for fejlen
stat.DW	Durbin-Watson-statistik; Anvendes til at bestemme, om første-ordens auto-korrelationen er tilstede i modellen
stat.dfReg	Frihedsgrader i regressionen

Output-variabel	Beskrivelse
stat.SSReg	Kvadraternes regressionsum
stat.MSReg	Middelkvadrat af regression
stat.dfFejl	Frihedsgrader for fejl
stat.SSError	fejl, kvadratsum
stat.MSError	fejl, middelkvadrat
stat.bList	{b0,b1,...} Liste med koefficienter
stat.tList	Liste med t statistikker for hver koefficient i bListen
stat.PList	Liste P-værdier for hver t-statistik
stat.SEList	Liste med standardfejl for koefficienter i bListe
stat.ŷList	$\hat{y}$ Liste = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualer fra regressionen
stat.sResid	Standardiserede residualer; Værdi opnået ved at dividere en residual med dens standardafvigelse
stat.CookDist	Cooks distance; Mål for påvirkningen af en observation baseret på residual og udnyttelse
stat.Leverage	Mål for, hvor langt værdierne for de uafhængige variable er fra deres middelværdier

## nand

  -taster

*BoolskUdtryk1* **nand** *BoolskUdtryk2*  
returnerer *boolsk udtryk*

$x \geq 3$ and $x \geq 4$	$x \geq 4$
---------------------------	------------

*BoolskListe1* **nand** *BoolskListe2*  
returnerer *Boolsk liste*

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

*BoolskMatrix1* **nand** *BoolskMatrix2*  
returnerer *Boolsk matrix*

Returnerer negationen af en logisk **and** operation anvendt på de to argumenter. Returnerer true eller false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdierne element for element.

*Heltal1* nand *Heltal2*  $\Rightarrow$  *heltal*

Sammenligner to reelle heltal bit for bit med en **nand** operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarende bits sammenlignes, er resultatet 1, hvis begge bits er 1. Ellers er resultatet 0. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemtilstand.

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimalt indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talsystem).

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}

## N

### nCr()

Katalog > 

$nCr(Værdi1, Værdi2) \Rightarrow$  udtryk

$nCr(z,3); z=5$  10

For heltal  $Værdi1$  og  $Værdi2$  med  $Værdi1 \geq Værdi2 \geq 0$ ,  $nCr()$  er antallet af kombinationer af  $Værdi1$  ting taget  $Værdi2$  ad gangen. (Dette kendes også som en binomial koefficient).

$nCr(z,3); z=6$  20

$nCr(Tal, 0) \Rightarrow 1$

$nCr(Tal, negHeltal) \Rightarrow 0$

$nCr(Tal, posHeltal) \Rightarrow Tal \cdot (Tal-1) \dots$

$(Tal-posHeltal+1)/posHeltal!$

$nCr(Tal, ikkeHeltal) \Rightarrow$  udtryk!  
 $((Tal-ikkeHeltal)! \cdot ikkeHeltal!)$

$nCr(Liste1, Liste2) \Rightarrow$  liste

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

Returnerer en liste med kombinationer baseret på de sammenhørende elementpar i de to lister. Argumenterne skal være lister af samme størrelse.

$nCr(Matrix1, Matrix2) \Rightarrow matrix$

Returnerer en matrix af kombinationer baseret på de sammenhørende elementpar i de to matrixer. Argumenterne skal være matrixer af samme størrelse.

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

## nDerivative()

$nDerivative(Udtr1, Var=Værdi [, Orden]) \Rightarrow værdi$

$nDerivative(Udtr1, Var[, Orden]) | Var=Værdi \Rightarrow værdi$

Returnerer den numeriske differentialkvotient udregnet med en automatisk differentiationsmetode.

Når *Værdi* er angivet, tilsidesætter den alle forudgående variabeltildelinger og alle nuværende “|” substitutioner for variabelen.

Hvis variabelen *Var* ikke indeholder en numerisk værdi, skal du angive *Værdi*.

*Differentialkvotientens orden* skal være **1** eller **2**.

**Bemærk:** Algoritmen **nDerivative()** har en begrænsning: den arbejder sig rekursivt gennem det uforkortede udtryk og beregner den numeriske værdi af differentialkvotienten af første orden (og anden, hvis relevant) af hvert deludtryk, hvilket kan føre til et utilsigtet resultat.

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

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

## nDerivative()

Katalog > 

Betragt eksemplet til højre. Differentialkvotienten af første orden af  $x \cdot (x^2 + x)^{1/3}$  i  $x=0$  er lig med 0. Men da differentialkvotienten af underudtrykket  $(x^2 + x)^{1/3}$  ikke er defineret i  $x=0$ , og denne værdi anvendes til at beregne differentialkvotienten af det samlede udtryk, rapporterer **nDerivative()** resultatet som udefineret og viser en advarsel.

Hvis du rammer denne begrænsning, skal du kontrollere løsningen grafisk. Du kan også prøve med **centralDiff()**.

## newList()

Katalog > 

**newList(antalElementer)** ⇒ *liste*

`newList(4)`  $\{0,0,0,0\}$

Returnerer en liste med en dimension af *antalElementer*. Hvert element er nul.

## newMat()

Katalog > 

**newMat(antalRækker, antalKolonner)** ⇒ *matrix*

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

Returnerer en matrix med nulpunkter af dimensionen *antalRækker* gange *antalKolonner*.

## nfMax()

Katalog > 

**nfMax(Udtr, Var)** ⇒ *værdi*

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

**nfMax(Udtr, Var, nedreGrænse)** ⇒ *værdi*

`nfMax( $0.5 \cdot x^3 - x - 2, x, -5, 5$ )` 5.

**nfMax(Udtr, Var, nedreGrænse, øvreGrænse)** ⇒ *værdi*

**nfMax(Udtr, Var) | nedreGrænse ≤ Var ≤ øvreGrænse** ⇒ *værdi*

Returnerer mulig numerisk værdi for variabelen *Var*, hvor det lokale maksimum for *Udtr* optræder.

Hvis du opgiver *nedreGrænse* og *øvreGrænse*, søger funktionen i det lukkede interval [*nedreGrænse*,*øvreGrænse*] efter det lokale maksimum.

**nfMin**(*Udtr*, *Var*) $\Rightarrow$ værdi

$$\text{nfMin}(x^2+2\cdot x+5,x) \quad -1.$$

**nfMin**(*Udtr*, *Var*, *nedreGrænse*) $\Rightarrow$ værdi

$$\text{nfMin}(0.5\cdot x^3-x-2,x,-5,5) \quad -5.$$

**nfMin**(*Udtr*, *Var*, *nedreGrænse*, *øvreGrænse*) $\Rightarrow$ værdi

**nfMin**(*Udtr*, *Var*) | *nedreGrænse* $\leq$ *Var*  
 $\leq$ *øvreGrænse* $\Rightarrow$ værdi

Returnerer mulig numerisk værdi for variabelen *Var*, hvor det lokale minimum for *Udtr* optræder.

Hvis du opgiver *nedreGrænse* og *øvreGrænse*, søger funktionen i det lukkede interval [*nedreGrænse*,*øvreGrænse*] efter det lokale minimum.

**nInt**(*Udtr1*, *Var*, *nedre*, *øvre*) $\Rightarrow$ udtryk

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

Hvis integranden *Udtr1* ikke indeholder andre variable end *Var*, og hvis *Nedre* og *Øvre* er konstante,  $+\infty$  eller  $-\infty$ , så returnerer **nInt()** en tilnærmet værdi af  $\int$  (*Udtr1*, *Var*, *Nedre*, *Øvre*). Denne tilnærmede værdi er et vægtet gennemsnit af nogle eksempelverdier af integranden i intervallet *Nedre*<*Var*<*Øvre*.

Målet er seks betydende cifre. Algoritmen, der kan tilpasses, afsluttes hvis det virker sandsynligt, at målet er nået, eller når det virker usandsynligt, at ydeligere eksempler vil give en væsentlig forbedring.

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

## nInt()

Katalog > 

Der vises en advarsel ("Tvivl om nøjagtighed") når målet ikke ser ud til at være nået.

Indskyd flere **nInt()**, for at foretage numerisk integration i flere variable. Integrationsgrænser kan afhænge af integrationsvariable uden for 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*) ⇒ værdi



$$\text{nom}(5.90398, 12) \quad 5.75$$

Finansfunktion, der omregner den effektive årlige rente *effektivRente* til en nominel rente, hvor *CpY* er antallet af rentetilskrivninger per år.

*effektivRente* skal være et reelt tal, og *CpY* skal være et reelt tal > 0.

**Bemærk:** Se også **eff()**, side 46.

## nor

  -taster

*BoolskUdtryk1* **nor** *BoolskUdtryk2*  
returnerer *boolsk udtryk*

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

*BoolskMatrix1* **nor** *BoolskMatrix2*  
returnerer *Boolsk matrix*

Returnerer negationen af en logisk **or** operation anvendt på de to argumenter. Returnerer true eller false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdierne element for element.



*Heltal1* nor *Heltal2* ⇒ *heltal*

Sammenligner to reelle heltal bit for bit med en **nor** operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarende bits sammenlignes, er resultatet 0, hvis begge bits er 1. Ellers er resultatet 1. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemtilstand.

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimalle indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talsystem)

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**(*Matrix*) ⇒ *udtryk*

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

**norm**(*Vektor*) ⇒ *udtryk*

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

Returnerer Frobenius-normen.

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

## normCdf()

Katalog > 

**normCdf**(*nedreGrænse*, *øvreGrænse*,  $[\mu, \sigma]$ ) ⇒ *tal* hvis *nedreGrænse* og *øvreGrænse* er tal, *liste* hvis *nedreGrænse* og *øvreGrænse* er lister

Beregner normalfordelingens sandsynlighed mellem *nedreGrænse* og *øvreGrænse* for de angivne  $\mu$  (standard=0) og  $\sigma$  (standard=1).

For  $P(X \leq \textit{øvreGrænse})$ , sæt *nedreGrænse* = -9E999.

**normPdf**( $XV\text{æ}rdi[\mu, \sigma]$ )  $\Rightarrow$  tal hvis  $XV\text{æ}rdi$  er et tal, *liste* hvis  $XV\text{æ}rdi$  er en liste

Beregner tæthedsfunktionen for normalfordelingen i en angivet  $XV\text{æ}rdi$  for de angivne  $\mu$  og  $\sigma$ .

**not**

**not** *Boolsk udtr1*  $\Rightarrow$  *Boolsk udtryk*

Returnerer true eller false eller en forenklet form af argumentet.

**not** *Heltal1*  $\Rightarrow$  *heltal*

Returnerer 1's komplement til et reelt heltal. Internt konverteres *Heltal1* til et 64-bit binært tal med fortegn. Værdien af hver bit vendes (0 bliver 1, og omvendt) for 1's komplement. Resultatet vises i den valgte tilstand for talsystem.

Du kan indtaste heltallet i ethvert talsystem. Til binære eller hexadecimal indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallet som decimaltal (10-talssystem).

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Yderligere oplysninger findes under **Base2**, side 17.

<code>not (2≥3)</code>	true
<code>not 0hB0</code> ▶ <code>Base16</code>	<code>0hFFFFFFFFFFFFFFF4F</code>
<code>not not 2</code>	2

I hexadecimal tilstand:

**Vigtigt:** Tallet nul, ikke bogstavet 0.

<code>not 0h7AC36</code>	<code>0hFFFFFFFFFFFF853C9</code>
--------------------------	----------------------------------

I binær tilstand:

<code>0b100101</code> ▶ <code>Base10</code>	37
<code>not 0b100101</code>	
<code>0b11111111111111111111111111111111</code> ▶	
<code>not 0b100101</code> ▶ <code>Base10</code>	-38

Du kan se hele resultatet ved at trykke på  $\blacktriangleleft$  og derefter bruge  $\blacktriangleleft$  og  $\blacktriangleright$  til at bevæge markøren.

**Bemærk:** En binær indtastning kan have op til 64 cifre (præfikset 0b ikke medregnet). En hexadecimal indtastning kan have op til 16 cifre.

**nPr()**

**nPr**( $V\text{æ}rdi1, V\text{æ}rdi2$ )  $\Rightarrow$  *udtryk*

For heltal er  $V\text{æ}rdi1$  og  $V\text{æ}rdi2$  med  $V\text{æ}rdi1 \geq V\text{æ}rdi2 \geq 0$ , **nPr**() antallet af permutationer af  $V\text{æ}rdi1$  ting taget  $V\text{æ}rdi2$  ad gangen.

**nPr**( $Tal, 0$ )  $\Rightarrow$  1

<code>nPr(z,3)</code> ; $z=5$	60
<code>nPr(z,3)</code> ; $z=6$	120
<code>nPr</code> { {5,4,3}, {2,4,2} }	{20,24,6}
<code>nPr</code> $\left[ \begin{matrix} 6 & 5 \\ 4 & 3 \end{matrix} \right]_{ \begin{matrix} 2 & 2 \\ 2 & 2 \end{matrix} }$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$

$nPr(Tal, negHeltal) \Rightarrow 1/((Tal+1) \cdot (Tal+2) \dots (Tal-negHeltal))$

$nPr(Tal, posHeltal) \Rightarrow Tal \cdot (Tal-1) \dots (Tal-posHeltal+1)$

$nPr(Tal, ikke-Heltal) \Rightarrow Tal! / (Tal-ikke-Heltal)!$

$nPr(Liste1, Liste2) \Rightarrow liste$

Returnerer en liste med permutationer baseret på de sammenhørende elementpar i de to lister. Argumenterne skal være lister af samme størrelse.

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

$nPr(Matrix1, Matrix2) \Rightarrow matrix$

Returnerer en matrix med permutationer baseret på de sammenhørende elementpar i de to matrixer. Argumenterne skal være matrixer af samme størrelse.

$nPr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$
--	---

$npv(RenteSats, CFO, CFListe [, CFFrekv])$

Finansfunktion, der beregner nettonutidsværdien. Summen af de aktuelle værdier for indkommende og udgående pengestrømme. Et positivt resultat for npv indikerer en profitabel investering.

*RenteSats* er renten, som pengestrømmen skal reduceres med over en periode (pengenes pris).

*CFO* er startpengestrømmen på tidspunkt 0. Den skal være et reelt tal.

*CFListe* er en liste over pengestrømsbeløb efter startpengestrømmen *CFO*.

$list1 := \{6000, -8000, 2000, -3000\}$	$\{6000, -8000, 2000, -3000\}$
$list2 := \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$npv(10, 5000, list1, list2)$	4769.91

*CFFrekv* er en liste, hvor hvert element angiver hyppighedsfrekvensen for et grupperet (fortløbende) pengestrømsbeløb, som er det tilsvarende element i *CFListe*. Standardværdien er 1. Hvis du indtaster værdier, skal de være positive heltal < 10.000.

## nSolve()

**nSolve**(*Ligning*,*Var*[=*Gæt*]) $\Rightarrow$ *tal* eller *fejlstreng*

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

**nSolve**(*Ligning*,*Var* [*=Gæt*],*nedreGrænse*) $\Rightarrow$ *tal* eller *fejlstreng*

$\text{nSolve}(x^2=4,x=-1)$	-2.
-----------------------------	-----

$\text{nSolve}(x^2=4,x=1)$	2.
----------------------------	----

**Bemærk:** Hvis der er flere løsninger, kan du anvende et gæt til at finde en partikulær løsning.

**nSolve**(*Ligning*,*Var* [*=Guess*],*nedreGrænse*,*øvreGrænse*) $\Rightarrow$ *tal* eller *fejlstreng*

**nSolve**(*Ligning*,*Var*[=*Guess*]) | *nedreGrænse* $\leq$ *Var* $\leq$ *øvreGrænse*  $\Rightarrow$  *tal* eller *fejlstreng*

Søger iterativt efter en approksimeret reel numerisk løsning af *Ligning* for dens ene variabel. Angiv variabelen som:

*variabel*

– eller –

*variabel* = *reelt tal*

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

**nSolve**() forsøger at bestemme enten et punkt, hvor residualen er nul, eller to forholdsvis tætte punkter, hvor residualerne har modsatte fortegn, og residualen ikke er for stor. Hvis dette ikke kan opnås med et beskedent antal datapunkter, returneres strengen "Ingen løsning blev fundet."

$\text{nSolve}(x^2+5\cdot x-25=9,x) x<0$	-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** [1,] $X$ ],[*Hyppeghed*]  
[,*Kategori*,*Medtag*]

**OneVar** [ $n$ ,] $X1$ , $X2$ [ $X3$ [...[, $X20$ ]]]

Beregner statistik med en variabel på op til 20 lister. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

$X$  argumenterne er datalister.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende  $X$  værdi. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste med numeriske kategorikoder for tilsvarende  $X$  værdier.

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Et tomt (ugyldigt) element i en af listerne  $X$ , *Freq* eller *Category* resulterer i at tilsvarende element i alle disse lister bliver ugyldigt. Et tomt element i en af listerne  $X1$  til  $X20$  resulterer i at tilsvarende element i alle disse lister bliver ugyldigt. Yderligere oplysninger om tomme elementer findes på side 224

Output-variabel	Beskrivelse
stat. $\bar{x}$	Gennemsnit af $x$ -værdier
stat. $\Sigma x$	Summen af $x$ -værdier
stat. $\Sigma x^2$	Summen af $x^2$ værdier
stat.sx	Standardafvigelse for målingen for $x$
stat. x	Populations standardafvigelse for $x$

Output-variabel	Beskrivelse
stat.n	Antal datapunkter
stat.MinX	Minimum af x-værdier
stat.Q <sub>1</sub> X	1. kvartil af x
stat.MedianX	Median af x.
stat.Q <sub>3</sub> X	3. kvartil af x.
stat.MaxX	Maksimum af x-værdier.
stat.SSX	Summen af kvadraterne på afvigelser fra middelværdien for x.

## or

Katalog > 

*BoolskUdtryk1* **or** *BoolskUdtryk2*  
returnerer *boolsk udtryk*

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

*BoolskMatrix1* **or** *BoolskMatrix2*  
returnerer *Boolsk matrix*

Returnerer true eller false eller en forenklet form af den oprindelige indtastning.

Returnerer true, hvis enten et eller begge udtryk kan reduceres til true. Returnerer kun false, hvis begge udtryk evalueres til false.

**Bemærk:** Se *xor*.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

*Heltal1* **or** *Heltal2* ⇒ *heltal*

```

Define g(x)=Func
    If x<=0 or x>=5
        Goto end
    Return x*3
    Lbl end
EndFunc

```

---

*g(3)* 9

---

*g(0)* *A function did not return a value*

I hexadecimal tilstand:

---

0h7AC36 or 0h3D5F 0h7BD7F

---

**Vigtigt:** Tallet nul, ikke bogstavet O.

I binær tilstand:

Sammenligner to reelle heltal bit for bit med en or-operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarede bits sammenlignes, er resultatet 1, hvis en af bittene er 1. Resultatet er kun 0, hvis begge bits er 0. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemstilstand.

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimalt indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talsystem).

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Yderligere oplysninger findes under ►Base2, side 17.

**Bemærk:** Se xor.

0b100101 or 0b100

0b100101

**Bemærk:** En binær indtastning kan have op til 64 cifre (præfikset 0b ikke medregnet). En hexadecimal indtastning kan have op til 16 cifre.

## ord()

**ord**(*Streng*)⇒*heltal*

ord("hello")	104
--------------	-----

**ord**(*Liste1*)⇒*liste*

char{104}	"h"
-----------	-----

Returnerer den numeriske kode til første tegn i tegnstrengen *Streng*, eller en liste med de første tegn i hvert listeelement.

ord(char{24})	24
---------------	----

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

## P

## P►Rx()

**P►Rx**(*rUdtr*,  $\theta$ *Udtr*)⇒*udtryk*

I vinkeltilstanden Radian:

**P►Rx**(*rListe*,  $\theta$ *Liste*)⇒*liste*

P►Rx(4,60°)	2.
-------------	----

**P►Rx**(*rMatrix*,  $\theta$ *Matrix*)⇒*matrix*

P►Rx( $\{-3,10,1.3\}, \{\frac{\pi}{3}, \frac{\pi}{4}, 0\}$ )	$\{-1.5, 7.07107, 1.3\}$
--	--------------------------

Returnerer den ækvivalente x-koordinat til parret  $(r, \theta)$ .

**Bemærk:** Argumentet  $\theta$  tolkes i grader, nygrader eller radianer afhængigt af den aktuelle vinkeltilstand. Hvis argumentet er et udtryk, kan du anvende  $^{\circ}$ ,  $^{\text{G}}$  eller  $^{\text{r}}$  til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive `P@>Rx (...)`.

`P▶Ry(rTal, θTal)⇒værdi`

I vinkeltilstanden Radian:

`P▶Ry(rListe, θListe)⇒liste`

`P▶Ry(4,60°)` 3.4641

`P▶Ry(rMatrix, θMatrix)⇒matrix`

`P▶Ry({-3,10,1.3}, {π/3, π/4, 0})`  
`{-2.59808,-7.07107,0}`

Returnerer den ækvivalente y-koordinat til parret  $(r, \theta)$ .

**Bemærk:** Argumentet  $\theta$  tolkes i grader, nygrader eller radianer afhængigt af den aktuelle vinkeltilstand.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive `P@>Ry (...)`.

**PassErr**

Se et eksempel på **PassErr** i Eksempel2 under **Try**-kommandoen, side 167.

Videresender en fejl til næste niveau.

Hvis systemvariabel `errCode` er nul, gør **PassErr** ingenting.

**Else** betingelsen i **Try...Else...EndTry**-blokken bør anvende **ClrErr** eller **PassErr**. Brug **ClrErr**, hvis fejlen skal behandles eller ignoreres. Brug **PassErr**, hvis det ikke er kendt, hvad der skal gøres ved fejlen, for at sende den til den næste fejlhåndtering. Hvis der ikke er flere ventende **Try...Else...EndTry**-fejlhåndteringer, vises fejldialogboksen som normalt.



**Bemærk:** Se også **ClrErr**, side 23, og **Try**, side 167.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

## piecewise()

Katalog > 

**piecewise**(*Udtr1* [, *Betingelse1* [, *Udtr2* [, *Betingelse2* [, ... ]]]])

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

Returnerer definitioner for en stykkevis funktion i form af en liste. Du kan også oprette stykkevise definitioner ved hjælp af en skabelon.

$p(1)$	1
$p(-1)$	undef

**Bemærk:** Se også **Stykkevis skabelon**, side 3.

## poissCdf()

Katalog > 

**poissCdf**( $\lambda$ , *nedreGrænse*, *øvreGrænse*)  $\Rightarrow$  tal  
hvis *nedreGrænse* og *øvreGrænse* er tal,  
*liste* hvis *nedreGrænse* og *øvreGrænse* er lister

**poissCdf**( $\lambda$ , *øvreGrænse*) (for  $P(0 \leq X \leq \text{øvreGrænse}) \Rightarrow$  tal hvis *øvreGrænse* er et tal, liste hvis *øvreGrænse* er en liste

Beregner den kumulerede sandsynlighed for den diskrete Poisson-distribution med en angivet middelværdi  $\lambda$ .

For  $P(X \leq \text{øvreGrænse})$ , sæt *nedreGrænse*=0

## poissPdf()

Katalog > 

**poissPdf**( $\lambda$ , *XVærdi*)  $\Rightarrow$  tal hvis *XVærdi* er et tal, *liste* hvis *XVærdi* er en liste

Beregner en sandsynlighed for den diskrete Poisson-distribution med den angivne middelværdi  $\lambda$ .

*Vektor* ►Polar $[1 \ 3.]$ ►Polar $[3.16228 \ \angle 71.5651]$ 

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @>Polar.

Viser *vektor* i polær form  $[r \angle \theta]$ . Vektoren skal være af dimensionen 2 og kan være en række eller kolonne.

**Bemærk:** ►Polar er en displayformat-kommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje, og den opdaterer ikke *ans*.

**Bemærk:** Se også ►Rect, side 129.

*kompleksVærdi* ►Polar

Viser *kompleksVærdi* i polær form.

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

*kompleksVærdi* kan have enhver kompleks form. Men en  $re^{i\theta}$ -indtastning udløser en fejl i vinkeltilstanden Grader.

**Bemærk:** Du skal anvende parenteser til en  $(r \angle \theta)$  polær indtastning.

I vinkeltilstanden Radian:

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

I vinkeltilstanden Nygrader:

 $(4i)$ ►Polar $(4 \angle 100.)$ 

I vinkeltilstanden Grader:

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

## polyEval()

polyEval(Liste1, Udtr1)⇒udtryk

polyEval({1,2,3,4},2)

26

polyEval(Liste1, Liste2)⇒udtryk

polyEval({1,2,3,4},{2,-7})

{26, -262}

Fortolker første argument som koefficienten til et polynomium i faldende grad og returnerer polynomiet beregnet for værdien af det andet argument.

## polyRoots()

Katalog > 

**polyRoots(Poly, Var)** ⇒

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

**polyRoots(ListeAfKoeff)** ⇒

$$\text{cPolyRoots}(y^3+1,y) \\ \{-1, 0.5-0.8660254i, 0.5+0.8660254i\}$$

Den første syntaks, **polyRoots(Poly, Var)**, returnerer en liste af reelle rødder af polynomiet *Poly* med hensyn til variabelen *Var*. Hvis der ikke findes en reel rod, returneres en tom liste: {}.

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

*Poly* skal være et polynomium i udviklet form i en variabel. Anvend ikke uudviklede former som  $y^2\cdot y+1$  eller  $x\cdot x+2\cdot x+1$

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

Den anden syntaks, **polyRoots(ListeAfKoeff)**, returnerer en liste med reelle rødder for koefficienterne i *ListeAfKoeff*.

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

## PowerReg

Katalog > 

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

Beregner den lineære regression  $y = (a \cdot (x)^b)$  på liste *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver hyppigheden af hændelse for hvert tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for *X* og *Y* data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot (x)^b$
stat.a, stat.b	Regressionskoefficienter
stat.r <sup>2</sup>	Koefficient af en lineær forklaringsgrad til transformerede data
stat.r	Korrelationskoefficient til transformerede data ( $\ln(x)$ , $\ln(y)$ )
stat.Resid	Residualer forbundet med eksponentielmodellen
stat.ResidTrans	Residualer associeret med lineær tilpasning af transformerede data
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

**Prgm***Blok***EndPrgm**

Skabelon til oprettelse af et brugerdefineret program. Skal anvendes sammen med **Define**, **Define LibPub**, eller **Define LibPriv**-kommandoer.

*Blok* kan være en enkelt sætning, en række sætninger adskilt med kolon eller en række sætninger på separate linjer.

Beregn GCD og vis mellemresultater.

---

```

Define proggcd(a,b)=Prgm
  Local d
  While b≠0
  d:=mod(a,b)
  a:=b
  b:=d
  Disp a," ",b
  EndWhile
  Disp "GCD=",a
EndPrgm

```

---

*Done*

**Prgm**

Katalog &gt;

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

<code>proggcd(4560,450)</code>	
	450 60
	60 30
	30 0
	GCD=30
	Done

**prodSeq()**Se  $\Pi()$ , side 198.**Product (P1)**Se  $\Pi()$ , side 198.**product()**

Katalog &gt;

**product(Liste[, Start[, slut]])**  $\Rightarrow$  *udtryk*

Returnerer produktet af elementerne indeholdt i *Liste*. *Start* og *Slut* er valgfri. De angiver en serie af elementer.

**product(Matrix1[, Start[, slut]])**  $\Rightarrow$  *matrix*

Returnerer en rækkevektor med produkterne af elementerne i kolonnerne i *Matrix1*. *Start* og *slut* er valgfri. De angiver en serie af rækker.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

<code>product({1,2,3,4})</code>	24
<code>product({4,5,8,9},2,3)</code>	40
<code>product(<math>\begin{pmatrix} 1 &amp; 2 &amp; 3 \\ 4 &amp; 5 &amp; 6 \\ 7 &amp; 8 &amp; 9 \end{pmatrix}</math>)</code>	$[28 \ 80 \ 162]$
<code>product(<math>\begin{pmatrix} 1 &amp; 2 &amp; 3 \\ 4 &amp; 5 &amp; 6 \\ 7 &amp; 8 &amp; 9 \end{pmatrix},1,2)</math></code>	$[4 \ 10 \ 18]$

**propFrac()**

Katalog &gt;

**propFrac(Værdi1[, Var])**  $\Rightarrow$  *værdi*

**propFrac(rationalt\_tal)** returnerer *rationalt\_tal* som summen af et heltal og en brøk med samme fortegn og en større nævner end tæller.

<code>propFrac(<math>\frac{4}{3}</math>)</code>	$1 + \frac{1}{3}$
<code>propFrac(<math>-\frac{4}{3}</math>)</code>	$-1 - \frac{1}{3}$

**propFrac**(*rationalt\_Udtryk*, *Var*)  
 returnerer summen af ægte brøker og et polynomium med hensyn til *Var*. Graden af *Var* i nævneren overstiger graden af *Var* i tælleren i hver enkelt ægte brøk. Ens potenser af *Var* samles. Leddene og deres faktorer sorteres med *Var* som hovedvariabel.

Hvis *Var* udelades, foretages en udvikling i ægte brøker med hensyn til den hyppigst forekommende variabel. Koefficienterne af polynomiumdelen gøres derefter ægte med hensyn til deres hyppigst forekommende variabel osv.



## Q

## QR

**QR** *Matrix*, *qmatNavn*, *rmatNavn*[, *Tol*]

Beregner Householder QR faktoropløsningen af en reel eller kompleks matrix. De resulterende Q og R-matricer lagres i de angivne *matNavne*. Q matrix er unitær. R matrix er øvre-triangulær.

Ethvert matricelement kan valgfrit behandles som nul, hvis dets absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers ignoreres *Tol*.

- Hvis du anvender   eller indstiller **Auto eller tilnærmet** - tilstanden til Approks, foretages beregningerne med aritmetik med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:  
 $5E-14 \cdot \max(\dim(\text{Matrix})) \cdot \text{rowNorm}(\text{Matrix})$

Tallet med flydende komma (9). i m1 får resultaterne beregnet med flydende komma.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix}$
QR <i>m1</i> , <i>qm</i> , <i>rm</i>	Done
<i>qm</i>	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

QR faktoropløsningen beregnes numerisk med Householder-transformationer. Den symbolske løsning beregnes med Gram-Schmidt. Kolonnerne i *qmatNavn* er ortonormale vektorer, der udspænder rummet defineret ved *matrix*.

## QuadReg

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

Beregner andengrads polynomiell regression  $y = a \cdot x^2 + b \cdot x + c$  på listerne *X* og *Y* med hyppighed *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver frekvensen af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for *X* og *Y* data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot x^2 + b \cdot x + c$

stat.a, stat.b, stat.c	Regressionskoefficienter
stat.r <sup>2</sup>	Forklaringsgraden
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <i>X</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

## QuartReg

Katalog > 

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

Beregner den polynomielle tredjegradsregression

$y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$  på listerne *X* og *Y* med frekvens *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med hyppigheder. Hvert element i *Frekv* angiver frekvensen af hændelse for hver tilsvarende *X* og *Y* datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for *X* og *Y* data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.



Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Regressionskoefficienter
stat.r <sup>2</sup>	Forklaringsgraden
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

## R

### R ▶ Pθ()

R ▶ Pθ (*xValue*, *yValue*) ⇒ værdi

R ▶ Pθ (*xList*, *yList*) ⇒ liste

R ▶ Pθ (*xMatrix*, *yMatrix*) ⇒ matrix

Returnerer den ækvivalente θ-koordinat for (x,y) argumentparret.

**Bemærk:** Resultatet returneres som en vinkel i grader, nygrader eller radian afhængig af den aktuelle vinkeltilstand.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **R(▶Ptheta (...))**.

I vinkeltilstanden Grader:

R▶P0(2,2) 45.

I vinkeltilstanden Nygrader:

R▶P0(2,2) 50.

I vinkeltilstanden Radian:

R▶P0(3,2) 0.588003

R▶P0([3 -4 2], [0  $\frac{\pi}{4}$  1.5])  
[0. 2.94771 0.643501]

**R ▶ Pr()**Katalog > **R ▶ Pr** (*xValue*, *yValue*) ⇒ værdi**R ▶ Pr** (*xList*, *yList*) ⇒ liste**R ▶ Pr** (*xMatrix*, *yMatrix*) ⇒ matrixReturnerer den ækvivalente r-kordinat for (*x*,*y*) argumentparret.**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **R@>Pr (...)**.

I vinkeltilstanden Radian:

---

**R ▶ Pr**(3,2) 3.60555

---

---

**R ▶ Pr** $\left(\left[3 \ -4 \ 2\right], \left[0 \ \frac{\pi}{4} \ 1.5\right]\right)$ 

---

 $\left[3 \ 4.07638 \ \frac{5}{2}\right]$ 

---

**▶ Rad**Katalog > *Value1* ▶ **Rad** ⇒ værdi

Konverterer argumentet til vinkelmåling i radian.

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive **@>Rad**.

I vinkeltilstanden grader:

---

(1.5) ▶ **Rad** (0.02618)<sup>r</sup>

---

I vinkeltilstanden Nygrader:

---

(1.5) ▶ **Rad** (0.023562)<sup>r</sup>

---

**rand()**Katalog > **rand()** ⇒ *udtryk***rand**(#*Trials*) ⇒ liste**rand()** returnerer en vilkårlig værdi mellem 0 og 1.**rand**(#*Trials*) returnerer en liste med #*Trials* vilkårlig værdier mellem 0 og 1.

Angiv seed-værdien for et vilkårligt tal.

---

RandSeed 1147 Done

---

rand(2) {0.158206,0.717917}

---

**randBin()**Katalog > **randBin**(*n*, *p*) ⇒ *udtryk***randBin**(*n*, *p*, #*Trials*) ⇒ liste**randBin**(*n*, *p*) returnerer et vilkårlig reelt tal fra en given binomialfordeling.**randBin**(*n*, *p*, #*Trials*) returnerer en liste med #*Trials* vilkårlige reelle tal fra en given binomialfordeling.

---

randBin(80,0,5) 46.

---

randBin(80,0,5,3) {43.,39.,41.}

---

**randInt()**Katalog > **randInt**

(  
*lowBound,upBound*)  
 ⇒ *udtryk*

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

**randInt**

(*lowBound,upBound*  
 ,#Trials) ⇒ *liste*

**randInt**

(  
*lowBound,upBound*)  
 returnerer et  
 vilkårligt heltal i det  
 område, der angives  
 af heltalsgrænserne  
*lowBound* og  
*upBound*.

**randInt**

(*lowBound,upBound*  
 ,#Trials) returnerer  
 en liste med #Trials  
 vilkårlige heltal i det  
 angivne område.

**randMat()**Katalog > 

**randMat**(*numRows, numColumns*) ⇒  
*matrice*

Returnerer en matrix med heltal mellem  
 -9 og 9 af den angivne dimension.

Begge argumenter skal kunne reduceres  
 til heltal.

RandSeed 1147	Done
randMat(3,3)	$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$

**Bemærk:** Værdierne i denne matrix ændres,  
 hver gang du trykker .

**randNorm()**Katalog > 

**randNorm**( $\mu, \sigma$ ) ⇒ *udtryk*

**randNorm**( $\mu, \sigma, \#Trials$ ) ⇒ *liste*

**randNorm**( $\mu, \sigma$ ) returnerer et decimalt  
 tal fra den angivne normalfordeling. Det  
 kan være ethvert reelt tal, men vil være  
 kraftigt koncentreret i intervallet [ $\mu-3\cdot\sigma$ ,  
 $\mu+3\cdot\sigma$ ].

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

## randNorm()

Katalog > 

**randNorm**( $\mu$ ,  $\sigma$ , #Trials) returnerer en liste med #Trials decimaltal fra den angivne normalfordeling.

## randPoly()

Katalog > 

**randPoly**(Var, Order) ⇒ *udtryk*

Returnerer et polynomium i Var af den angivne Order. Koefficienterne er vilkårlige heltal i området -9 til 9. Koefficienten af højeste grad vil ikke være nul.

Order skal være 0–99.

RandSeed 1147	Done
randPoly(x,5)	$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$

## randSamp()

Katalog > 

**randSamp**(List, #Trials, [noRepl]) ⇒ *liste*

Returnerer en liste med en vilkårlig stikprøve af #Trials målinger fra List med mulighed for tilbagelægning (noRepl=0) eller ingen tilbagelægning (noRepl=1). Standardindstillingen er med tilbagelægning.

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

## RandSeed

Katalog > 

**RandSeed** Number

Hvis Number = 0, indstilles startværdien til fabriksindstillingerne for vilkårlig-talgeneratoren. Hvis Number ≠ 0, anvendes det til at generere to startværdier, der lagres i systemvariablerne seed1 og seed2.

RandSeed 1147	Done
rand()	0.158206

## real()

Katalog > 

**real**(ValueI) ⇒ *værdi*

Returnerer den reelle del af argumentet.

**real**(ListI) ⇒ *liste*

real(2+3·i)	2
real({1+3·i,3,i})	{1,3,0}

Returnerer realdelen af alle elementer.

**real(Matrix1) ⇒ matrix**

Returnerer realdelen af alle elementer.

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

## ► Rect

*Vektor* ► **Rect**

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**Rect**.

Viser *Vektor* i rektangulær form [x, y, z]. Vektoren skal være af dimensionen 2 eller 3 og kan være en række eller kolonne.

**Bemærk:** ► **Rect** er en displayformatkommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje, og den opdaterer ikke *ans*.

**Bemærk:** Se også ► **Polær ligning**, side 118.

*complexValue* ► **Rect**

Viser *complexValue* i rektangulær form a+bi. *complexValue* kan have enhver kompleks form. Men en  $re^{i\theta}$ -indtastning udløser en fejl i vinkeltilstanden Grader.

**Bemærk:** Du skal anvende parenteser til en ( $r \angle \theta$ ) polær indtastning.

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

I vinkeltilstanden Radian:

$$\left(4 \cdot e^{\frac{\pi}{3}}\right) \text{►Rect} \quad 11.3986$$

$$\left(4 \angle \frac{\pi}{3}\right) \text{►Rect} \quad 2.+3.4641 \cdot i$$

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden grader:

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

**Bemærk:** Du kan skrive  $\angle$  ved at vælge det i symbollisten i kataloget.

$\text{ref}(\text{Matrix1}, \text{Tol}) \Rightarrow \text{matrix}$

Returnerer række-echelonformen af *Matrix1*.

Ethvert matriceelement kan valgfrit behandles som nul, hvis dets absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke har fået tildelt en værdi. Ellers ignoreres *Tol*.

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

- Hvis du anvender `ctrl` `enter` eller indstiller **Auto- eller Approximate** -tilstanden til Approximate, foretages beregningerne med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:  $5E-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

Undgå udefinerede elementer i *Matrix1*. De kan føre til uventede resultater.

Hvis for eksempel *a* er udefineret i følgende udtryk, vil en advarsel blive vist, og resultatet vises som:

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

Advarslen vises, fordi det generaliserede element  $1/a$  ikke ville være gyldig for  $a=0$ .

Du kan undgå dette ved i forvejen at gemme en værdi *tila* eller ved at bruge betingelsesoperatoren ("`|`") til at substituere en værdi som vist i følgende eksempel.


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

Bemærk: Se også rref(), page 140.

## RefreshProbeVars

## RefreshProbeVars

Giver dig adgang til sensordata fra alle tilkoblede sensorprober i dit TI-Basic-program.

StatusVar-værdi	Status
<i>statusVar</i> =0	Normal (fortsæt med programmet) Vernier DataQuest™-applikation er i dataindsamlingstilstand.
<i>statusVar</i> =1	<b>Bemærk:</b> Vernier DataQuest™-applikation skal være i målerstilstand for at denne kommando virker. 
<i>statusVar</i> =2	Vernier DataQuest™-applikation blev ikke startet.
<i>statusVar</i> =3	Vernier DataQuest™-applikationen blev startet, men du har ikke tilkoblet nogen prober.

## Eksempel

```
Define temp()=
Prgm
© Tjek, om systemet er klar
RefreshProbeVars-status
If status=0 Then
Disp "klar"
For n,1,50
RefreshProbeVars-status
temperatur:=meter.temperatur
Disp "Temperatur: ",temperatur
If temperature>30 Then
Disp "For varmt"
EndIf
© Vent i 1 sekund mellem
stikprober
Wait 1
EndFor
Else
Disp "Ikke klar. Prøv igen senere"
EndIf
EndPrgm
```

Bemærk: Dette kan også bruge med TI-

## remain()

**remain(Value1, Value2)** ⇒ værdi

**remain(List1, List2)** ⇒ liste

**remain(Matrix1, Matrix2)** ⇒ matrix

Returnerer resten af det første argument med hensyn til det andet argument som defineret af identiteterne:

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

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

Bemærk, at som følge heraf, **remain**  $(-x,y) = -\text{remain}(x,y)$ . Resultatet er enten nul, eller det har samme fortegn som det første argument.

**Bemærk:** Se også **mod()**, side 100.

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

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

## Request

**Request** *promptString*, *var* [, *DispFlag* [, *statusVar*]]

**Request** *promptString*, *func*(*arg1*, ...*argn*) [, *DispFlag* [, *statusVar*]]

Programmeringskommando: Standser programmet midlertidigt og viser en dialogboks med meddelelsen *promptString* og et indtastningsfelt til brugerens svar.

Når brugeren skriver et svar og klikker på **OK**, tildeles indtastningsfeltets indhold til variabelen *var*.

Hvis brugeren klikker på **Annuller**, fortsætter programmet uden at acceptere noget input. Programmet bruger den tidligere værdi for *var* hvis *var* var defineret i forvejen.

Det valgfrie argument *DispFlag* kan være et hvilket som helst udtryk.

Definer et program:

```
Define request_demo()=Prgm
  Request "Radius: ",r
  Disp "Areal = ",pi*r^2
EndPrgm
```

Start programmet, og indtast et svar:

request\_demo()



Resultat efter valg af **OK**:

```
Radius: 6/2
Areal = 28,2743
```



- Hvis *DispFlag* er udelades eller evalueres til **1**, vil klarmeddelelsen og brugerens svar blive vist i beregnerens historik.
- Hvis *DispFlag* evalueres til **0**, vil klarmeddelelsen og svaret ikke blive vist historikken.

Det valgfri argument *statusVar* gør det muligt for programmet at bestemme, hvordan brugeren forlod dialogboksen. Bemærk, at *statusVar* kræver argumentet *DispFlag*.

- Hvis brugeren klikkede **OK** eller trykkede på **Enter** eller **Ctrl+Enter**, indstilles variablen *statusVar* til en værdi på **1**.
- I modsat fald indstilles variablen *statusVar* til en værdi på **0**.



Med argumentet *func()* kan et program lagre brugerens svar som en funktionsdefinition. Denne syntaks virker, ligesom hvis brugeren udførte kommandoen:

```
Define func(arg1, ...argn) = brugers svar
```

Så kan programmet bruge den definerede funktion *func()*. *promptString* bør vejlede brugeren til at indtaste en passende *brugers svar*, der fuldender funktionsdefinitionen.

**Bemærk:** Du kan bruge kommandoen **Request** i et brugerdefineret program, men ikke i en funktion.

Sådan stoppes et program, der indeholder kommandoen **Request** i en uendelig løkke:

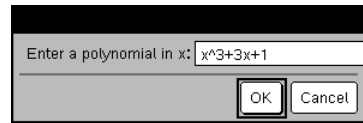
- **Håndholdt:** Hold tasten  nede, mens du gentagne gange trykker på .
- **Windows®:** Hold tasten **F12** nede, mens

Definer et program:

```
Define polynomial()=Prgm
  Request "Indtast et polynomium i
x:",p(x)
  Disp "Reelle rødder er:",polyRoots
(p(x),x)
EndPrgm
```

Start programmet, og indtast et svar:

```
polynomium()
```



Resultat efter indtastning af  $x^3+3x+1$  og valg af **OK**:

```
Reelle rødder er: {-0.322185}
```

du gentagne gange trykker på **Enter**.

- **Macintosh**®: Hold tasten **F5** nede, mens du gentagne gange trykker på **Enter**.
- **iPad**®: App'en viser en meddelelse. Du kan fortsat vente eller annullere.

**Bemærk:** Se også **RequestStr**, page 134.



## RequestStr

**RequestStr** *promptString*, *var*[, *DispFlag*]

Programmeringskommando: Fungerer identisk med den første syntaks i kommandoen **Request**, bortset fra at brugerens svar altid fortolkes som en streng. I modsætning hertil fortolker kommandoen **Request** svarene som et udtryk, medmindre brugeren omslutter det med citationstegn ("").

**Bemærk:** Du kan bruge kommandoen **RequestStr** i et brugerdefineret program, men ikke i en funktion.

Sådan standses et program, der indeholder en **RequestStr** kommando i en uendelig løkke:

- **Håndholdt:** Hold tasten  nede, mens du gentagne gange trykker på .
- **Windows**®: Hold tasten **F12** nede, mens du gentagne gange trykker på **Enter**.
- **Macintosh**®: Hold tasten **F5** nede, mens du gentagne gange trykker på **Enter**.
- **iPad**®: App'en viser en meddelelse. Du kan fortsat vente eller annullere.

**Bemærk:** Se også **Request**, page 132.

Definer et program:

```
Define requestStr_demo()=Prgm
    RequestStr "Dit navn:",name,0
    Disp "Svaret har ",dim(name),"
    tegn."
EndPrgm
```

Start programmet, og indtast et svar:

```
requestStr_demo()
```



Resultat efter valg af **OK** (Bemærk, at argumentet *DispFlag* fra **0** udelader klarmeddelelsen og svaret fra historikken):

```
requestStr_demo()
```

Svaret har fem tegn.

**Return** [*Expr*]

Returnerer *Expr* som resultatet af funktionen. Bruges i en **Func...EndFunc**-blok.

**Bemærk:** Brug **Return** uden et argument med en **Prgm...EndPrgm**-blok for at lukke et program.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

```
Define factorial (nn)=
Func
Local answer,counter
1 → answer
For counter,1,nn
answer·counter → answer
EndFor
Return answer|
EndFunc
```

*factorial* (3)

6

**right()**

**right**(*List1* [, *Num*]) ⇒ *liste*

Returnerer *Num*-elementer længst til højre i *List1*.

Hvis du udelader *Num*, returneres hele *List1*.

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

Returnerer *Num*-tegn længst til højre i tegnstrengen *sourceString*.

Hvis du udelader *Num*, returneres hele *sourceString*.

**right**(*Comparison*) ⇒ *udtryk*

Returnerer højre side af en ligning eller ulighed.

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

{ 3,-2,4 }

*right*("Hello",2)

"lo"

**rk23 ()**

**rk23**(*Expr*, *Var*, *depVar*, {*Var0*, *VarMax*}, *depVar0*, *VarStep* [, *diftol*]) ⇒ *matrix*

**rk23**(*SystemOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep* [, *diftol*]) ⇒ *matrix*

Differentialligning:

$y' = 0.001 * y * (100 - y)$  og  $y(0) = 10$

<b>rk23</b> ( $0.001 \cdot y \cdot (100 - y)$ , $t, y$ , {0,100}, 10, 1)				
0.	1.	2.	3.	4.
10.	10.9367	11.9493	13.042	14.2

**rk23**(*ListOfExpr*, *Var*, *ListOfDepVars*, {*Var0*, *VarMax*}, *ListOfDepVars0*, *VarStep*, *difto1*) ⇒ *matrix*

Anvender Runge-Kutta-metoden til at løse systemet

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

med  $\text{depVar}(\text{Var}0) = \text{depVar}0$  i intervallet [*Var0*, *VarMax*]. Returnerer en matrix, hvor første række definerer *Var*-outputværdierne som defineret af *VarStep*. Den anden række definerer værdien af den første løsningskomponent i de tilsvarende *Var*-værdier, osv.

*Expr* er højresiden, som definerer den ordinære differentialligning (ODE - ordinary differential equation).

*SystemOfExpr* er et system af højresider, der definerer ODE'erne (svarende til rækkefølgen af afhængige variable i *ListOfDepVars*).

*ListOfExpr* er en liste af højresider, der definerer systemet af ODE'er (svarende til rækkefølgen af afhængige variable i *ListOfDepVars*).

*Var* er den uafhængige variabel.

*ListOfDepVars* er en liste med afhængige variabler.

{*Var0*, *VarMax*} er en liste med to elementer, der informerer funktionen om at integrere fra *Var0* til *VarMax*.

*ListOfDepVars0* er en liste af oprindelige værdier for afhængige variabler.

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

Samme ligning med *difto1* sat til 1.E-6

$$\text{rk23}\left(0.001 \cdot y \cdot \{100 - y\}, t, y, \{0, 100\}, 10, 1, 1.E-6\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

System af 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



## rotate()

Katalog > 

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#ofRotations* er negativt, kører rotationen mod højre. Standardindstillingen er  $-1$  (rotér en bit til højre).

For eksempel i en højrerotation:

Hver bit roterer til højre.

0b00000000000001111010110000110101

Bitten længst til højre roterer længst mod venstre.

giver:

0b1000000000000111101011000011010

Resultatet vises i den valgte tilstand for talsystem.

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

Returnerer en kopi af *ListI* roteret til højre eller venstre med *#ofRotations*-elementer. Ændrer ikke *ListI*.

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#ofRotations* er negativt, kører rotationen mod højre. Standardindstillingen er  $-1$  (roter en bit til højre).

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

Returnerer en kopi af *StringI* roteret til højre eller venstre med *#ofRotations*-tegn. Ændrer ikke *StringI*.

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#ofRotations* er negativt, kører rotationen mod højre. Standardindstillingen  $-1$  (rotér en bit til højre).

I hexadecimal tilstand:

rotate(0h78E)	0h3C7
rotate(0h78E,-2)	0h8000000000001E3
rotate(0h78E,2)	0h1E38

**Vigtigt:** Til binære eller hexadecimal indtastninger skal du som præfiks altid benytte henholdsvis 0b eller 0h (nul, ikke bogstavet 0).

I decimal tilstand:

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

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"bcda"

## round()

Katalog > 

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

round(1.234567,3)	1.235
-------------------	-------

Returnerer argumentet afrundet til det angivne antal cifre efter decimalpunktet.

## round()

Katalog > 

*digits* skal være et heltal i området 0–12. Hvis *digits* ikke er inkluderet, returneres argumentet afrundet til 12 væsentlige cifre.

**Bemærk:** Vis cifre-tilstanden kan påvirke den måde, dette vises på.

**round(List1[, digits])** ⇒ *liste*

Returnerer en liste med elementerne afrundet til det angivne antal cifre.

$$\text{round}(\{\pi, \sqrt{2}, \ln(2)\}, 4)$$

---

$$\{3.1416, 1.4142, 0.6931\}$$

**round(Matrix1[, digits])** ⇒ *matrix*

Returnerer en matrix med elementerne afrundet til det angivne antal cifre.

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

---

$$\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$$

## rowAdd()

Katalog > 

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

Returnerer en kopi af *Matrix1* med rækken *rIndex2* udskiftet med summen af rækker *rIndex1* og *rIndex2*.

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

---

$$\begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$$

## rowDim()

Katalog > 

**rowDim(Matrix)** ⇒ *udtryk*

Returnerer antallet af rækker i *Matrix*.

**Bemærk:** Se også **colDim()**, side 24.

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

---

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

---

$$3$$

## rowNorm()

Katalog > 

**rowNorm(Matrix)** ⇒ *udtryk*

Returnerer maksimum for summerne af de absolutte værdier for elementerne i *Matrix*-rækkerne.

**Bemærk:** Alle matrixens elementer skal kunne reduceres til tal. Se også **colNorm()**, side 24.

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

---

$$25$$

## rowSwap()

Katalog > 

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

Returnerer *Matrix1* med rækkerne *rIndex1* og *rIndex2* byttet.

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

## rref()


Katalog > 

**rref**(*Matrix1* [, *Tol*]) ⇒ *matrix*

Returnerer den reducerede række-echelonform af *Matrix1*.

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

Ethvert element af matricen kan valgfrit behandles som nul, hvis dets absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers ignoreres *Tol*.

- Hvis du anvender  eller indstiller **Auto- eller Approximate** -tilstanden til Approximate, foretages beregningerne med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:  
 $5E-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

**Bemærk:** Se også **ref()**, page 130.

## S

## sec()

 -tast

**sec**(*Værdi1*) ⇒ *værdi*

I vinkeltilstanden Grader:

**sec**(*Liste1*) ⇒ *list*

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



## sec()

-tast

Returnerer sekans til *Værdi1* eller returnerer en liste med sekansen til alle elementer i *Liste1*.

**Bemærk:** Argumentet fortolkes som en vinkel målt i grader, nygrader eller radianer afhængigt af den aktuelt indstillede vinkeltilstand. Du kan bruge °, G eller r til midlertidigt at ignorere vinkeltilstanden.

## sec<sup>-1</sup>()

-tast

sec<sup>-1</sup>(*Værdi1*) ⇒ værdi

I vinkeltilstanden Grader:

sec<sup>-1</sup>(*Liste1*) ⇒ liste

sec<sup>-1</sup>(1) 0.

Returnerer den vinkel, hvis sekans er *Værdi1* eller returnerer en liste med de inverse sekanser til de enkelte elementer på *Liste1*.

I vinkeltilstanden Nygrader:

**Bemærk:** Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.

sec<sup>-1</sup>(√2) 50.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arcsec (...)**.

I vinkeltilstanden Radian:

sec<sup>-1</sup>{1,2,5} {0,1.0472,1.36944}

## sech()

Katalog > 

sech(*Værdi1*) ⇒ værdi

sech(3) 0.099328

sech(*Liste1*) ⇒ list

sech{1,2,3,4}  
{0.648054,0.198522,0.036619}

Returnerer den hyperbolske sekans til *Værdi1* eller returnerer en liste med den hyperbolske sekans til elementerne *Liste1*.

## sech<sup>-1</sup>()

Katalog > 

sech<sup>-1</sup>(*Værdi1*) ⇒ værdi

I vinkeltilstanden radian og tilstanden rektangulært kompleks:

sech<sup>-1</sup>(*Liste1*) ⇒ liste

Returnerer den inverse hyperbolske sekans til *Værdi1* eller returnerer en liste med den inverse hyperbolske sekans til de enkelte elementer i *Liste1*.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive `arcsech (...)`.

sech <sup>-1</sup> (1)	0
sech <sup>-1</sup> ({1,-2,2.1})	{0,2.0944·i,8.Е-15+1.07448·i}

## Send

## Hub-menu

**Send** *exprOrString1*, *exprOrString2*] ...

Programmeringskommando: Send én eller flere TI-Innovator™ Hub kommandoer til en tilsluttet hub.

*exprOrString* skal være en gyldig TI-Innovator™ Hub kommando. Normalt vil *exprOrString* indeholde en "SET ..." kommando for at kontrollere en enhed eller en "READ ..." kommando for at anmode om data.

Argumenterne sendes til hubben efter hinanden.

**Bemærk:** Du kan bruge kommandoen **Send** i et brugerdefineret program, men ikke i en funktion.

**Bemærk:** Se også **Get** (side 62), **GetStr** (side 68), og **eval()** (side 50).

Eksempel: Tænd det blå element i den indbyggede RGB LED i 0,5 sekunder.

Send "SET COLOR.BLUE ON TIME .5"	Done
----------------------------------	------

Eksempel: Anmod om den nuværende værdi for hubbens indbyggede lysniveausensor. En **Get**-kommando henter værdien og tildeler den til den variable *lightval*.

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

Eksempel: Send en beregnet frekvens til hubbens indbyggede højttaler. Brug den særlige variable *iostr.SendAns* til at vise hub-kommandoen med udtrykket evalueret.

<i>n</i> :=50	50
<i>m</i> :=4	4
Send "SET SOUND eval(m·n)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

## seq()

Katalog >

**seq(Udtr, Var, Lav, Høj[, Trin])** ⇒ liste

Øger *Var* fra *Low* til *High* i trin på *Step*, beregner *Expr* og returnerer resultaterne som en liste. Det oprindelige indhold af *Var* er der stadigvæk, når **seq()** er gennemført.

Standardværdien for *trin* = 1.

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

**Bemærk:** Sådanne gennemtvungne et tilnærmet resultat,

**Håndholdt:** Tryk på .

**Windows®:** Tryk **Ctrl+Enter**.

**Macintosh®:** Tryk på **⌘+Enter**.

**iPad®:** Hold **ENTER** nede, og vælg .

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

## seqGen()

Katalog >

**seqGen(Expr, Var, depVar, {Var0, VarMax}[, ListOfInitTerms [, VarStep [, CeilingValue]])** ⇒ liste

Genererer en liste med led for sekvensen  $depVar(Var)=Expr$  som følger: Øger den uafhængige variabel *Var* fra *Var0* til *VarMax* med *VarStep*, beregner *depVar* (*Var*) for tilsvarende værdier af *Var* vha. udtrykket *Expr* og *ListOfInitTerms* og returnerer resultaterne som en liste.

**seqGen(ListOrSystemOfExpr, Var, ListOfDepVars, {Var0, VarMax} [, MatrixOfInitTerms [, VarStep [, CeilingValue]])** ⇒ matrix

Genererer de første fem led i sekvensen  $u(n) = u(n-1)^2/2$  med  $u(1)=2$  og  $VarStep=1$ .

$\text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right)$
$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$

Eksempel, hvor  $Var0=2$ :

$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right)$
$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$

System med to sekvenser:

Genererer en matrix af led for et system (eller en liste) af sekvenser

*ListOfDepVars*

(*Var*)=*ListOrSystemOfExpr* som følger:

Øger den uafhængige variabel *Var* fra *Var0* til *VarMax* med *VarStep*, beregner

*ListOfDepVars(Var)* for tilsvarende

værdier af *Var* vha. udtrykket

*ListOrSystemOfExpr* og

*MatrixOfInitTerms*, og returnerer resultaterne som en matrix.

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u_2^{n-1}}{2} + u_1^{n-1}\right\}, n, \{u_1, u_2\}, \{1, 5\}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{2} & \frac{19}{2} \end{bmatrix}$$

Bemærk: Void ( ) i den oprindelige ledmatrix ovenfor bruges for at angive, at det oprindelige led for  $u_1(n)$  er beregnet vha. den eksplicitte sekvensformel  $u_1(n)=1/n$ .

Det oprindelige indhold af *Var* er uændret efter **seqGen()** er gennemført .

Standardværdien for *VarStep* = 1.

## seqn()

**seqn**(*Expr(u, n [, ListOfInitTerms[, nMax [, CeilingValue]]])*)⇒*liste*

Genererer en liste med led for sekvensen  $u(n)=\text{Expr}(u, n)$  som følger: Øger  $n$  fra 1 til  $nMax$  med 1, beregner  $u(n)$  for de tilsvarende værdier af  $n$  vha. udtrykket  $\text{Expr}(u, n)$  og *ListOfInitTerms*, og returnerer resultaterne som en liste.

Genererer de første fem led i sekvensen  $u(n) = u(n-1)/2$  med  $u(1)=2$ .

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

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

**seqn**(*Expr(n [, nMax [, CeilingValue]]*)⇒*liste*

Genererer en liste med led for sekvens  $u(n)=\text{Expr}(n)$  som følger: Øger  $n$  fra 1 til  $nMax$  med 1, beregner  $u(n)$  for de tilsvarende værdier af  $n$  vha. udtrykket  $\text{Expr}(n)$  og returnerer resultaterne som en liste.

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

Hvis  $nMax$  mangler, sættes  $nMax$  til 2500

Hvis  $nMax=0$ , sættes  $nMax$  til 2500

**Bemærk:** **seqn()** kalder **seqGen()** med  $n0=1$  og  $nstep=1$

**setMode**(*tilstandNavnHeltal*,  
*indstilHeltal*) ⇒ *heltal* **setMode**(*liste*)  
⇒ *heltalsliste*

Kun gyldig i en funktion eller et program.

**setMode**(*tilstandNavnHeltal*,  
*indstilHeltal*) indstiller midlertidigt  
tilstanden *tilstandNavnHeltal* til den  
nye indstilling *indstilHeltal* og  
returnerer et heltal, der svarer til denne  
tilstands oprindelige indstilling.  
Ændringen er begrænset til varigheden  
af eksekveringen af  
programmet/funktionen.

*tilstandNavnHeltal* angiver hvilken  
tilstand, du vil indstille. Det skal være et  
af tilstandsheltallene fra nedenstående  
tabel.

*indstilHeltal* angiver den nye indstilling  
for tilstanden. Det skal være et af  
indstillingsheltallene for den tilstand, du  
indstiller.

**setMode**(*liste*) lader dig ændre flere  
indstillinger. *liste* indeholder et par af  
tilstandsheltal og indstillingsheltal.  
**setMode**(*liste*) returnerer en tilsvarende  
liste, hvis heltalspar repræsenterer de  
oprindelige tilstande og indstillinger.

Hvis du har gemt alle  
tilstandsindstillinger med **getMode**(0) →  
*var*, kan du anvende **setMode**(*var*) til at  
gendanne disse indstillinger, indtil  
funktionen eller programmet afsluttes.  
Se **getMode**(), side 67.

**Bemærk:** De aktuelle  
tilstandsindstillinger videresendes til  
kaldte under rutiner. Hvis en under rutine  
ændrer en tilstandsindstilling, mistes  
tilstandsændringen, når kontrollen  
returnerer til den kaldende rutine.

Viser en tilnærmet værdi af  $\pi$  med  
standardindstillingen for Viste cifre, og viser  
derefter  $\pi$  med en indstilling på Fix2. Sørg for,  
at standardindstillingen gendannes efter  
programmet er eksekveret.

Define <i>prog1</i> ()=Prgm	<i>Done</i>
Disp $\pi$	
setMode(1,16)	
Disp $\pi$	
EndPrgm	
<hr/>	
<i>prog1</i> ()	3.14159
	3.14
	<i>Done</i>

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Tilstandsnavn	Tilstandsheltal	Indstillingsheltal
Viste cifre	1	1=Float, 2=Float1, 3=Float2, 4=Float3, 5=Float4, 6=Float5, 7=Float6, 8=Float7, 9=Float8, 10=Float9, 11=Float10, 12=Float11, 13=Float12, 14=Fix0, 15=Fix1, 16=Fix2, 17=Fix3, 18=Fix4, 19=Fix5, 20=Fix6, 21=Fix7, 22=Fix8, 23=Fix9, 24=Fix10, 25=Fix11, 26=Fix12
Vinkel	2	1=Radian, 2=Grader, 3=Gradian
Eksponentielt format	3	1=Normal, 2=Videnskabelig, 3=Teknisk
Reel eller komplekse	4	1=Reel, 2=Rektangulær, 3=Polær
Auto eller tilnærmet	5	1=Auto, 2=Tilnærmet
Vektorformat	6	1=Rektangulær, 2=Cylindrisk, 3=Sfærisk
Talsystem	7	1=Decimal, 2=Hex, 3=Binær

## shift()

**shift(Heltal[,antalFlyt])** ⇒ heltal

Flytter bittene i et binært heltal. Du kan indtaste *Heltal* i ethvert talsystem. Det konverteres automatisk til en 64-bit binær form med fortegn. Hvis *Heltal* er for stort til denne form, bringer en symmetrisk modulooperation værdien ind i det korrekte område. Yderligere oplysninger findes under **Base2**, side 17.

Hvis *antalFlytninger* er positivt, kører flytningen mod venstre. Hvis *antalFlytninger* er negativt, kører flytningen mod højre. Standardindstillingen er -1 (flytter en bit til højre).

I binær tilstand:

shift(0b1111010110000110101)	0b111101011000011010
shift(256,1)	0b1000000000

I hexadecimal tilstand:

shift(0h78E)	0h3C7
shift(0h78E,-2)	0h1E3
shift(0h78E,2)	0h1E38

**Vigtigt:** Til binære eller hexadecimal indtastninger skal du som præfiks altid benytte henholdsvis 0b eller 0h (nul, ikke bogstavet O).

I en flytning til højre droppes bitten længst mod højre, og 0 eller 1 indsættes for at matche bitten længst til venstre. I en flytning til venstre droppes bitten længst mod venstre, og 0 indsættes som bitten længst til højre.

For eksempel i en højreflytning:

Hver bit flytter til højre.

0b000000000000011110101100011010

Indsætter 0, hvis bitten længst til venstre er 0,

eller 1, hvis bitten længst til venstre er 1.

giver:

00b000000000000001111010110001101

Resultatet vises i den valgte tilstand for talsystem. Der vises ikke foranstillede nuller.

**shift(Liste1 [,antalFlytninger])**⇒*liste*

Returnerer en kopi af *Liste1* flyttet til højre eller venstre med *antalflytninger* elementer. Ændrer ikke *Liste1*.

Hvis *antalFlytninger* er positivt, kører flytningen mod venstre. Hvis *antalFlytninger* er negativt, kører flytningen mod højre. Standardindstillingen er -1 (flyt en bit til højre).

Elementer indført i starten eller slutningen af *liste* af flytningen, indstilles til symbol "undef".

**shift(Streng1 [,antalFlytninger])**⇒*streng*

Returnerer en kopi af *Streng1* flyttet til højre eller venstre med *antalflytninger* tegn. Ændrer ikke *Streng1*.

I decimal tilstand:

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("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

## shift()

Katalog > 

Hvis *antalFlytninger* er positivt, kører flytningen mod venstre. Hvis *antalFlytninger* er negativt, kører flytningen mod højre. Standardindstillingen er -1 (flyt et tegn til højre).

Tegn indført i starten eller slutningen af *streng* af flytningen, indstilles til et mellemrum.

## sign()

Katalog > 

**sign(*Værdi*)** ⇒ værdi

$\text{sign}(-3.2)$	-1
---------------------	----

**sign(*Liste*)** ⇒ liste

$\text{sign}\{2,3,4,-5\}$	{1,1,1,-1}
---------------------------	------------

**sign(*Matrix*)** ⇒ matrix

For det reelle eller komplekse *Værdi*, returneres *Værdi* / **abs**(*Værdi*) når *Værdi* ≠ 0.

Hvis kompleks formatilstand er real:

$\text{sign}\begin{bmatrix} -3 & 0 & 3 \end{bmatrix}$	$\begin{bmatrix} -1 & \text{undef} & 1 \end{bmatrix}$
---	---

Returnerer 1, hvis *Værdi* er positivt.

Returnerer -1 hvis *Værdi* er negativt.

**sign(0)** returnerer ±1, hvis det komplekse format er reel. Ellers returnerer det sig selv.

**sign(0)** repræsenterer enhedscirklen i det komplekse domæne.

For en liste eller matrix returneres fortegnene af elementerne.

## simult()

Katalog > 

**simult(*koeffMatrix*, *konstVektor*, *Tol*)** ⇒ matrix

Løs for x og y:

$$x + 2y = 1$$

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

Returnerer en kolonnevektor, der indeholder løsningen til et system af lineære ligninger.

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

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

Løsningen er x=-3 og y=2.



*koeffMatrix* skal være en kvadratisk matrix, der indeholder koefficienterne til ligningerne.

*konstVektor* skal have samme antal rækker (samme dimension) som *koeffMatrix* og indeholde konstanterne.

Ethvert matricielement kan valgfrit behandles som nul, hvis den absolutte værdi er mindre end *Tol*. Denne tolerance anvendes kun, hvis matricen har elementer med flydende decimaler og ikke indeholder symbolske variable, der ikke er tildelt en værdi. Ellers ignoreres *Tol*.

- Hvis du sætter tilstanden **Auto eller tilnærmet** til Approks, foretages beregningerne med aritmetik med flydende komma.
- Hvis *tol* udelades eller ikke anvendes, beregnes standardtolerancen som:  
 $5E-14 \cdot \max(\dim(\textit{koeffMatrix})) \cdot \text{rækkeNorm}(\textit{koeffMatrix})$

**simult**(*koeffMatrix*, *konstMatrix*[, *tol*]) $\Rightarrow$ *matrix*

Løser flere systemer af lineære ligninger, hvor hvert system har de samme ligningskoefficienter men forskellige konstanter.

Hver kolonne i *konstMatrix* skal indeholde konstanterne for et ligningssystem. Hver kolonne i den resulterende indeholder løsningen for det tilsvarende system.

Løs:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{array}{l} \left[ \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right] \rightarrow \textit{matx1} \\ \textit{simult}\left(\textit{matx1}, \left[ \begin{array}{c} 1 \\ 2 \end{array} \right] \right) \end{array} \quad \left[ \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right] \quad \left[ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} \right]$$

Løs:

$$x + 2y = 1$$

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

$$x + 2y = 2$$

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

$$\textit{simult}\left(\left[ \begin{array}{cc} 1 & 2 \\ 3 & 4 \end{array} \right], \left[ \begin{array}{cc} 1 & 2 \\ -1 & -3 \end{array} \right] \right) \quad \left[ \begin{array}{cc} -3 & -7 \\ 2 & \frac{9}{2} \end{array} \right]$$

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

## sin()



**sin**(*Værdi*) $\Rightarrow$ *værdi*

I vinkeltilstanden Grader:

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

## sin()

 -tast

**sin(*Værdi*)** returnerer sinus til argumentet.

**sin(*ListeI*)** returnerer en liste med sinus til alle elementer i *ListeI*.

**Bemærk:** Argumentet fortolkes som en vinkel i enten grader eller radianer afhængigt af den aktuelle vinkeltilstand. Du kan bruge °, G, eller r til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

**sin(*kvadratMatrixI*)** ⇒ *kvadratMatrix*

Returnerer matrixsinus af *kvadratMatrixI*. Dette er ikke det samme som at beregne sinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

*KvadratMatrixI* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

---

$$\sin\left(\frac{\pi}{4}\right) \quad 0.707107$$

---

$$\sin(45) \quad 0.707107$$

---

$$\sin(\{0,60,90\}) \quad \{0,0.866025,1\}$$

---

I vinkeltilstanden Nygrader:

---

$$\sin(50) \quad 0.707107$$

---

I vinkeltilstanden Radian:

---

$$\sin\left(\frac{\pi}{4}\right) \quad 0.707107$$

---

$$\sin(45^\circ) \quad 0.707107$$

---

I vinkeltilstanden Radian:

---

$$\sin\left(\begin{array}{ccc} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{array}\right) \quad \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

---

## sin<sup>-1</sup>()

 -tast

**sin<sup>-1</sup>(*Værdi*)** ⇒ *værdi*

**sin<sup>-1</sup>(*ListeI*)** ⇒ *liste*

**sin<sup>-1</sup>(*Værdi*)** Returnerer den vinkel, hvis sinus er *Værdi*.

**sin<sup>-1</sup>(*ListeI*)** returnerer en liste med de inverse sinusværdier for hvert element af *ListeI*.

**Bemærk:** Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.

I vinkeltilstanden Grader:

---

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

I vinkeltilstanden Nygrader:

---

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

I vinkeltilstanden Radian:

---

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

---

## $\sin^{-1}()$



**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive `arcsin(...)`.

$\sin^{-1}(\text{kvadratMatrixI}) \Rightarrow \text{kvadratMatrix}$

Returnerer den matrixinverse sinus af *kvadratMatrixI*. Dette er ikke det samme som at beregne den inverse sinus for hvert element. Oplysninger om beregningsmetoden findes i `cos()`.

*KvadratMatrixI* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

I vinkeltilstanden radian og tilstanden rektangulært komplekst format:

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

## $\sinh()$

Katalog >

$\sinh(\text{TalverI}) \Rightarrow \text{værdi}$

$$\sinh(1.2) = 1.50946$$

$\sinh(\text{ListeI}) \Rightarrow \text{liste}$

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

$\sinh(\text{VærdiI})$  returnerer den hyperbolske sinus til argumentet.

$\sinh(\text{ListeI})$  returnerer en liste af de hyperbolske sinuser af hvert element af *ListeI*.

$\sinh(\text{kvadratMatrixI}) \Rightarrow \text{kvadratMatrix}$

Returnerer den matrixhyperbolske sinus af *kvadratMatrixI*. Dette er ikke det samme som at beregne den hyperbolske sinus for hvert element. Oplysninger om beregningsmetoden findes i `cos()`.

*KvadratMatrixI* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

I vinkeltilstanden Radian:

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

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

Katalog >

$\sinh^{-1}(\text{VærdiI}) \Rightarrow \text{værdi}$

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

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

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

$\sinh^{-1}(\text{VærdiI})$  returnerer den inverse hyperbolske sinus af argumentet.

sinh<sup>-1</sup>(*Liste1*) returnerer den inverse hyperbolske sinus til hvert element i *Liste1*.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arcsinh** (...).

sinh<sup>-1</sup>(*kvadratMatrix1*) ⇒ *kvadratMatrix*

I vinkeltilstanden Radian:

Returnerer den matrixinverse hyperbolske sinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse hyperbolske sinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

## SinReg

SinReg *X*, *Y* [, [*Iterationer*],[*Periode*] [, [*Kategori*, *Medtag*] ]

Beregner sinusregressionen på listerne *X* og *Y*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Iterationer* er en valgfri værdi, som angiver det maksimale antal gange (1 til 16) en løsning vil forsøges. Hvis udeladt, anvendes 8. Typisk resulterer større værdier i større nøjagtighed men længere eksekveringstider og omvendt.

*Periode* angiver en estimeret periode. Hvis den udelades, skal forskellen mellem værdierne i *X* være lige store og i sekventiel orden. Hvis du angiver *Periode*, kan forskellen mellem *x*-værdierne være forskellig.

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for  $X$  og  $Y$  data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Output af **SinReg** er altid i radianer, uanset vinkelindstillingen.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Regressionskoefficienter
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede $X$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede $Y$ -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

## SortA

**SortA** *Liste1* [, *Liste2*] [, *Liste3*] ...

$\{2,1,4,3\} \rightarrow list1$        $\{2,1,4,3\}$

**SortA** *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

SortA *list1*      Done

Sorterer elementerne i første argument i stigende rækkefølge.

*list1*       $\{1,2,3,4\}$

$\{4,3,2,1\} \rightarrow list2$        $\{4,3,2,1\}$

Hvis du medtager yderligere argumenter, sorteres elementerne i hvert argument således, at deres nye positioner passer til de nye positioner for elementerne i det første argument.

SortA *list2*, *list1*      Done

*list2*       $\{1,2,3,4\}$

*list1*       $\{4,3,2,1\}$

Alle argumenter skal være navne på lister eller vektorer. Alle argumenterne skal have ens dimensioner.

tomme (ugyldige) elementer i det første argument flyttes til bunden. Yderligere oplysninger om tomme elementer findes på side 224.

## SortD

**SortD** *Liste1* [, *Liste2*] [, *Liste3*] ...

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
---------------------------------	---------------

**SortD** *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

$\{1,2,3,4\} \rightarrow list2$	$\{1,2,3,4\}$
---------------------------------	---------------

Identisk med **SortA**, med den undtagelse, at **SortD** sorterer elementerne i faldende rækkefølge.

SortD <i>list1</i> , <i>list2</i>	Done
-----------------------------------	------

<i>list1</i>	$\{4,3,2,1\}$
--------------	---------------

<i>list2</i>	$\{3,4,1,2\}$
--------------	---------------

tomme (ugyldige) elementer i det første argument flyttes til bunden. Yderligere oplysninger om tomme elementer findes på side 224.

## Sphere

*Vektor* ▶**Sphere**

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @>**Sphere**.

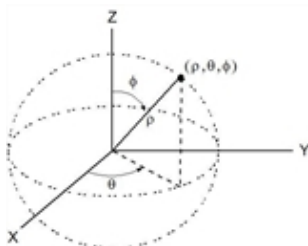
$\left[ \begin{array}{ccc} 1 & 2 & 3 \end{array} \right] \text{▶Sphere}$
$\left[ 3.74166 \quad \angle 1.10715 \quad \angle 0.640522 \right]$

Viser række- eller kolonnevektoren i sfærisk form  $[\rho \angle \theta \angle \phi]$ .

*Vektor* skal have dimensionen 3 og kan være enten en række- eller kolonnevektor.

$\left( \begin{array}{ccc} 2 & \angle \frac{\pi}{4} & 3 \end{array} \right) \text{▶Sphere}$
$\left[ 3.60555 \quad \angle 0.785398 \quad \angle 0.588003 \right]$

**Bemærk:** ▶**Sphere** er en displayformatkommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje.



$\text{sqrt}(V\text{ærdi}) \Rightarrow \text{værdi}$ 

$$\sqrt{4} \quad 2$$

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

$$\sqrt{\{9,2,4\}} \quad \{3,1.41421,2\}$$

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

**Bemærk:** Se også **Kvadratrodsskabelon**, side 1.

**stat.results** $xlist:=\{1,2,3,4,5\} \quad \{1,2,3,4,5\}$ 

Viser resultater fra en statistikberegning.

 $ylist:=\{4,8,11,14,17\} \quad \{4,8,11,14,17\}$ 

Resultaterne vises som en mængde af navn-værdi-par. De viste specifikke navne afhænger af den senest beregnede statistiske funktion eller kommando.

LinRegMx *xlist,ylist,1: stat.results*

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r <sup>2</sup> "	0.996109
"r"	0.998053
"Resid"	"{...}"

Du kan kopiere et navn eller en værdi og sætte den ind andre steder.

<i>stat.values</i>	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	"{-0.4,0.4,0.2,0,-.02}"

**Bemærk:** Undgå at definere variable, der anvender samme navne som dem, der anvendes til statistisk analyse. I visse tilfælde kan der opstå en fejl.

Variabelnavne, der anvendes til statistisk analyse, vises i nedenstående tabel.

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBlock
stat.AdjR <sup>2</sup>	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r <sup>2</sup>	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.ox	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.oy	stat.tList

stat.b6	stat.e	stat.MSReg	stat.ox1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.ox2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.̄x
stat.b9	stat.FBlock	stat.̂p	stat.Σx <sup>2</sup>	stat.̄x1
stat.b10	stat.Fcol	stat.̂p1	stat.Σxy	stat.̄x2
stat.bList	stat.FInteract	stat.̂p2	stat.Σy	stat.̄xDiff
stat.χ <sup>2</sup>	stat.FreqReg	stat.̂pDiff	stat.Σy <sup>2</sup>	stat.̄xList
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg
stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat.ȳ
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat.ŷ
stat.CookDist	stat.MaxX	stat.PValRow	stat.SESlope	stat.ŷList
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

**Bemærk:** Hver gang en liste- & regneark-funktion beregner statistiske resultater, kopierer den "stat."gruppevariable til en "stat#."gruppe, hvor # er et tal der automatisk sammenlignes. Dette gør, at man kan bevare tidligere resultater, mens man udfører flere beregninger.

## stat.values

Katalog > 

### stat.values

Se eksemplet med `stat.results`.

Viser en matrix med de beregnede værdier for den senest beregnede statistiske funktion eller kommando.

I modsætning til `stat.results` udelader `stat.values` de navne, der er knyttet til værdierne.

Du kan kopiere en værdi og sætte den ind andre steder.



**stDevPop**(*Liste* [, *Hyppighedsliste*]) ⇒ *udtryk*

Returnerer population standardafvigelsen af elementerne i *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

**Bemærk:** *Liste* skal have mindst to elementer. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224

**stDevPop**(*Matrix1* [, *Hyppighedsmatrix*]) ⇒ *matrix*

Returnerer en rækkevektor af populationsstandardafvigelser for kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

**Bemærk:** *Matrix1* skal have mindst to rækker. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

I vinkeltilstanden Radian og tilstanden Auto:

$\text{stDevPop}(\{1,2,5,-6,3,-2\})$	3.59398
$\text{stDevPop}(\{1.3,2.5,-6.4\},\{3,2,5\})$	4.11107

$\text{stDevPop}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right)$	$[3.26599 \quad 2.94392 \quad 1.63299]$
$\text{stDevPop}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right)$	$[2.52608 \quad 5.21506]$

## stDevSamp()

**stDevSamp**(*Liste* [, *Hyppighedsliste*]) ⇒ *udtryk*

Returnerer stikprøvestandardafvigelsen af elementerne i *liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

**Bemærk:** *Liste* skal have mindst to elementer. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224

$\text{stDevSamp}(\{1,2,5,-6,3,-2\})$	3.937
$\text{stDevSamp}(\{1.3,2.5,-6.4\},\{3,2,5\})$	4.33345

## stDevSamp()

Katalog > 

**stDevSamp**(*Matrix1* [, *Hyppighedsmatrix*]) ⇒ *matrix*

Returnerer en rækkevektor af standardafvigelser for målingerne i kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

**Bemærk:** *Matrix1* skal have mindst to rækker. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

$\text{stDevSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right)$
$[4. \quad 3.60555 \quad 2.]$
$\text{stDevSamp}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right)$
$[2.7005 \quad 5.44695]$

## Stop

Katalog > 

### Stop

Programmeringskommando: Afslutter programmet.

**Stop** er ikke tilladt i funktioner.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

<i>i</i> :=0	0
Define <i>prog1</i> ()=Prgm	Done
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1</i> ()	Done
<i>i</i>	5

## Store

Se → (store), side 207.

## string()

Katalog > 

**string**(*Udtr*) ⇒ *streng*

Reducerer *Udtr* og returnerer resultatet som en tegnstreng.

string(1.2345)	"1.2345"
string(1+2)	"3"

**subMat()**Katalog > 

**subMat**(*Matrix1* [, *Startrække*] [, *Startkolonne*] [, *Slutrække*] [, *Slutkolonne*]) ⇒ *matrix*

Returnerer den angivne delmatrix af *Matrix1*.

Standardindstillinger: *Startrække*=1, *Startkolonne*=1, *Slutrække*=sidste række, *Slutkolonne*=sidste kolonne.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
<code>subMat(m1,2,1,3,2)</code>	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
<code>subMat(m1,2,2)</code>	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

**Sum (Sigma)**

Se G(), side 198.

**sum()**Katalog > 

**sum**(*Liste* [, *Start*] [, *Slut*]) ⇒ *udtryk*

Returnerer summen af elementer i *Liste*.

*Start* og *Slut* er valgfri. De angiver en serie af elementer.

Alle ugyldige argumenter giver et ugyldigt resultat. Tomme (ugyldige) elementer i *List* ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

**sum**(*Matrix1* [, *Start*] [, *Slut*]) ⇒ *matrix*

Returnerer en rækkevektor med summerne af elementerne i kolonnerne i *Matrix1*.

*Start* og *Slut* er valgfri. De angiver en serie af rækker.

Alle ugyldige argumenter giver et ugyldigt resultat. Tomme (ugyldige) elementer i *Matrix1* ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

<code>sum({1,2,3,4,5})</code>	15
<code>sum({a,2·a,3·a})</code>	"Error: Variable is not defined"
<code>sum(seq(n,n,1,10))</code>	55
<code>sum({1,3,5,7,9},3)</code>	21

<code>sum</code> $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
<code>sum</code> $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
<code>sum</code> $\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, 2, 3\right)$	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

**sumIf(Liste, Kriterie[, SumListe])** ⇒ værdi

Returnerer den akkumulerede sum af alle elementerne i *Liste*, der opfylder det angivne *Kriterie*. Du kan også vælge at angive en alternativ liste, *sumListe*, for at levere de elementer, der skal akkumuleres.

*Liste* kan være et udtryk, en liste eller en matrix. *SumListe* skal, hvis den er angivet, have samme dimensioner som *Liste*.

*Kriterie* kan være:

- En værdi, et udtryk eller en streng. For eksempel akkumulerer **34** kun de elementer i *Liste*, der reduceres til værdien 34.
- Et Boolsk udtryk, der indeholder symbolet **?** som pladsholder for hvert element. For eksempel akkumulerer **? < 10** kun de elementer i *Liste*, der er mindre end 10.

Når et element i *Liste* opfylder *Kriterie*, føjes elementet til den akkumulerede sum. Hvis du inkluderer *sumListe*, tilføjes det tilsvarende element fra *sumListe* til summen i stedet.

I applikationen Lister og regneark kan du anvende et celleområde i stedet for *Liste* og *sumListe*.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 224.

**Bemærk:** Se også **countIf()**, side 31.

---

```
sumIf({1,2,e,3,π,4,5,6},2.5<?<4.5)
```

12.859874482

---

```
sumIf({1,2,3,4},2<?<5,{10,20,30,40})
```

70

---

**system(Værdi1 [, Værdi2 [, Værdi3 [, ...]])**

Returnerer et system af ligninger formateret som en liste. Du kan også oprette et system ved hjælp af en skabelon.

## T

## T (transponere)

$MatrixI^T \Rightarrow matrix$

Returnerer den kompleks konjugerede transponerede  $MatrixI$ .

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

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @t.

## tan()

$\tan(VærdiI) \Rightarrow værdi$

I vinkeltilstanden Grader:

$\tan(ListeI) \Rightarrow liste$

$$\tan\left(\left(\frac{\pi}{4}\right)^r\right) \qquad 1.$$

$\tan(VærdiI)$  returnerer tangens til argumentet.

$$\tan(45) \qquad 1.$$

$\tan(ListeI)$  returnerer en liste med tangens til alle elementer i  $ListeI$ .

$$\tan(\{0,60,90\}) \qquad \{0.,1.73205,undef\}$$

**Bemærk:** Argumentet fortolkes som en vinkel i enten grader eller radianer afhængigt af den aktuelle vinkeltilstand. Du kan bruge °, g, eller r til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

I vinkeltilstanden Nygrader:

$$\tan\left(\left(\frac{\pi}{4}\right)^r\right) \qquad 1.$$

$$\tan(50) \qquad 1.$$

$$\tan(\{0,50,100\}) \qquad \{0.,1.,undef\}$$

I vinkeltilstanden Radian:

$$\tan\left(\frac{\pi}{4}\right) \qquad 1.$$

$$\tan(45^\circ) \qquad 1.$$

$$\tan\left(\left\{\pi, \frac{\pi}{3}, \pi, \frac{\pi}{4}\right\}\right) \qquad \{0.,1.73205,0.,1.\}$$

$\tan(kvadratMatrixI) \Rightarrow kvadratMatrix$

I vinkeltilstanden Radian:

## tan()

 -tast

Returnerer matrixtangens til *kvadratMatrix1*. Dette er ikke det samme som at beregne tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

$$\tan \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

## tan<sup>-1</sup>()

 -tast

tan<sup>-1</sup>(*Værdi1*) ⇒ *værdi*

I vinkeltilstanden Grader:

tan<sup>-1</sup>(*Liste1*) ⇒ *liste*

$$\tan^{-1}(1) \quad 45$$

tan<sup>-1</sup>(*Værdi1*) Returnerer den vinkel, hvis tangent is *Værdi1*.

I vinkeltilstanden Nygrader:

tan<sup>-1</sup>(*Liste1*) returnerer en liste med den inverse tangens til hvert element af *Liste1*.

$$\tan^{-1}(1) \quad 50$$

**Bemærk:** Resultatet returneres som en vinkel i grader eller radianer afhængigt af den aktuelle vinkeltilstand.

I vinkeltilstanden Radian:

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arctan(...)**.

$$\tan^{-1}(\{0,0.2,0.5\}) \quad \{0,0.197396,0.463648\}$$

tan<sup>-1</sup>(*kvadratMatrix1*) ⇒ *kvadratMatrix*

I vinkeltilstanden Radian:

Returnerer den matrixinverse tangens af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$$\tan^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

## tanh()

Katalog > 

tanh(*Værdi1*) ⇒ *værdi*

$$\tanh(1.2) \quad 0.833655$$

tanh(*Liste1*) ⇒ *liste*

$$\tanh(\{0,1\}) \quad \{0,0.761594\}$$

**tanh(Værdi)** returnerer den hyperbolske tangens til argumentet.

**tanh(Liste1)** returnerer en liste med den hyperbolske tangens til hvert enkelt element i *Liste1*.

**tanh(kvadratMatrix1)⇒kvadratMatrix**

Returnerer den matrixhyperbolske tangens af *kvadratMatrix1*. Dette er ikke det samme som at beregne den hyperbolske tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

I vinkeltilstanden Radian:

$$\tanh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

## tanh<sup>-1</sup>()

**tanh<sup>-1</sup>(Værdi)⇒værdi**

**tanh<sup>-1</sup>(Liste1)⇒liste**

**tanh<sup>-1</sup>(Værdi)** returnerer den inverse hyperbolske tangens til argumentet.

**tanh<sup>-1</sup>(Liste1)** returnerer en liste med den inverse hyperbolske tangens til alle elementer i *Liste1*.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **arctanh (...)**.

**tanh<sup>-1</sup>(kvadratMatrix1)⇒kvadratMatrix**

Returnerer den matrixinverse hyperbolske tangens af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse hyperbolske tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

I rektangulært komplekst format:

$$\begin{array}{l} \tanh^{-1}(0) \quad 0. \\ \tanh^{-1}(\{1,2,1,3\}) \\ \{ \text{undef}, 0.518046-1.5708\cdot i, 0.346574-1.570\cdot i \} \end{array}$$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

I vinkeltilstanden Radian og rektangulært komplekst format:

$$\tanh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.099353+0.164058\cdot i & 0.267834-1.4908 \\ -0.087596-0.725533\cdot i & 0.479679-0.9473\cdot i \\ 0.511463-2.08316\cdot i & -0.878563+1.7901 \end{bmatrix}$$

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

**tCdf**(*nedreGrænse*,*øvreGrænse*,*fg*) $\Rightarrow$ *tal*  
 hvis *nedreGrænse* og *øvreGrænse* er tal,  
*liste* hvis *nedreGrænse* og *øvreGrænse* er  
 lister

Beregner Student-*t*  
 sandsynlighedsfordelingen mellem  
*nedreGrænse* og *øvreGrænse* for de  
 angivne frihedsgrader *fg*.

For  $P(X \leq \textit{øvreGrænse})$ , sæt *nedreGrænse*=  
 -9E999.

## Text

**TextpromptStreng**[, *DispFlag*]

Programmeringskommando: Standser  
 programmet midlertidigt og viser  
 tegnstrengen *promptStreng* i en  
 dialogboks.

Når en bruger vælger **OK**, fortsætter  
 programudførelsen.


Det valgfrie argument *flag* kan være  
 et hvilket som helst udtryk.

- Hvis *DispFlag* udelades eller  
 beregnes til **1**, føjes  
 tekstmeddelelsen til Regner-  
 historikken.
- Hvis *DispFlag* udelades eller  
 beregnes til **0**, føjes  
 tekstmeddelelsen ikke til  
 historikken.

Hvis programmet skal bruge et  
 skriftligt svar fra brugeren, henvises til  
**Request**, side 134 eller **RequestStr**, side  
 134.

**Bemærk:** Du kan bruge denne  
 kommando i et brugerdefineret  
 program men ikke i en funktion.

Definer et program, der standser midlertidigt for  
 at vise hvert af fem tilfældige tal i en dialogboks.

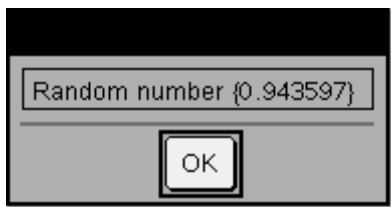
I Prgm...EndPrgm-skabelonen udfyldes hver linje  
 ved at trykke på  i stedet for på **enter**. På  
 computerens tastatur skal du holde **Alt** nede og  
 trykke på **Enter**.

```
Define text_demo()=Prgm
  For i,1,5
    stringfo:="Random number " & string
    (rand(i))
    Text stringfo
  EndFor
EndPrgm
```

Kør programmet:

```
text_demo()
```

Eksempel på en dialogboks:





**tInterval**Katalog > **tInterval** *Liste*[,*Frekv*[,*CNiveau*]]

(Datalisteinput)

**tInterval**  $\bar{x}$ ,*sx*,*n*[,*CNiveau*]

(Sammenfatning, stat input)

Beregner et *t* konfidensinterval En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval for et ukendt populationsgennemsnit
stat. $\bar{x}$	Middelværdi af stikprøven fra den uniforme fordeling
stat.ME	Fejlmargen
stat.fg	Frihedsgrader
stat. $\sigma_x$	Stikprøve standardafvigelse
stat.n	Længde på datasekvensen med målingsgennemsnit

**tInterval\_2Samp**Katalog > 
**tInterval\_2Samp** *Liste1*,*Liste2*  
 [,*Hyppighed1*[,*Hyppighed12*[,*CNiveau*  
 [,*Puljet*]]]]

(Datalisteinput)

**tInterval\_2Samp**  $\bar{x}1$ ,*sx1*,*n1*, $\bar{x}2$ ,*sx2*,*n2*  
 [,*CNiveau*[,*Puljet*]]

(Sammenfatning, stat input)

Beregner et *t* konfidensinterval med to målinger. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

*Puljet=1* puljer varianser. *Puljet=0* puljer ikke varianser.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelings sandsynlighed
stat. $\bar{x}1-\bar{x}2$	Middelværdi stikprøver i i datasekvenserne fra den vilkårlige normalfordeling
stat.ME	Fejlmargen
stat.df	Frihedsgrader
stat. $\bar{x}1$ , stat. $\bar{x}2$	Middelværdi stikprøver i i datasekvenserne fra den vilkårlige normalfordeling
stat. $\sigma x1$ , stat. $\sigma x2$	Stikprøve standardafvigelse for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antal målinger i datasekvenserne
stat.sp	Den puljede standardafvigelse. Beregnes, når <i>Puljet</i> = 1.

## tPdf()

**tPdf(*XVærdi*,*fg*)** ⇒ *tal* hvis *XVærdi* er et tal,  
*liste* hvis *XVærdi* er en liste

Beregner tæthedsfunktionen (pdf) for Student-*t* fordelingen ved en angivet *x* værdi med angivne frihedsgrader *fg*.

## trace()

**trace(*kvadratMatrix*)** ⇒ *værdi*

Returner sporet (sum af alle elementer på hoveddiagonalen) af *kvadratMatrix*.

trace $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	15
<i>a</i> := 12	12
trace $\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}$	24

**Try***blok1***Else***blok2***EndTry**

Eksekverer *blok1*, med mindre der opstår en fejl. Programmets eksekvering fortsætter til *blok2*, hvis der opstår en fejl i *blok1*. Systemvariablen *errCode* indeholder fejlkoden, der gør det muligt for programmet at udføre fejlretning. En liste med fejlkoder findes i "Fejlkoder og fejlmeddelelser," side 234.

*blok1* og *blok2* kan enten være en enkelt sætning eller en række sætninger adskilt med kolon.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

## Eksempel 2

Du kan se kommandoerne **Try**, **ClrErr** og **PassErr** i funktion ved at indtaste programmet `eigenvals()` vist til højre. Kør programmet ved at eksekvere hvert af de følgende udtryk.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, [-1 \quad 2 \quad -3.1]\right)$$

**Bemærk:** Se også **ClrErr**, side 23, og **PassErr**, side 116.

```
Define prog1()=Prgm
  Try
    z:=z+1
    Disp "z incremented."
  Else
    Disp "Sorry, z undefined."
  EndTry
EndPrgm
```

Done

```
z:=1:prog1()
-----
z incremented.
```

Done

```
DelVar z:prog1()
-----
Sorry, z undefined.
```

Done

```
Define eigenvals(a,b)=Prgm
```

© Program `eigenvals(A,B)` viser  
eigenværdier for A-B

**Try**

```
Disp "A= ",a
```

```
Disp "B= ",b
```

```
Disp " "
```

```
Disp "Eigenværdier for A-B
```

```
er: ",eigVl(a*b)
```

**Else**

```
If errCode=230 Then
```

```
  Disp "Error: Produktet af A-B skal  
være en kvadratisk matrix"
```

```
  ClrErr
```

```
Else
```

```
  PassErr
```

```
EndIf
```

EndTry

EndPrgm

**tTest****tTest**  $\mu_0$ ,*Liste*[,*Hyppighed*[,*Hypot*]]

(Datalisteinput)

**tTest**  $\mu_0$ , $\bar{x}$ ,*sx*,*n*,[*Hypot*]

(Sammenfatning, stat input)

Udfører en hypotesetest for et enkelt ubekendt populationsgennemsnit  $\mu$ , når populationens standardafvigelse er ubekendt. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

Test  $H_0: \mu = \mu_0$ , mod en af de følgende:

Til  $H_a: \mu < \mu_0$ , sæt *Hypot*<0

til  $H_a: \mu \neq \mu_0$  (standard), sæt *Hypot*=0

til  $H_a: \mu > \mu_0$ , sæt *Hypot*>0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.t	$(\bar{x} - \mu_0) / (\text{stdafvigelse} / \sqrt{n})$
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.fg	Frihedsgrader
stat. $\bar{x}$	Middelværdi af stikprøver af datasekvensen i <i>Liste</i>
stat.sx	Standardmåleafvigelse for datasekvensen
stat.n	Stikprøvens størrelse

**tTest\_2Samp****tTest\_2Samp** *Liste1*,*Liste2*[,*Hyppighed1*

[,Hypighed2[,Hypot[,Puljet]]]

(Datalisteinput)

**tTest\_2Samp**  $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2$  [,Hypot  
[,Puljet]]

(Sammenfatning, stat input)

Beregner en to-prøvers  $t$  test. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

Test  $H_0: \mu_1 = \mu_2$ , mod en af de følgende:

Til  $H_a: \mu_1 < \mu_2$ , sæt *Hypot*<0

til  $H_a: \mu_1 \neq \mu_2$  (standard), sæt *Hypot*=0

til  $H_a: \mu_1 > \mu_2$ , sæt *Hypot*>0

*Puljet*=1 puljer varianser

*Puljet*=0 puljer ikke varianser

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.t	Standardnormalværdi beregnet som differens af gennemsnit
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader for t-statistik
stat. $\bar{x}1$ , stat. $\bar{x}2$	Middelværdi af stikprøver for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Størrelse på stikprøverne
stat.sp	Den puljede standardafvigelse. Beregnes, når <i>Puljet</i> =1.

### tvmFV()

**tvmFV**( $N, I, PV, Pmt, [PpY], [CpY], [PmtAt]$ ) $\Rightarrow$ værdi

tvmFV(120,5,0,-500,12,12) 77641.1

Finansfunktion, der beregner penges fremtidige værdi.

**tvmFV()**Katalog > 

**Bemærk:** Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 171. Se også **amortTbl()**, side 7.

**tvmI()**Katalog > 

**tvmI**( $N, PV, Pmt, FV, [PpY], [CpY], [PmtAt]$ ) $\Rightarrow$ værdi

tvmI(240,100000,-1000,0,12,12)      10.5241

Finansfunktion, der beregner den årlige rente.

**Bemærk:** Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 171. Se også **amortTbl()**, side 7.

**tvmN()**Katalog > 

**tvmN**( $I, PV, Pmt, FV, [PpY], [CpY], [PmtAt]$ ) $\Rightarrow$ værdi

tvmN(5,0,-500,77641,12,12)      120.

Finansfunktion, der beregner antallet af betalingsperioder.

**Bemærk:** Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 171. Se også **amortTbl()**, side 7.

**tvmPmt()**Katalog > 

**tvmPmt**( $N, I, PV, FV, [PpY], [CpY], [PmtAt]$ ) $\Rightarrow$ værdi

tvmPmt(60,4,30000,0,12,12)      -552.496

Finansfunktion, der beregner beløbet for hver betaling.

**Bemærk:** Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 171. Se også **amortTbl()**, side 7.

**tvmPV**(*N,I,Pmt,FV,[PpY],[CpY],[PmtAt]*) $\Rightarrow$ værdi

tvmPV(48,4,-500,30000,12,12) -3426.7

Finansfunktion, der beregner den aktuelle værdi.

**Bemærk:** Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 171. Se også **amortTbl()**, side 7.

TVM-argument*	Beskrivelse	Datatype
N	Antal betalingsperioder	reelt tal
I	Årlig rentesats	reelt tal
PV	Nutidsværdi	reelt tal
Pmt	Betalingsbeløb	reelt tal
FV	Fremtidsværdi	reelt tal
PpY	Betalinger pr år, standardværdi=1	heltal > 0
CpY	Rentetilskrivninger per år, standardværdi=1	heltal > 0
PmtAt	Betaling, der forfalder ved starten af hver periode, standardværdi=slut	heltal (0=slut, 1=start)

\* Disse argumentnavne for tidsdiskonterede pengeværdier svarer til til TVM-variabelnavnene (som f.eks. **tvm.pv** og **tvm.pmt**), der anvendes af *Calculator* applikationens FinansRegner. Finansfunktioner gemmer dog ikke deres argumentværdier eller resultater i TVM-variablene.

**TwoVar** *X, Y, [Frekv] [, Kategori, Medtag]*

Beregner statistik med to variable. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 155.)

Alle lister skal have ens dimensioner med undtagelse af *Medtag*.

*X* og *Y* er lister med uafhængige og afhængige variable.

*Frekv* er en valgfri liste med Frekvensværdier. Hvert element i *Frekv* angiver hyppigheden af hændelse for hver tilsvarende  $X$  og  $Y$  datapunkt. Standardværdien er 1. Alle elementer skal være heltal  $\geq 0$ .

*Kategori* er en liste, der indeholder numeriske eller strengkategorikoder for  $X$  og  $Y$  data..

*Medtag* er en liste med en eller flere af kategorikoderne. Kun de dataelementer hvis kategorikode er medtaget i denne liste, er medtaget i beregningen.

Et tomt (ugyldigt) element i en af listerne  $X$ ,  $Freq$  eller  $Category$  resulterer i at det tilsvarende element i alle disse lister bliver ugyldigt. Et tomt element i en af listerne  $X1$  til  $X20$  resulterer i at det tilsvarende element i alle disse lister bliver ugyldigt. Yderligere oplysninger om tomme elementer findes på side 224.

Output-variabel	Beskrivelse
stat. $\bar{x}$	Gennemsnit af x-værdier
stat. x	Summen af x-værdier
stat. x2	Summen af x2-værdier
stat.sx	Standardafvigelse for målingen for x
stat. x	Populations standardafvigelse for x
stat.n	Antal datapunkter
stat. $\bar{y}$	Gennemsnit af y-værdier
stat. y	Summen af y-værdier
stat. y <sup>2</sup>	Summen af y2-værdier
stat.sy	Standardafvigelse fra målingen for y
stat. y	Standardafvigelse fra populationen for y
stat. xy	Summen af x · y værdier
stat.r	Korrelationskoefficient



Output-variabel	Beskrivelse
stat.MinX	Minimum af x-værdier
stat.Q <sub>1</sub> X	1. kvartil af x
stat.MedianX	Median af x
stat.Q <sub>3</sub> X	3. kvartil af x
stat.MaxX	Maksimum af x-værdier
stat.MinY	Minimum af y-værdier
stat.Q <sub>1</sub> Y	1. kvartil af y
stat.MedY	Median af y
stat.Q <sub>3</sub> Y	3. kvartil af y
stat.MaxY	Maksimum af y-værdier
stat. (x- ) <sup>2</sup>	Summen af kvadraterne på afvigelser fra middelværdien for x
stat. (y- ) <sup>2</sup>	Summen af kvadraterne på afvigelser fra middelværdien for y

## U

### unitV()

Katalog > 

**unitV(*Vektor1*)** ⇒ vektor

Returnerer enten en række- eller en kolonneenhedsvektor afhængigt af formen af *Vektor1*.

*Vektor1* skal være enten en enkeltrækkematrix eller en enkeltkolonnematrix.

unitV( $\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$ )	$\begin{bmatrix} 0.408248 & 0.816497 & 0.408248 \end{bmatrix}$
unitV( $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ )	$\begin{bmatrix} 0.267261 \\ 0.534522 \\ 0.801784 \end{bmatrix}$

**unLock**Catalog > **unLock***Var1* [, *Var2*] [, *Var3*] ...*a*:=65 65**unLock***Var*.Lock *a* Done

Oplåser de angivne variable eller variabelgrupper. Låste variable kan ikke redigeres eller slettes.

getLockInfo(*a*) 1*a*:=75 "Error: Variable is locked."DelVar *a* "Error: Variable is locked."Se **Lock**, side 89 og **getLockInfo()**, side 67.Unlock *a* Done*a*:=75 75DelVar *a* Done**V****varPop()**Katalog > **varPop**(*Liste*[, *hyppighedsliste*])⇒*udtryk*

varPop({5,10,15,20,25,30}) 72.9167

Returnerer populationsvariansen af *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

**Bemærk:** *Liste* skal indeholde mindst to elementer.

Hvis et element i en af listerne er tom (ugyldig), ignoreres dette element, og det tilsvarende element i den anden liste ignoreres også. Yderligere oplysninger om tomme elementer findes på side 224.

**varSamp()**Katalog > **varSamp**(*Liste*[, *Hyppighedsliste*])⇒*udtryk*varSamp({{1,2,5,6,3,-2}})  $\frac{31}{2}$ Returnerer stikprøvevariansen for *Liste*.varSamp({{1,3,5},{4,6,2}})  $\frac{68}{33}$ 

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

**Bemærk:** *Liste* skal indeholde mindst to elementer.

Hvis et element i en af listerne er tom (ugyldig), ignoreres dette element, og det tilsvarende element i den anden liste ignoreres også. Yderligere oplysninger om tomme elementer findes på side 224.

**varSamp**(*MatrixI*,  
*Hyppighedsmatrix*) $\Rightarrow$ *matrix*

Returnerer en rækkevektor med stikprøvevariansen for hver kolonne i *MatrixI*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *MatrixI*.

**Bemærk:** *MatrixI* skal indeholde mindst to rækker.

Hvis et element i en af matricerne er tom (ugyldig), ignoreres dette element, og det tilsvarende element i den anden matrix ignoreres også. Yderligere oplysninger om tomme elementer findes på side 224.

varSamp	$\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{pmatrix}$	$[4.75 \ 1.03 \ 4]$
varSamp	$\begin{pmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{pmatrix}, \begin{pmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{pmatrix}$	$[3.91731 \ 2.08411]$

## W

### Wait



#### Wait *tidISekunder*

Afbryder afvikling i en periode på *tidISekunder* sekunder.

**Wait** er især nyttig i et program, der behøver en kort forsinkelse for at lade ønskede data blive tilgængelige.

Argumentet *tidISekunder* skal være et udtryk, der reduceres til en decimal værdi inden for området 0 til og med 100. Kommandoer runder denne værdi op til det nærmeste 0,1 sekund.

For at annullere en **Wait** der er i gang,

- **Håndholdt:** Hold tasten  nede, mens du gentagne gange trykker på .
- **Windows®:** Hold tasten **F12** nede, mens du gentagne gange trykker på **Enter**.

For at vente 4 sekunder:

**Wait 4**

For at vente 1/2 sekund:

**Wait 0.5**

For at vente 1,3 sekund ved hjælp af variabelen *seccount*:

**seccount:=1.3**

**Wait seccount**

Dette eksempel tænder en grøn LED i 0,5 sekunder, og slukker den så.

**Send "SET GREEN 1 ON"**

**Wait 0.5**

**Send "SET GREEN 1 OFF"**

- **Macintosh®:** Hold tasten **F5** nede, mens du gentagne gange trykker på **Enter**.
- **iPad®:** App'en viser en meddelelse. Du kan fortsat vente eller annullere.

**Bemærk:** kan du bruge kommandoen **Wait** i et brugerdefineret program, men ikke i en funktion.

## warnCodes ()


**warnCodes**(*Expr1*, *StatusVar*) $\Rightarrow$ udtryk

Beregner udtrykket *Expr1*, returnerer resultatet og gemmer koderne fra eventuelle fejl, der er opstået, i listevariablen *StatusVar*. Hvis der ikke genereres fejl, tildeler denne funktion en tom liste til *StatusVar*.

*Expr1* kan være et vilkårligt, gyldigt matematisk TI-Nspire™- eller TI-Nspire™ CAS-udtryk. Du kan ikke bruge en kommando eller tildeling som *Expr1*.

*StatusVar* skal være et gyldigt variabelnavn.

Se side 242 for at få en oversigt over fejlkoder og tilhørende meddelelser.

 warnCodes(det([1.23456E-999]),warn)	1.23456E-999
warn	{ 10029 }

## when()

**when**(*Betingelse*, *sandtResultat* [, *falskResultat*][, *ubekendtResultat*])  
 $\Rightarrow$ udtryk

Returnerer *sandtResultat*, *falskResultat* eller *ubekendtResultat* afhængigt af, om *betingelse* er true, false eller ubekendt. Returnerer inputtet, hvis der er for få argumenter til at angive det korrekte resultat.

Udelad både *falskResultat* og *ubekendtResultat* for at lave et udtryk, der kun er defineret i området, hvor *Betingelse* er true.

when( $x < 0, x + 3$ ), $x = 5$	undef
---------------------------------	-------

## when()

Katalog > 

Anvend en **undef** *falskResultat* til at definere et udtryk, der kun tegner grafen til et interval.

**when()** er nyttig til definition af rekursive funktioner.

$\text{when}(n > 0, n \cdot \text{factorial}(n-1), 1) \rightarrow \text{factorial}(n)$	Done
$\text{factorial}(3)$	6
$3!$	6

## While

Katalog > 

### While *Betingelse*

*Blok*

### EndWhile

Udfører sætningerne i *Blok*, så længe *Betingelse* er true.

*Blok* kan enten være en enkelt sætning eller en serie sætninger adskilt med kolon.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Define $\text{sum\_of\_recip}(n)$ =Func	
Local $i, \text{tempsum}$	
$1 \rightarrow i$	
$0 \rightarrow \text{tempsum}$	
While $i \leq n$	
$\text{tempsum} + \frac{1}{i} \rightarrow \text{tempsum}$	
$i + 1 \rightarrow i$	
EndWhile	
Return $\text{tempsum}$	
EndFunc	
	Done
$\text{sum\_of\_recip}(3)$	$\frac{11}{6}$

## X

## xor

Katalog > 

*BoolskUdtryk1* **xor** *BoolskUdtryk2*  
returnerer *Boolsk udtryk*

$\text{true xor true}$	false
$5 > 3 \text{ xor } 3 > 5$	true

*BoolskListe1* **xor** *BoolskListe2* returnerer  
*Boolsk liste*

*BoolskMatrix1* **xor** *BoolskMatrix2*  
returnerer *Boolsk matrix*

Returnerer true, hvis *Boolsk Udtr1* er true, og *Boolsk Udtr2* er false eller omvendt.

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulo-operation til at bringe værdien ind i det korrekte område. Yderligere oplysninger findes under ►**Base2**, side 17.

**Bemærk:** Se **or**, side 114.

*Heltal1 xor Heltal2 ⇒ heltal*

Sammenligner to reelle heltal bit for bit med en **xor**-operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarende bits sammenlignes, er resultatet 1, hvis en af bittene (men ikke begge to) er 1. Resultatet er 0, hvis begge bits er 0 eller begge bits er 1. Den returnerede værdi repræsenterer bitresultaterne og vises i overensstemmelse med den valgte talsystemstilstand.

Du kan indtaste heltallene i ethvert talsystem. Til binære eller hexadecimalt indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talsystem).

Hvis du indtaster et decimalt heltal, der er for stort til en 64-bit binær form med fortegn, anvendes en symmetrisk modulooperation til at bringe værdien ind i det korrekte område.

**Bemærk:** Se **or**, side 114.

I hexadecimal tilstand:

**Vigtigt:** Tallet nul, ikke bogstavet 0.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

I binær tilstand:

0b100101 xor 0b100	0b100001
--------------------	----------

**Bemærk:** En binær indtastning kan have op til 64 cifre (præfikset 0b ikke medregnet). En hexadecimal indtastning kan have op til 16 cifre.

## Z

### zInterval

**zInterval**  $\sigma$ ,*Liste*[,*Hyppighed*[,*CNiveau*]]

(Datalisteinput)

**zInterval**  $\sigma$ , $\bar{x}$ ,*n* [,*CNiveau*]

(Sammenfatning, stat input)

Beregner et  $z$  konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 155.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval for en ukendt populationsmiddelværdi
stat. $\bar{x}$	Middelværdi af stikprøven fra den uniforme fordeling
stat.ME	Fejlmargen
stat.sx	Stikprøve standardafvigelse
stat.n	Længde af datasekvens med stikprøvemiddelværdi
stat. $\sigma$	Kendt populationsstandardafvigelse for datasekvensen <i>Liste</i>

## zInterval\_1Prop

**zInterval\_1Prop**  $x, n$  [,CNiveau]

Beregner én-proportion  $z$  konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 155.)

$x$  er et ikke-negativt heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
stat. $\hat{p}$	Den beregnede brøkdelt af succeser
stat.ME	Fejlmargen
stat.n	Antal stikprøver i datasekvens

## zInterval\_2Prop

**zInterval\_2Prop**  $x1, n1, x2, n2$  [,CNiveau]

Beregner et to-proportion  $z$  konfidensinterval. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

$x_1$  og  $x_2$  er ikke-negative heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingsandsynlighed
stat. $\hat{p}$ Diff	Den beregnede differens mellem proportioner
stat.ME	Fejlmargen
stat. $\hat{p}_1$	Første stikprøves proportionsestimat
stat. $\hat{p}_2$	Anden stikprøves proportionsestimat
stat.n1	Stikprøvestørrelsen i datasekvens 1
stat.n2	stikprøvestørrelsen i datasekvens 2

## zInterval\_2Samp

**zInterval\_2Samp**  $\sigma_1, \sigma_2, \text{Liste1}, \text{Liste2}$   
[,Hyppighed1[,Hyppighed2,[CNiveau]]]

(Datalisteinput)

**zInterval\_2Samp**  $\sigma_1, \sigma_2, \bar{x}_1, n_1, \bar{x}_2, n_2$   
[,CNiveau]

(Sammenfatning, stat input)

Beregner et  $z$  konfidensinterval med to målinger. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingsandsynlighed



Output-variabel	Beskrivelse
stat. $\bar{x}1$ - $\bar{x}2$	Middelværdi stikprøver i i datasekvenserne fra den vilkårlige normalfordeling
stat.ME	Fejlmargen
stat. $\bar{x}1$ , stat. $\bar{x}2$	Middelværdi stikprøver i datasekvenserne fra den vilkårlige normalfordeling
stat. $\sigma x1$ , stat. $\sigma x2$	Stikprøve standardafvigelse for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antal stikprøver i datasekvenserne
stat.r1, stat.r2	Kendt populationsstandardafvigelse for datasekvenserne <i>Liste 1</i> og <i>Liste 2</i>

## zTest

Katalog > 

**zTest**  $\mu0, \sigma, Liste, [Hyppighed[, Hypot]]$

(Datalisteinput)

**zTest**  $\mu0, \sigma, \bar{x}, n[, Hypot]$

(Sammenfatning, stat input)

Udfører en  $z$  test med hyppigheden *Hyppighedsliste*. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

Test  $H_0: \mu = \mu0$ , mod en af de følgende:

Til  $H_a: \mu < \mu0$ , sæt *Hypot*<0

Til  $H_a: \mu \neq \mu0$  (standard), sæt *Hypot*=0

Til  $H_a: \mu > \mu0$ , sæt *Hypot*>0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.z	$(\bar{x} - \mu0) / (\sigma / \text{kvrod}(n))$
stat.P Værdi	Mindste sandsynlighed, ved hvilken nul-hypotesen kan forkastes
stat. $\bar{x}$	Middelværdi af stikprøver af datasekvensen i <i>Liste</i>
stat.sx	Standardmåleafvigelse for datasekvensen. Returneres kun for <i>Data</i> input.

Output-variabel	Beskrivelse
stat.n	Stikprøvens størrelse

## zTest\_1Prop

Katalog > 

### zTest\_1Prop $p_0, x, n[, Hypot]$

Beregner en én-proportion  $z$  test. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

$x$  er et ikke-negativt heltal.

Test  $H_0: p = p_0$  mod en af følgende:

Til  $H_a: p > p_0$ , sæt *Hypot*>0

til  $H_a: p \neq p_0$  (standard), sæt *Hypot*=0

til  $H_a: p < p_0$ , sæt *Hypot*<0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.p0	Antaget populationsproportion
stat.z	Standardnormalværdi beregnet for proportionen
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat. $\hat{p}$	Estimeret stikprøveproportion
stat.n	Stikprøvens størrelse

## zTest\_2Prop

Katalog > 

### zTest\_2Prop $x_1, n_1, x_2, n_2[, Hypot]$

Beregner en toproportional  $z$  test. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

$x_1$  og  $x_2$  er ikke-negative heltal.

Test  $H_0: p_1 = p_2$  mod en af følgende:

Til  $H_a: p_1 > p_2$ , sæt *Hypot*>0

til  $H_a: p1 \neq p2$  (standard), sæt  $Hypot=0$

til  $H_a: p < p0$ , sæt  $Hypot<0$

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

Output-variabel	Beskrivelse
stat.z	Standardnormalværdi beregnet som differens af proportioner
stat.PVal	Mindste signifikansniveau, ved hvilket hvilket nul-hypotesen kan forkastes
stat.p1	Første stikprøves proportionsestimat
stat.p2	Anden stikprøves proportionsestimat
stat.p	Puljet stikprøves proportionsestimat
stat.n1, stat.n2	Antal stikprøver taget i forsøg 1 og 2

**zTest\_2Samp**  $\sigma_1, \sigma_2, List1, List2$   
 $[, Hyppighed1 [, Hyppighed2 [, Hypot]]]$

(Datalisteinput)

**zTest\_2Samp**  $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2 [, Hypot]$

(Sammenfatning, stat input)

Beregner en to-prøvers  $z$  test. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 155.)

Test  $H_0: \mu 1 = \mu 2$ , mod af de følgende:

Til  $H_a: \mu 1 < \mu 2$ , sæt  $Hypot < 0$

til  $H_a: \mu 1 \neq \mu 2$  (standard), sæt  $Hypot = 0$

til  $H_a: \mu 1 > \mu 2$ ,  $Hypot > 0$

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 224.

<b>Output-variabel</b>	<b>Beskrivelse</b>
stat.z	Standardnormalværdi beregnet som differens af gennemsnit
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat. $\bar{x}1$ , stat. $\bar{x}2$	Middelværdi af stikprøver for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Størrelse på stikprøverne

# Symboler

## + (adder)

**+**-tast

$Værdi1 + Værdi2 \Rightarrow værdi$

56	56
----	----

Returnerer summen af de to argumenter.

$56+4$	60
--------	----

$60+4$	64
--------	----

$64+4$	68
--------	----

$68+4$	72
--------	----

$Liste1 + Liste2 \Rightarrow liste$

$\left\{22, \pi, \frac{\pi}{2}\right\} \rightarrow l1$	$\{22, 3.14159, 1.5708\}$
--	---------------------------

$Matrix1 + Matrix2 \Rightarrow matrix$

Returnerer en liste (eller matrix), der indeholder summerne af tilsvarende elementer i *Liste1* og *Liste2* (eller *Matrix1* og *Matrix2*).

$\left\{10, 5, \frac{\pi}{2}\right\} \rightarrow l2$	$\{10, 5, 1.5708\}$
--	---------------------

$l1+l2$	$\{32, 8.14159, 3.14159\}$
---------	----------------------------

Argumenternes dimensioner må være ens.

$Tal + Liste1 \Rightarrow liste$

$15 + \{10, 15, 20\}$	$\{25, 30, 35\}$
-----------------------	------------------

$Liste1 + Tal \Rightarrow liste$

$\{10, 15, 20\} + 15$	$\{25, 30, 35\}$
-----------------------	------------------

Returnerer en liste med summerne af *Tal* og hvert element i *Liste1*.

$Tal + Matrix1 \Rightarrow matrix$

$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
---	--

$Matrix1 + Tal \Rightarrow matrix$

Returnerer en matrix med *Tal* adderet til hvert element på diagonalen af *Matrix1*. *Matrix1* skal være kvadratisk.

**Bemærk:** Anvend .+ (punktum plus) til at addere et udtryk til hvert element.

## -(subtraher)

**-**-tast

$Værdi1 - Værdi2 \Rightarrow værdi$

$6-2$	4
-------	---

Returnerer *Værdi1* minus *Værdi2*.

$\pi - \frac{\pi}{6}$	2.61799
-----------------------	---------

**-(subtraher)****-**-tast $Liste1 - Liste2 \Rightarrow liste$ 

$$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\} \quad \left\{ 12, 1.85841, 0 \right\}$$

 $Matrix1 - Matrix2 \Rightarrow matrix$ 

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

Subtraherer hvert element i *Liste2* (eller *Matrix1*) fra det tilsvarende element i *Liste1* (eller *Matrix1*), og returnerer resultaterne.

Argumenternes dimensioner må være ens.

 $Tal - Liste1 \Rightarrow liste$ 

$$15 - \{10, 15, 20\} \quad \{5, 0, -5\}$$

 $Liste1 - Tal \Rightarrow liste$ 

$$\{10, 15, 20\} - 15 \quad \{-5, 0, 5\}$$

Subtraherer hver *Liste1*-element fra *Tal* eller subtraherer *Tal* fra hvert *Liste1*-element og returnerer en liste med resultaterne.

 $Tal - Matrix1 \Rightarrow matrix$ 

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

 $Matrix1 - Tal \Rightarrow matrix$ 

$Tal - Matrix1$  returnerer en matrix med *Tal* gange identitetsmatricen minus *Matrix1*. *Matrix1* skal være kvadratisk.

$Matrix1 - Tal$  returnerer en matrix på *Tal* gange identitetsmatricen subtraheret fra *Matrix1*. *Matrix1* skal være kvadratisk.

**Bemærk:** Anvend .- (punktum minus) til at subtrahere et udtryk fra hvert element.

**·(multiplicer)****·**-tast $Værdi1 \cdot Værdi2 \Rightarrow værdi$ 

$$2 \cdot 3.45 \quad 6.9$$

Returnerer produktet af de to argumenter.

 $Liste1 \cdot Liste2 \Rightarrow liste$ 

$$\{1., 2, 3\} \cdot \{4, 5, 6\} \quad \{4, 10, 18\}$$

Returnerer en liste med produkterne af de tilsvarende elementer i *Liste1* og *Liste2*.

Listernes dimensioner må være ens.

**· (multiplicer)****× -tast** $Matrix1 \cdot Matrix2 \Rightarrow matrix$ Returnerer matrixproduktet af  $Matrix1$  og  $Matrix2$ .

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 7 & 8 \\ 7 & 8 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 42 & 48 \\ 105 & 120 \end{bmatrix}$$

Antallet af kolonner i  $Matrix1$  skal være lig med antallet af rækker i  $Matrix2$ . $Tal \cdot Liste1 \Rightarrow liste$ 

$$\pi \cdot \{4,5,6\} = \{12.5664, 15.708, 18.8496\}$$

 $Liste1 \cdot Tal \Rightarrow liste$ Returnerer en liste med produkterne af  $Tal$  og hvert element i  $Liste1$ . $Tal \cdot Matrix1 \Rightarrow matrix$ 

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 = \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

 $Matrix1 \cdot Tal \Rightarrow matrix$ Returnerer en matrix med produkterne af  $Tal$  og hvert element i  $Matrix1$ .

$$6 \cdot \text{identity}(3) = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

**Bemærk:** Anvend  $\cdot$  (punktum gangetegn) til at gange et udtryk med hvert element.**/ (divider)****÷ -tast** $Værdi1 / Værdi2 \Rightarrow værdi$ Returnerer kvotienten af  $Værdi1$  divideret med  $Værdi2$ .

$$\frac{2}{3.45} = 0.57971$$

**Bemærk:** Se også **Brøkskabelon**, side 1. $Liste1 / Liste2 \Rightarrow liste$ Returnerer en liste med kvotienterne af  $Liste1$  divideret med  $Liste2$ .

$$\frac{\{1,2,3\}}{\{4,5,6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

Listernes dimensioner må være ens.

 $Tal / Liste1 \Rightarrow liste$ 

$$\frac{6}{\{3,6,\sqrt{6}\}} = \{2, 1, 2.44949\}$$

 $Liste1 / Tal \Rightarrow liste$ Returnerer en liste med kvotienterne af  $Tal$  divideret med  $Liste1$  eller  $Liste1$  divideret med  $Tal$ .

$$\frac{\{7,9,2\}}{7 \cdot 9 \cdot 2} = \left\{\frac{1}{18}, \frac{1}{14}, \frac{1}{63}\right\}$$

 $Tal / Matrix1 \Rightarrow matrix$  $Matrix1 / Tal \Rightarrow matrix$ 

$$\frac{\begin{bmatrix} 7 & 9 & 2 \end{bmatrix}}{7 \cdot 9 \cdot 2} = \begin{bmatrix} \frac{1}{18} & \frac{1}{14} & \frac{1}{63} \end{bmatrix}$$

## / (divider)

-tast

Returnerer en matrix med kvotienterne af  $Matrix1/Tal$ .

**Bemærk:** Anvend . / (punktum divisionstegn) til at dividere et udtryk med hvert element.

## ^ (potens)

-tast

$Værdi1 \wedge Værdi2 \Rightarrow værdi$

$$4^2 \qquad 16$$

$Liste1 \wedge Liste2 \Rightarrow liste$

$$\{2,4,6\} \{1,2,3\} \qquad \{2,16,216\}$$

Returnerer det første argument opløftet til potensen af det andet argument.

**Bemærk:** Se også **Eksponentskabelon**, side 1.

Ved en liste, returneres elementerne i  $Liste1$  opløftet til potensen af de tilsvarende elementer i  $Liste2$ .

I det reelle domæne anvender brøkpotenser, der har reducerede eksponenter med ulige nævnere, den reelle gren, i stedet for den principale gren i kompleks tilstand.

$Tal \wedge Liste1 \Rightarrow liste$

$$\pi \{1,2,-3\} \qquad \{3.14159,9.8696,0.032252\}$$

Returnerer  $Tal$  opløftet til potensen af elementerne i  $Liste1$ .

$Liste1 \wedge Tal \Rightarrow liste$

$$\{1,2,3,4\}^{-2} \qquad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

Returnerer elementerne i  $Liste1$  opløftet til potensen af  $Tal$ .

$kvadratMatrix1 \wedge heltal \Rightarrow matrix$

Returnerer  $kvadratMatrix1$  opløftet til  $heltal$ spotensen.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \qquad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

$kvadratMatrix1$  skal være en kvadratisk matrix.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \qquad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

Hvis  $heltal = -1$ , beregnes den inverse matrix.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \qquad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

Hvis  $heltal < -1$ , beregnes den inverse matrix til en passende positiv potens.



**x<sup>2</sup> (kvadrat)****x<sup>2</sup> -tast***Værdi*<sup>2</sup> ⇒ *værdi*

4 <sup>2</sup>	16
----------------	----

Returnerer kvadratet af argumentet.

{2,4,6} <sup>2</sup>	{4,16,36}
----------------------	-----------

*Liste*<sup>2</sup> ⇒ *liste*

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$	$\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$
---	--

Returnerer en liste med kvadraterne på elementerne i *Liste1*.*kvadratMatrix*<sup>2</sup> ⇒ *matrix*

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^{\wedge 2}$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$
--	--

Returnerer matrix i anden potens af *kvadratMatrix1*. Dette er ikke det samme som at beregne kvadratet på hvert element. Brug <sup>^</sup>2 til at beregne kvadratet på hvert element.**.+ (punktum plustegn)****.+ -taster***Matrix1* .+ *Matrix2* ⇒ *matrix*

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix}$	$\begin{bmatrix} 11 & 32 \\ 23 & 44 \end{bmatrix}$
--	--

*Tal* .+ *Matrix1* ⇒ *matrix*

5 .+ $\begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix}$	$\begin{bmatrix} 15 & 35 \\ 25 & 45 \end{bmatrix}$
---	--

*Matrix1* .+ *Matrix2* returnerer en matrix, der er summen af hvert par tilsvarende elementer i *Matrix1* og *Matrix2*.*Tal* .+ *Matrix1* returnerer en matrix, der er summen af *Tal* og hvert element i *Matrix1*.**.- (punktum minus.)****.- -taster***Matrix1* .- *Matrix2* ⇒ *matrix*

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .- \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$	$\begin{bmatrix} -9 & -18 \\ -27 & -36 \end{bmatrix}$
--	---

*Tal* .- *Matrix1* ⇒ *matrix*

5 .- $\begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix}$	$\begin{bmatrix} -5 & -15 \\ -25 & -35 \end{bmatrix}$
---	---

*Matrix1* .- *Matrix2* returnerer en matrix, der er differensen mellem hvert par af tilsvarende elementer i *Matrix1* og *Matrix2*.*Tal* .- *Matrix1* returnerer en matrix, der er der er differensen af *Tal* og hvert element i *Matrix1*.

**. · (punktum mult.)** **$\cdot$   $\times$  -taster***Matrix1* · *Matrix2* ⇒ *matrix*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 10 & 40 \\ 90 & 160 \end{bmatrix}$$

*Tal* · *Matrix1* ⇒ *matrix*

$$5 \cdot \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 50 & 100 \\ 150 & 200 \end{bmatrix}$$

*Matrix1* · *Matrix2* returnerer en matrix, der er produktet af hvert par tilsvarende elementer i *Matrix1* og *Matrix2*.

*Tal* · *Matrix1* returnerer en matrix med produkterne af *Tal* og hvert element i *Matrix1*.

**. / (punktum divider)** **$\div$  -taster***Matrix1* / *Matrix2* ⇒ *matrix*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \div \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} \end{bmatrix}$$

*Tal* / *Matrix1* ⇒ *matrix*

*Matrix1* / *Matrix2* returnerer en matrix, der er kvotient af hvert par af tilsvarende elementer i *Matrix1* og *Matrix2*.

*Tal* / *Matrix1* returnerer en matrix, der er kvotienten af *Tal* og hvert element i *Matrix1*.

$$5 \div \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{8} \end{bmatrix}$$

**. ^ (punktum potens)** **$\wedge$  -taster***Matrix1* ^ *Matrix2* ⇒ *matrix*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \wedge \begin{bmatrix} 0 & 2 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 27 & \frac{1}{4} \end{bmatrix}$$

*Tal* ^ *Matrix1* ⇒ *matrix*

*Matrix1* ^ *Matrix2* returnerer en matrix, hvor hvert element i *Matrix2* er eksponent til det tilsvarende element i *Matrix1*.

*Tal* ^ *Matrix1* returnerer en matrix, hvor hvert element i *Matrix1* er eksponenten til *Tal*.

$$5 \wedge \begin{bmatrix} 0 & 2 \\ 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 25 \\ 125 & \frac{1}{5} \end{bmatrix}$$

**- (neg)** **$\leftarrow$  -tast**

-Værdi ⇒ værdi

$$-2.43 \rightarrow -2.43$$

-Liste ⇒ liste

$$-\{-1, 0.4, 1.2 \times 10^9\} \rightarrow \{1, -0.4, -1.2 \times 10^9\}$$

## -(neger)

-tast

*-Matrix1* ⇒ *matrix*

Returnerer negationen til argumentet.

Ved en liste eller matrix, returneres alle elementer negeret.

Hvis argumentet er et binært eller hexadecimalt heltal, giver negationen 2's komplement.

I binær tilstand:

**Vigtigt:** Tallet nul, ikke bogstavet O

```
-0b100101
0b11111111111111111111111111111111▶
```

Du kan se hele resultatet ved at trykke på ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

## % (procent)

-taster

*Værdi1 %* ⇒ *værdi*

*Liste1 %* ⇒ *liste*

*Matrix1 %* ⇒ *matrix*

argument

Returnerer 100

For en liste eller matrix returneres en liste eller matrix med hvert element divideret med 100.

**Bemærk:** Sådan gennemtvinges et tilnærmet resultat,

**Håndholdt:** Tryk på .

**Windows®:** Tryk **Ctrl+Enter**.

**Macintosh®:** Tryk på **⌘+Enter**.

**iPad®:** Hold **ENTER** nede, og vælg .

---

13%	0.13
-----	------

---

---

{1,10,100}%	{0.01,0.1,1.}
-------------	---------------

---

## = (lig med)

-tast

*Udtr1 = Udtr2* ⇒ *Boolsk udtryk*

*Liste1 = Liste2* ⇒ *Boolsk liste*

*Matrix1 = Matrix2* ⇒ *Boolsk matrix*

Returnerer sand, hvis *Udtr1* bestemmes til at være lig med *Udtr2*.

Returnerer false, hvis *Udtr1* bestemmes til at være forskellig fra *Udtr2*.

Alt andet returnerer en reduceret form af ligningen.

For lister og matricer returneres sammenligninger element for element.

Eksempelfunktion, der anvender 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
```

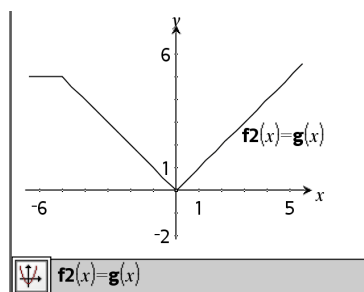
*Done*

## = (lig med)



 -tast

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

Tegnet resultat af grafen  $g(x)$



## ≠ (forskellig fra)

  -taster

$Udtr1 \neq Udtr2 \Rightarrow$  Boolsk udtryk

Se "=" (lig med)-eksemplet.

$Liste1 \neq Liste2 \Rightarrow$  Boolsk liste

$Matrix1 \neq Matrix2 \Rightarrow$  Boolsk matrix

Returnerer true, hvis  $Udtr1$  bestemmes til at være forskellig fra  $Udtr2$ .



Returnerer false, hvis  $Udtr1$  bestemmes til at være lig med  $Udtr2$ .

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

**Bemærk:** Du kan indsætte denne operator fra tastaturet ved at skrive  $\neq$

## < (mindre end)

  taster

$Udtr1 < Udtr2 \Rightarrow$  Boolsk udtryk



Se "=" (lig med)-eksemplet.

$Liste1 < Liste2 \Rightarrow$  Boolsk liste

$Matrix1 < Matrix2 \Rightarrow$  Boolsk matrix

Returnerer true, hvis  $Udtr1$  bestemmes til at være mindre end  $Udtr2$ .

## < (mindre end)



  taster

Returnerer false, hvis *Udtr1* bestemmes til at være større end *Udtr2*.

Alt andet returnerer en reduceret form af ligningen.

For lister og matricer returneres sammenligninger element for element.

## ≤ (mindre end eller lig med)

  taster

$Udtr1 \leq Udtr2 \Rightarrow$  *Boolsk udtryk*

Se "=" (lig med)-eksemplet.

$Liste1 \leq Liste2 \Rightarrow$  *Boolsk liste*

$Matrix1 \leq Matrix2 \Rightarrow$  *Boolsk matrix*

Returnerer true, hvis *Udtr1* bestemmes til at være mindre end eller lig med *Udtr2*.



Returnerer false, hvis *Udtr1* bestemmes til at være større end eller lig med *Udtr2*.

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

**Bemærk:** Du kan indsætte denne operator fra tastaturet ved at skrive <=

## > (større end)

  taster

$Udtr1 > Udtr2 \Rightarrow$  *Boolsk udtryk*

Se "=" (lig med)-eksemplet.

$Liste1 > Liste2 \Rightarrow$  *Boolsk liste*



$Matrix1 > Matrix2 \Rightarrow$  *Boolsk matrix*

Returnerer true, hvis *Udtr1* bestemmes til at være større end *Udtr2*.

Returnerer false, hvis *Udtr1* bestemmes til at være mindre end eller lig med *Udtr2*.



Alt andet returnerer en reduceret form af uligheden.

## > (større end)

  taster

For lister og matricer returneres sammenligninger element for element.

## ≥ (større end eller lig med)

  taster

$Udtr1 \geq Udtr2 \Rightarrow$  *Boolsk udtryk*

Se "=" (lig med)-eksemplet.

$Liste1 \geq Liste2 \Rightarrow$  *Boolsk liste*

$Matrix1 \geq Matrix2 \Rightarrow$  *Boolsk matrix*

Returnerer true, hvis *Udtr1* bestemmes til at være større end eller lig med *Udtr2*.



Returnerer false, hvis *Udtr1* bestemmes til at være mindre end *Udtr2*.

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

**Bemærk:** Du kan indsætte denne operator fra tastaturet ved at skrive >=

## ⇒ (medfører)

  -taster

$BoolskUdtryk1 \Rightarrow BoolskUdtryk2$   
returnerer *Boolsk udtryk*

$5 > 3$ or $3 > 5$	true
--------------------	------

$5 > 3 \Rightarrow 3 > 5$	false
---------------------------	-------

$BoolskListe1 \Rightarrow BoolskListe2$   
returnerer *Boolsk liste*

3 or 4	7
--------	---

$BoolskMatrix1 \Rightarrow BoolskMatrix2$   
returnerer *Boolsk matrix*

$3 \Rightarrow 4$	-4
-------------------	----

$\{1,2,3\}$ or $\{3,2,1\}$	$\{3,2,3\}$
----------------------------	-------------

$\{1,2,3\} \Rightarrow \{3,2,1\}$	$\{-1,-1,-3\}$
-----------------------------------	----------------



$Heltal1 \Rightarrow Heltal2$  returnerer *Heltal*

Beregner udtrykket **not** <argument1> or <argument2> og returnerer true, false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdier element for element.

**Bemærk:** Du kan indsætte denne operator fra tastaturet ved at skrive =>

## ⇔ (ænsbetydende med)

  -taster

*BoolskUdtryk1* ⇔ *BoolskUdtryk2*  
returnerer *Boolsk udtryk*

$5 > 3 \text{ xor } 3 > 5$	true
----------------------------	------

*BoolskListe1* ⇔ *BoolskListe2*  
returnerer *Boolsk liste*

$5 > 3 \Leftrightarrow 3 > 5$	false
-------------------------------	-------

*BoolskMatrix1* ⇔ *BoolskMatrix2*  
returnerer *Boolsk matrix*

$3 \text{ xor } 4$	7
--------------------	---

$3 \Leftrightarrow 4$	-8
-----------------------	----

*Heltal1* ⇔ *Heltal2* returnerer *Heltal*

$\{1,2,3\} \text{ xor } \{3,2,1\}$	$\{2,0,2\}$
------------------------------------	-------------


$\{1,2,3\} \Leftrightarrow \{3,2,1\}$	$\{-3,-1,-3\}$
---------------------------------------	----------------

Returnerer negationen af en logisk **XOR** Boolsk operation anvendt på de to argumenter. Returnerer true eller false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdierne element for element.

**Bemærk:** Du kan indsætte denne operator fra tastaturet ved at skrive <=>

## ! (fakultet)

 -tast

*Værdi!* ⇒ *værdi*

$5!$	120
------	-----

*Liste!* ⇒ *liste*

$\{\{5,4,3\}\}!$	$\{120,24,6\}$
------------------	----------------

*Matrix!* ⇒ *matrix*

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$	$\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$
---	---

Returnerer fakultetværdien af argumentet.

For en liste eller matrix returneres en liste eller matrix af elementerne.

## & (tilføj)

  -taster

*Streng1* & *Streng2* ⇒ *streng*

"Hello "&"Nick"
-----------------

"Hello Nick"
--------------

Returnerer en tekststreng, der er *Streng2* adderet til *Streng1*.

**d(Udtr1, Var[,Orden]) |**  
*Var=Værdi⇒værdi*

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

**d(Udtr1, Var[,Orden])⇒værdi**

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

**d(1, Var[,Orden])⇒liste**

$$x:=3: \frac{d}{dx}(\{x^2, x^3, x^4\}) \quad \{6, 27, 108\}$$

**d(Matrix1, Var[,Orden])⇒matrix**

Med undtagelse af, når første syntaks anvendes, skal den numeriske værdi lagres i en variabel *Var*, før der beregnes på **d()**. Se eksemplerne.

**d()** kan anvendes til beregning af differentialkvotienter af første og anden orden numerisk i et punkt ved hjælp af automatiske differentiationsmetoder.

Hvis *Orden* medtages, skal den være =1 eller 2. Standardindstillingen er 1.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **derivative (...)**.

**Bemærk:** Se også Differentialkvotient af første orden, side 5 eller Differentialkvotient af anden orden, side 6.

**Bemærk:** Algoritmen **d()** har en begrænsning: Den arbejder sig rekursivt gennem det uforkortede udtryk og beregner den numeriske værdi af differentialkvotienten af første orden (og anden, hvis relevant) af hvert deludtryk, hvilket kan føre til et utilsigtet resultat.

Betragt eksemplet til højre. Differentialkvotienten af første orden af  $x \cdot (x^2+x)^{1/3}$  at  $x=0$  er lig med 0. Man da differentialkvotienten af underudtrykket  $(x^2+x)^{1/3}$  ikke er defineret i  $x=0$ , og denne værdi anvendes til at beregne differentialkvotienten af det samlede udtryk, rapporterer **d()** resultatet som udefineret og viser en advarsel.

$$\frac{d}{dx} \left( x \cdot (x^2+x)^{\frac{1}{3}} \right) |_{x=0} \quad \text{undef}$$


---


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


---

0.000033



## d() (differentialkvotient)

Katalog > 

Hvis du rammer denne begrænsning, skal du kontrollere løsningen grafisk. Du kan også prøve med **centralDiff()**.

## ∫() (integral)

Katalog > 

∫(*Udtr1*, *Var*, *Nedre*, *Øvre*) ⇒ værdi



Returnerer integralet af *Udtr1* med hensyn til variabelen *var* fra *Nedre* til *Øvre*. Kan anvendes til at beregne det bestemte integral numerisk med samme metode som **nInt()**.

$\int_0^1 x^2 dx$	0.333333
-------------------	----------

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **integral (...)**.

**Bemærk:** Se også **nInt()**, side 107, og **Bestemt integralskabelon**, side 6.

## √() (kvadratrod)

  -taster

√(*Værdi1*) ⇒ værdi

$\sqrt{4}$	2
------------	---

√(*Liste1*) ⇒ liste

$\sqrt{\{9,2,4\}}$	{3,1.41421,2}
--------------------	---------------

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **sqrt (...)**

**Bemærk:** Se også **Kvadratrods-kabelon**, side 1.

## $\Pi()$ (prodSeq)

Katalog >  $\Pi(Udtr1, Var, Lav, Høj) \Rightarrow udtryk$ 

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **prodSeq (...)**.

Beregn  $Udtr1$  for hver værdi af  $Var$  fra  $Lav$  til  $Høj$  og returnerer produktet af resultaterne.

**Bemærk:** Se også **Produktskabelon (II)**, side 5.

 $\Pi(Udtr1, Var, Lav, Lav-1) \Rightarrow 1$  $\Pi(Udtr1, Var, Lav, Høj) \Rightarrow 1/\Pi(Udtr1, Var, Høj+1, Lav-1)$  hvis  $Høj < Lav-1$ 

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

$$\prod_{n=1}^5 \left\{ \left\{ \frac{1}{n}, n, 2 \right\} \right\} \quad \left\{ \frac{1}{120}, 120, 32 \right\}$$

$$\prod_{k=4}^3 (k) \quad 1$$

De anvendte produktformler stammer fra følgende reference:

Ronald L. Graham, Donald E. Knuth og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \quad 6$$

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \cdot \prod_{k=2}^4 \left(\frac{1}{k}\right) \quad \frac{1}{4}$$

## $\Sigma()$ (sumSeq)

Katalog >  $\Sigma(Udtr1, Var, Lav, Høj) \Rightarrow udtryk$ 

**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **sumSeq (...)**.

Beregn  $Udtr1$  for hver værdi af  $Var$  fra  $Lav$  til  $Høj$  og returnerer summen af resultaterne.

**Bemærk:** Se også **Sumskabelon**, side 5.

 $\Sigma(Udtr1, Var, Lav, Lav-1) \Rightarrow 0$  $\Sigma(Udtr1, Var, Lav, Høj) \Rightarrow -\Sigma(Udtr1, Var, Høj+1, Lav-1)$  hvis  $Høj < Lav-1$ 

$$\sum_{n=1}^5 \left(\frac{1}{n}\right) \quad \frac{137}{60}$$

$$\sum_{k=4}^3 (k) \quad 0$$

De anvendte summationsformler stammer fra følgende reference:

Ronald L. Graham, Donald E. Knuth og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\sum_{k=4}^1 (k) \quad -5$$


---


$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) \quad 4$$

 $\Sigma\text{Int}()$ 

$\Sigma\text{Int}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [afrundVærdi]) \Rightarrow værdi$

$$\Sigma\text{Int}(1,3,12,4,75,20000,,,12,12) \quad -218.11$$

 $\Sigma\text{Int}$ 

$(NPmt1, NPmt2, amortTabel) \Rightarrow værdi$

Amortiseringsfunktion, der beregner summen af renter under en angivet række af betalinger.

$NPmt1$  og  $NPmt2$  definerer starten og slutningen af betalingsrækken.

$N, I, PV, Pmt, FV, PpY, CpY$  og  $PmtAt$  er beskrevet i tabellen over TVM-argumenter på side 171.

- Hvis du udelader  $Pmt$ , sættes den som standard til  $Pmt = \text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ .
- Hvis du udelader  $FV$ , sættes den som standard til  $FV = 0$ .
- Standarderne for  $PpY, CpY$  og  $PmtAt$  er de samme som for TVM-funktionerne.

$afrundVærdi$  angiver antallet af decimaler til afrunding.  
Standardværdi=2.

$\Sigma\text{Int}(NPmt1, NPmt2, amortTabel)$  beregner summen af renten baseret på amortiseringstabellen  $amortTabel$ .  $amortTabel$ -argumentet skal være en matrix i formen beskrevet under  $\text{amortTbl}()$ , side 7.

$tbl := \text{amortTbl}(12, 12, 4, 75, 20000, ,, 12, 12)$			
0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

$$\Sigma\text{Int}(1,3,tbl) \quad -218.11$$

**Bemærk:** Se også  $\Sigma\text{Prn}()$  nedenfor og **Bal()**, side 16.

 $\Sigma\text{Prn}()$ 

$\Sigma\text{Prn}(\text{NPmt1}, \text{NPmt2}, N, I, PV, [\text{Pmt}], [\text{FV}], [\text{PpY}], [\text{CpY}], [\text{PmtAt}], [\text{afrundVærdi}]) \Rightarrow \text{værdi}$

$\Sigma\text{Prn}(1,3,12,4.75,20000,,12,12)$  -4911.47

 $\Sigma\text{Prn}$ 

$(\text{NPmt1}, \text{NPmt2}, \text{amortTabel}) \Rightarrow \text{værdi}$

Amortiseringsfunktion, der beregner summen af afdraget under en angivet række af betalinger.

$\text{NPmt1}$  og  $\text{NPmt2}$  definerer starten og slutningen af betalingsrækken.

$N, I, PV, Pmt, FV, PpY, CpY$  og  $PmtAt$  er beskrevet i tabellen over TVM-argumenter på side 171.

- Hvis du udelader  $Pmt$ , sættes den som standard til  $Pmt=\text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ .
- Hvis du udelader  $FV$ , sættes den som standard til  $FV=0$ .
- Standardværdierne for  $PpY, CpY$  og  $PmtAt$  er de samme som for TVM-funktionerne.

$\text{afrundVærdi}$  angiver antallet af decimaler til afrunding.  
Standardværdi=2.

$\Sigma\text{Prn}(\text{NPmt1}, \text{NPmt2}, \text{amortTabel})$  beregner summen af afdrag på hovedstolen baseret på amortiseringstabellen  $\text{amortTabel}$ .  $\text{amortTabel}$ -argumentet skal være en matrix i formen beskrevet under **amortTbl()**, side 7.

**Bemærk:** Se også  $\Sigma\text{Int}()$  ovenfor og **Bal()**, side 16.

$\text{tbl}:=\text{amortTbl}(12,12,4.75,20000,,12,12)$			
0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

$\Sigma\text{Prn}(1,3,\text{tbl})$  -4911.47

## # (henvisning)

  -taster

# *varNavnestreng*

<code>xyz:=12</code>	12
<code>#{"x"&amp;"y"&amp;"z"}</code>	12

Kalder variablen, hvis navn er *varNavnestreng*. Dermed kan du anvende strenge til at oprette variabelnavne fra en funktion.

Opretter eller kalder variablen xyz.

<code>10→r</code>	10
<code>"r"→s1</code>	"r"
<code>#s1</code>	10

Returnerer værdien af variablen (r), hvis navn er lagret i variablen s1.

## E (videnskabelig notation)

 -tast

*mantisseEeksponent*

23000.	23000.
2300000000.+4.1E15	4.1E15
$3 \cdot 10^4$	30000

Indtaster et tal i videnskabelig notation. Tallet fortolkes som *mantisse* × 10<sup>eksponent</sup>.

Tip: Hvis du vil indtaste en 10. potens uden at give et decimalværdiresultat, skal du bruge 10<sup>^heltal</sup>.

**Bemærk:** Du kan indsætte denne operator fra computerens tastatur ved at skrive @E. Skriv for eksempel 2.3@E4 for at indtaste 2.3E4.

## g (nygrader)

 -tast

*Udtr1g⇒udtryk*

I vinkeltilstanden Grader, Nygrader eller Radianer:

*Udtr1g⇒udtryk*

<code>cos(50<sup>g</sup>)</code>	0.707107
<code>cos({0,100<sup>g</sup>,200<sup>g</sup>})</code>	{1,0.,-1.}

*Liste1g⇒liste*

*Matrix1g⇒matrix*

Denne funktion giver mulighed for at angive en vinkel i nygrader, mens programmet er i vinkeltilstanden Grader eller Radianer.

## g (nygrader)

1-tast

I vinkeltilstanden Radian ganges *Udtr1* med  $\pi/200$ .

I vinkeltilstanden Grader ganges *Udtr1* med  $g/100$ .

I vinkeltilstanden Nygrader returneres *Udtr1* uændret.

**Bemærk:** Du kan indsætte dette symbol fra computerens tastatur ved at skrive @g.

## r (radian)

1-tast

Værdi  $I^r \Rightarrow værdi$

Liste  $I^r \Rightarrow liste$

Matrix  $I^r \Rightarrow matrix$

Denne funktion giver mulighed for at angive en vinkel i radianer, mens programmet er i vinkeltilstanden Grader eller Nygrader.

I vinkeltilstanden Grader ganges argumentet med  $180/\pi$ .

I vinkeltilstanden Radian, returneres argument uændret.

I vinkeltilstanden Nygrader ganges argumentet med  $200/\pi$ .

Tip: Anvend  $r$ , hvis du vil fremtvinge radianer i en funktionsdefinition uanset den aktuelle tilstand, når funktionen anvendes.

**Bemærk:** Du kan indsætte dette symbol fra computerens tastatur ved at skrive @r.

I vinkeltilstanden Grader, Nygrader eller Radianer:

$\cos\left(\frac{\pi}{4^r}\right)$	0.707107
$\cos\left(\left\{0^r, \left(\frac{\pi}{12}\right)^r, -(\pi)^r\right\}\right)$	{1,0.965926,-1.}

## ° (grader)

1-tast

Værdi  $I^\circ \Rightarrow værdi$

Liste  $I^\circ \Rightarrow liste$

I vinkeltilstanden Grader, Nygrader eller Radianer:

## ° (grader)

1-tast

$MatrixI^{\circ} \Rightarrow matrix$

$\cos(45^{\circ})$  0.707107

Denne funktion giver mulighed for at angive en vinkel i grader, mens programmet er i vinkeltilstanden Nygrader eller Radianer.

I vinkeltilstanden Radianer:

I vinkeltilstanden Radian ganges argumentet med  $\pi/180$ .

I vinkeltilstanden grader returneres argument uændret.

I vinkeltilstanden Nygrader ganges argumentet med 10/9.

**Bemærk:** Du kan indsætte dette symbol fra computerens tastatur ved at skrive @d.

## °, ', " (grader/minutter/sekunder)

ctrl  -taster

$gg^{\circ}mm'ss.ss'' \Rightarrow udtryk$

I vinkeltilstanden Grader:

ggEt positivt eller negativt tal

$25^{\circ}13'17.5''$  25.2215

mmEt ikke-negativt tal

$25^{\circ}30'$   $\frac{51}{2}$

ss.ssEt ikke-negativt tal


Returnerer  $gg+(mm/60)+(ss.ss/3600)$ .

Med denne indtastning i 60-talsformat kan du:

- Indtaste en vinkel i grader/minutter/sekunder uden hensyn til den aktuelle vinkeltilstand.
- Indtaste tiden som timer/minutter/sekunder.

**Bemærk:** Efterfølg ss.ss med to apostroffer (""), ikke et citationstegn (").

## ∠ (vinkel)


ctrl  -taster

$[Radius, \angle \theta \ Vinkel] \Rightarrow vektor$   
(polært input)

I vinkeltilstanden Radian og vektorformatet indstillet til:

rektangulær

## ∠ (vinkel)

ctrl  -taster

[Radius,∠θ\_Vinkel,Z\_Koordinat]  
]⇒vektor  
(cylindrisk input)

[5 ∠60° ∠45°]  
[1.76777 3.06186 3.53553]

[Radius,∠θ\_Vinkel,∠θ\_Vinkel]  
]⇒vektor  
(sfærisk input)

cylindrisk

Returnerer koordinater som en vektor afhængigt af tilstandsindstillingen for vektorformat: rektangulær, cylindrisk eller sfærisk.

[5 ∠60° ∠45°]  
[3.53553 ∠1.0472 3.53553]

**Bemærk:** Du kan indsætte dette symbol fra computerens tastatur ved at skrive @<.

sfærisk

[5 ∠60° ∠45°]  
[5. ∠1.0472 ∠0.785398]

(Størrelse ∠ Vinkel)⇒kompleksVærdi  
(polært input)

I vinkeltilstanden radian og rektangulært komplekst format:

Indtaster en kompleks værdi i den polære form ( $r\angle\theta$ ). *Vinkel* fortolkes efter den aktuelle indstilling af vinkeltilstand.

$5+3\cdot i-\left(10\angle\frac{\pi}{4}\right)$  -2.07107-4.07107·i

$5+3\cdot i-\left(10\angle\frac{\pi}{4}\right)$  -2.07107-4.07107·i

## \_ (understregning som et tomt element)

Se |“Tomme (ugyldige) elementer” side 224, .

## \_ (understregning som en enhedsbenævnelse)

ctrl  -taster

Variabel\_

Det antages, at z er udefineret:

Når *Variabel* ikke har nogen værdi, behandles den, som om den repræsenterer et komplekst tal. Uden\_ behandles variabelen som reel.

real(z)	z
real(z_)	real(z_)
imag(z)	0
imag(z_)	imag(z_)

Hvis *Variabel* har en værdi, ignoreres \_ og *Variabel* bevarer sin oprindelige datatype.



## 10<sup>^</sup>()

Katalog > 

10<sup>^</sup> (Værdi) ⇒ værdi

 $10^{1.5}$ 

31.6228

10<sup>^</sup> (Liste1) ⇒ liste

Returnerer 10 opløftet til potensen af argumentet.

I en liste returneres 10 opløftet til potensen af elementerne i *Liste1*.

10<sup>^</sup>(kvadratMatrix1) ⇒ kvadratmatrix

Returnerer 10 opløftet til potensen af *kvadratMatrix1*. Dette er ikke det samme som at beregne 10 opløftet til potensen af hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} = \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

*KvadratMatrix1* skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

## <sup>^-1</sup>(reciprok)

Katalog > 

Værdi <sup>^-1</sup> ⇒ værdi

 $(3.1)^{-1}$ 

0.322581

Liste1 <sup>^-1</sup> ⇒ liste

Returnerer den reciprokke værdi af argumentet.

For en liste returneres de reciprokke værdier af elementerne i *Liste1*.

*kvadratMatrix1* <sup>^-1</sup> ⇒ kvadratMatrix

Returnerer den inverse værdi af *kvadratMatrix1*.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

*kvadratMatrix1* skal være en ikke-singulær kvadratisk matrix.

## | (betingelses-operator)

ctrl  -taster

Udtryk | BoolskUdtryk1  
[andBoolskUdtryk2]...

 $x+1|x=3$ 

4

 $x+55|x=\sin(55)$ 

54.0002

Udtryk | BoolskUdtryk1  
[orBoolskUdtryk2]...

("|")-betingelses-tegnet fungerer som en binær operator. Argumentet til venstre for | er et udtryk. Argumentet til højre for | angiver en eller flere betingelser, der skal tages hensyn til i reduktionen af udtrykket. Hvis der er flere betingelser efter |, skal de samles med en logisk "and" eller "or"-kommando.

Betingelses-operatoren bruges på tre grundlæggende måder:

- Substitutioner
- Intervalafgrænsninger
- Udelukkelse

Substitutioner er i form af en ligning, som  $x=3$  eller  $y=\sin(x)$ . For at være så effektiv som muligt skal venstre side af betingelsen være en simpel variabel.

*Udtryk | Variabel = værdi* vil substituere *værdi* for enhver forekomst af *Variabel* i *Udtryk*.

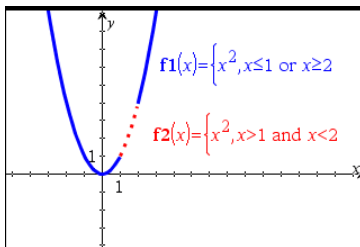
Intervalafgrænsning antager form af en eller flere uligheder samlet af logiske "and" eller "or"-kommandoer.

Intervalafgrænsninger tillader også reduktioner, der ellers ville være ugyldige eller ikke kunne beregnes.

Udelukkelse benytter "forskellig fra"-kommandoen ( $\neq$  eller  $\neq$ ) til at udelukke en specifik værdi fra reduktionen.

$x^3-2\cdot x+7 \rightarrow f(x)$	Done
$f(x) x=\sqrt{5}$	8.73205

$nSolve(x^3+2\cdot x^2-15\cdot x=0,x)$	0.
$nSolve(x^3+2\cdot x^2-15\cdot x=0,x) x>0 \text{ and } x<5$	3.



## → (lagring)

ctrl var -tast

Værdi → *Var*

$$\frac{\pi}{4} \rightarrow \text{myvar} \quad 0.785398$$

Liste → *Var*

$$2 \cdot \cos(x) \rightarrow yI(x) \quad \text{Done}$$

Matrix → *Var*

$$\{1,2,3,4\} \rightarrow \text{lst5} \quad \{1,2,3,4\}$$

Udtr → *Funktion*(*Param1*,...)

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

List → *Funktion*(*Param1*,...)

$$\text{"Hello"} \rightarrow \text{str1} \quad \text{"Hello"}$$

Matrix → *Funktion*(*Param1*,...)

Hvis variabelen *Var* ikke findes, oprettes *Var* og initialiseres til *Værdi*, *Liste* eller *Matrix*.

Hvis *Var* findes i forvejen og ikke er låst eller beskyttet, erstattes dens indhold med *Værdi*, *Liste* eller *Matrix*.

**Bemærk:** Du kan indsætte denne operator fra tastaturet ved at skrive =: som en genvej. Skriv for eksempel **pi/4 =: myvar**.

## := (tildel)

ctrl := -taster

*Var* := *Værdi*

$$\text{myvar} := \frac{\pi}{4} \quad .785398$$

*Var* := *Liste*

$$yI(x) := 2 \cdot \cos(x) \quad \text{Done}$$

*Var* := *Matrix*

$$\text{lst5} := \{1,2,3,4\} \quad \{1,2,3,4\}$$

*Funktion*(*Param1*,...) := *Udtr*

$$\text{matg} := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

*Funktion*(*Param1*,...) := *Liste*

$$\text{str1} := \text{"Hello"} \quad \text{"Hello"}$$

*Funktion*(*Param1*,...) := *Matrix*

Hvis variabelen *Var* ikke findes, oprettes *Var* og initialiseres til *Værdi*, *Liste* eller *Matrix*.

Hvis *Var* findes i forvejen og ikke er låst eller beskyttet, erstattes dens indhold med *Værdi*, *Liste* eller *Matrix*.

## © [tekst]

© behandler *tekst* som en kommentarlinje, så du kan skrive kommentarer til funktioner og programmer, du opretter.

© kan være i starten eller overalt i linjen. Alt til højre for © til linjens slutning er kommentaren.

**Bemærk indtastning af eksemplet:** For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.

---

 Define  $g(n)=$ Func

© Declare variables

Local  $i,result$  $result:=0$ For  $i,1,n,1$  ©Loop  $n$  times $result:=result+i^2$ 

EndFor

Return  $result$ 

EndFunc

---

*Done*

---

 $g(3)$ 

---

14

## Ob, Oh

Ob *binært*Tal

I decimal tilstand:

---

0b10+0hF+10

27

Oh *hexadecimal*Tal

I binær tilstand:

---

0b10+0hF+10

0b11011

Betegner henholdsvis binært eller hexadecimalt tal. For at indtaste et binært eller hexadecimalt tal skal du indtaste præfikset Ob eller Oh uanset talsystemet. Uden præfiks behandles tallet som decimaltal (10 talsystem).

I hexadecimal tilstand:

---

0b10+0hF+10

0h1B

# TI-Nspire™ CX II - Tegn kommandoer

Dette er et supplerende dokument til TI-Nspire™ Referencevejledningen og TI-Nspire™ CAS Referencevejledningen. Alle TI-Nspire™ CX II-kommandoer vil blive inkorporerede og offentliggjort i version 5.1 i TI-Nspire™ Referencevejledningen og TI-Nspire™ CAS Referencevejledningen.

## Grafikprogrammering

Nye kommandoer er blevet tilføjet på TI-Nspire™ CX II håndholdte og TI-Nspire™ desktop applikationer til grafisk programmering.

TI-Nspire™ CX II håndholdte skifter til denne grafiktilstand, mens du udfører grafikkommandoer og skifter tilbage til den kontekst, hvor programmet blev udført efter programmets afslutning.

Skærmen viser "Running ..." i øverste linje mens programmet udføres. Det vil vise "Finished", når programmet fuldendes. Enhver tastetryk vil tage systemet ud af grafikfunktionen.

- Overgangen til grafiktilstand udløses automatisk, når en af Draw (grafik) - kommandoerne opstår under udførelsen af TI-Basic-programmet.
- Denne overgang vil kun ske, når der udføres et program fra lommeregneren; i et dokument eller en lommeregner i scratchpad.
- Overgangen fra grafiktilstand sker ved afslutning af programmet.
- Grafiktilstanden er kun tilgængelig på visninger på TI-Nspire™ CX II håndholdte og TI-Nspire™ CX II håndholdte. Det betyder, at den ikke er tilgængelig i computerdokumentvisningen på skrivebordet eller iOS.
  - Hvis der opstår en grafikkommando under udførelsen af et TI-Basic-program fra den forkerte kontekst, vises en fejlmeddelelse, og TI-Basic-programmet afsluttes.

## Grafikskærm

Grafikskærmen vil indeholde en overskrift øverst på skærmen, der ikke kan skrives til af grafikkommandoer.

Grafikskærmens tegningsområde bliver ryddet (farve = 255.255.255), når grafikskærmen initialiseres.

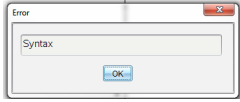
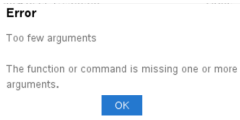
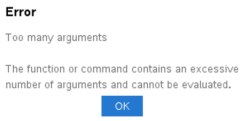
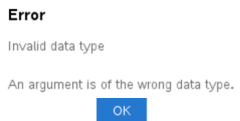
Grafikskærm	Standard
Højde	212
Bredde	318
Farve	hvid: 255,255,255

## Standardvisning og indstillinger

- Statusikonerne i øverste bjælke (batteristatus, tryk-til-test-status, netværksindikator osv.) vises ikke, mens et grafikprogram kører.
- Standard tegningsfarve: Sort (0,0,0)
- Standard pennefacon - normal, glat
  - Tykkelse: 1 (tynd), 2 (normal), 3 (tykkest)
  - Typografi 1 (glat), 2 (punkteret), 3 (stiplet)
- Alle tegningskommandoer bruger de nuværende indstillinger for farve og pen enten standardværdier eller dem, der blev indstillet via TI-Basic-kommandoer.
- Tekstens skrifttype er fast og kan ikke ændres.
- Enhver udgang til grafikskærmen tegnes inden for et klipvindue, som er størrelsen på grafikskærmens tegneområde. Enhver tegnet output, der strækker sig uden for dette klippede grafiske skærmområde, tegnes ikke. Der vises ingen fejlmeddelelse.
- Alle x, y-kordinater, der er angivet til tegningskommandoer, er defineret således, at 0,0 er øverst til venstre i grafikskærmens tegneområde.
  - **Undtagelser:**
    - **DrawText** bruger koordinaterne som nederste venstre hjørne af grænsekassen til teksten.
    - **SetWindow** bruger nederste venstre hjørne af skærmen
- Alle parametre for kommandoerne kan gives som udtryk, der evalueres til et tal, som derefter afrundes til nærmeste heltal.

## Fejlmeddelelser på grafikskærmen

Hvis validering mislykkes, vises en fejlmeddelelse.

Fejlmeddelelse	Beskrivelse	Vis
Fejl Syntaks	Hvis syntakskontrollen finder syntaksfejl, viser den en fejlmeddelelse, og prøver at placere markøren tæt på den første fejl, så du kan rette den.	
Fejl For få argumenter	Funktionen eller kommandoen mangler et eller flere argumenter	
Fejl For mange argumenter	Funktionen eller kommandoen indeholder for mange argumenter og kan ikke beregnes.	
Fejl Ugyldig datatype	Et argument er af forkert datatype.	

## Ugyldige kommandoer i grafiktilstand

Nogle kommandoer er ikke tilladt, når programmet skifter til grafiktilstand. Hvis disse kommandoer opstår i grafiktilstand og fejl vises, og programmet afsluttes.

Ikke-tilladt kommando	Fejlmeddelelse
Anmodning	Anmodning kan ikke udføres i grafiktilstand
RequestStr	RequestStr kan ikke udføres i grafiktilstand
tekst	Tekst kan ikke udføres i grafiktilstand

Kommandoerne, der udskriver tekst til lommeregneren - **disp** and **dispAt** - vil være understøttede kommandoer i grafiktilstanden. Teksten fra disse kommandoer vil blive sendt til lommeregnerskærmen (ikke på grafik) og vil være synlig, når programmet udgår, og systemet skifter tilbage til Lommeregner-appen

**Ryd (Clear)**Katalog >   
CXII**Ryd** *x, y, bredde, højde*

Ryd

Rydder hele skærmen, hvis der ikke er angivet nogen parametre.

Rydder hele skærmen

Hvis *x, y, bredde* og *højde* er angivet, vil rektanglet der defineres af parametrene vil blive ryddet.

Ryd 10,10,100,50

Rydder et rektangelområde med øverste venstre hjørne på (10, 10) og med bredde 100, højde 50



**DrawArc**
 Katalog >   
**CXII**

**DrawArc**  $x, y, \text{bredde}, \text{højde}, \text{startAngle}, \text{arcAngle}$

Tegner en bue i det afgrænsende rektangel med de tilvejebragte start- og buevinkler.

$x, y$ : øverste venstre koordinat af afgrænsende rektangel

$\text{bredde}, \text{højde}$ : dimensioner af et afgrænset rektangel

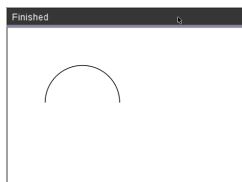
"Buevinklen" definerer buens størrelse.

Disse parametre kan gives som udtryk, der evalueres til et tal, som derefter afrundes til nærmeste heltal.

DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180



Se også: [FillArc](#)

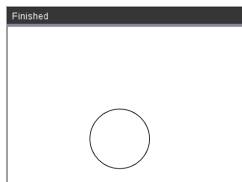
**DrawCircle**
 Katalog >   
**CXII**

**DrawCircle**  $x, y, \text{radius}$

$x, y$ : koordinat af centrum

$\text{radius}$ : cirkelns radius

DrawCircle 150,150,40



Se også: [FillCircle](#)

## DrawLine

Katalog >   
CXII

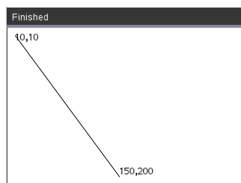
**DrawLine**  $x1, y1, x2, y2$

Tegn en linje fra  $x1, y1, x2, y2$ .

Udtryk der evalueres til et tal, som derefter afrundes til nærmeste heltal.

**Skærmgrænser:** Hvis de angivne koordinater forårsager, at en del af linjen tegnes uden for grafikskærmen, bliver den del af linjen klippet, og der vises ingen fejlmeddelelse.

DrawLine 10,10,150,200



## DrawPoly

Katalog >   
CXII

Kommandoerne har to varianter:

**DrawPoly**  $xlist, ylist$

or

**DrawPoly**  $x1, y1, x2, y2, x3, y3...xn, yn$

**Bemærk:** DrawPoly  $xlist, ylist$

Form vil forbinde  $x1, y1$  til  $x2, y2, x2, y2$  til  $x3, y3$  og så videre.

**Bemærk:** DrawPoly  $x1, y1, x2, y2, x3, y3...xn, yn$

$xn, yn$  vil **IKKE** blive automatisk forbundet med  $x1, y1$ .

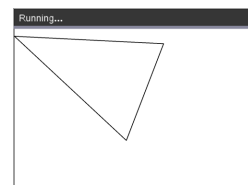
Udtryk der evalueres til en liste over reelle floats  
 $xlist, ylist$

Udtryk der evalueres til en enkelt reel float  
 $x1, y1...xn, yn$  = koordinater for polygonens hjørner

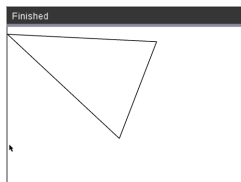
$xlist:={0,200,150,0}$

$ylist:={10,20,150,10}$

DrawPoly  $xlist,ylist$



DrawPoly 0,10,200,20,150,150,0,10



**Bemærk: DrawPoly:** Indtast størrelsesdimensioner (bredde/højde) i forhold til tegnede linjer. Linjerne er tegnet i en afgrænsningskasse omkring det angivne koordinat og dimensioner således, at den faktiske størrelse af den tegnede polygon vil være større end bredden og højden.

Se også: [FillPoly](#)

## DrawRect

**DrawRect** *x, y, bredde, højde*

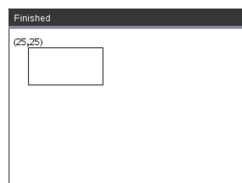
*x, y*: øverste venstre koordinat af rektangel

*bredde, højde*: bredde og højde af rektangel (rektangel trukket ned og til højre fra startkoordinat).

**Bemærk:** Linjerne er tegnet i en afgrænsningskasse omkring den angivne koordinat og dimensioner således, at den faktiske størrelse af det tegnede rektangel vil være større end bredden og højden der er indikeret.

Se også: [FillRect](#)

DrawRect 25,25,100,50



## DrawText

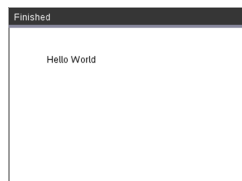
**DrawText** *x, y, exprOrString1*  
*[,exprOrString2]...*

*x, y*: koordinat af tekstoutput

Tegner teksten i *exprOrString* ved den specificerede *x, y* koordinatplacering.

Reglerne for *exprOrString* er de samme som for **Disp** – **DrawText** kan tage flere argumenter.

DrawText 50,50,"Hello World"



## FillArc

 Katalog >  CXII

**FillArc**  $x, y$ , *bredde*, *højde* *startAngle*, *arcAngle*

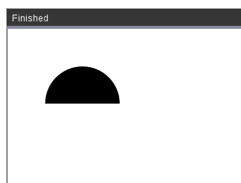
$x, y$ : øverste venstre koordinat af afgrænsende rektangel

Tegn og fyld en bue i det afgrænsende rektangel med de tilvejebragte start- og buevinkler.

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen

"Buevinklen" definerer buens svej.

FillArc 50,50,100,100,0,180



## FillCircle

 Katalog >  CXII

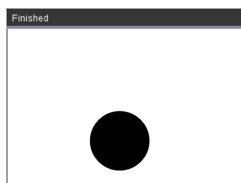
**FillCircle**  $x, y$ , *radius*

$x, y$ : koordinat af centrum

Tegn og fyld en cirkel på det specificerede center med den angivne radius.

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen.

FillCircle 150,150,40



Her!

## FillPoly

 Katalog >  CXII

**FillPoly**  $xlist, ylist$

or

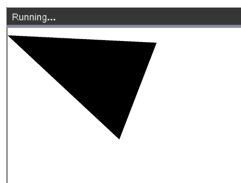
**FillPoly**  $x1, y1, x2, y2, x3, y3...xn, yn$

**Bemærk:** Linjen og farven er angivet af [SetColor](#) og [SetPen](#)

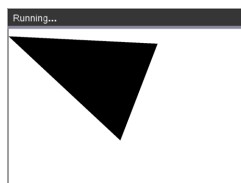
$xlist:={0,200,150,0}$

$ylist:={10,20,150,10}$

FillPoly  $xlist,ylist$



FillPoly 0,10,200,20,150,150,0,10

**FillRect****FillRect**  $x, y, \text{bredde}, \text{højde}$  $x, y$ : øverste venstre koordinat af rektangel $\text{bredde}, \text{højde}$ : bredde og højde af rektangel

Tegn og udfyld et rektangel med øverste venstre hjørne ved koordinaten angivet af  $(x,y)$

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen

**Bemærk:** Linjen og farven er angivet af [SetColor](#) og [SetPen](#)

FillRect 25,25,100,50



## G

### getPlatform()

Katalog >   
CXII

#### getPlatform()

getPlatform()

"dt"

Returnerer:

"dt" på desktop-softwareapplikationer

"hh" på TI-Nspire™ CX håndholdte

"ios" på TI-Nspire™ CX iPad® app

**PaintBuffer**

Grafisk buffer til skærmen

Denne kommando bruges sammen med UseBuffer for at øge visningshastighed på skærmen, når programmet genererer flere grafiske objekter.

**UseBuffer**

```
For n,1,10
```

```
x:=randInt(0,300)
```

```
y:=randInt(0,200)
```

```
radius:=randInt(10,50)
```

```
Wait 0,5
```

```
DrawCircle x,y,radius
```

```
EndFor
```

**PaintBuffer**

Dette program viser alle de 10 cirkler på én gang.

Hvis kommandoen "UseBuffer" fjernes, vises hver cirkel, som den er tegnet.

Se også: [UseBuffer](#)

**PlotXY**  $x, y, shape$ 

$x, y$ : koordinat til at tegne form

$shape$  : et tal mellem 1 og 13, der angiver formen

- 1 - Udfyldt cirkel
- 2 - Tom cirkel
- 3 - Udfyldt firkant
- 4 - Tom firkant
- 5 - Kryds
- 6 - Plus
- 7 - Tynd
- 8 - medium punkt, udfyldt
- 9 - medium punkt, tomt
- 10 - større punkt, udfyldt
- 11 - større punkt, tomt
- 12 - største punkt, udfyldt
- 13 - største punkt, tomt

PlotXY 100,100,1

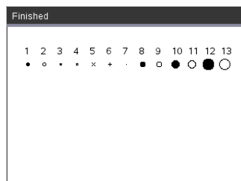


For n,1,13

DrawText 1+22\*n,40,n

PlotXY 5+22\*n,50,n

EndFor





**SetColor**
 Katalog >   
**CXII**
**SetColor**

Rød-værdi, Grøn-værdi, Blå-værdi

Værdierne for Rød, Grøn, Blå skal være mellem 0 og 255

Indstiller farven til efterfølgende tegningskommandoer

SetColor 255,0,0

DrawCircle 150,150,100

**SetPen**
 Katalog >   
**CXII**
**SetPen**

tykkelse, stil

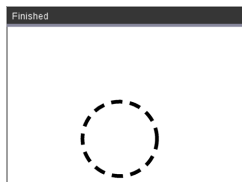
tykkelse: 1 &lt;= tykkelse &lt;= 3 | 1 er tyndest, 3 er tykkest

stil: 1 = Glat, 2 = Punkteret, 3 = Stiplet

Indstiller pennefaconen til efterfølgende tegningskommandoer

SetPen 3,3

DrawCircle 150,150,50

**SetWindow**
 Katalog >   
**CXII**
**SetWindow**

xMin, xMax, yMin, yMax

Etablerer et logisk vindue, der er knyttet til det grafiske tegneområde. Alle parametre er påkrævede.

Hvis delen af det tegnede objekt er uden for vinduet, vil resultatet blive klippet (ikke vist), og der vises ingen fejlmeddelelse.

SetWindow 0,160,0,120

vil indstille outputvinduet til 0,0 i nederste venstre hjørne med en bredde på 160 og en højde på 120

DrawLine 0,0,100,100

SetWindow 0,160,0,120

SetPen 3,3

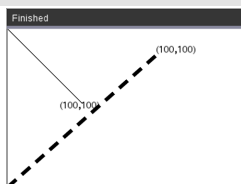
DrawLine 0,0,100,100

Hvis  $x_{\min}$  er større end eller lig med  $x_{\max}$  eller  $y_{\min}$  er større end eller lig med  $y_{\max}$ , vises en fejlmeddelelse.

Eventuelle objekter tegnet før en SetWindow-kommando vil ikke blive tegnet igen i den nye konfiguration.

For at nulstille vindueparametrene til standardværdien skal du bruge:

SetWindow 0,0,0,0



**UseBuffer**

Tegn til grafikbuffer i stedet for skærm (for at øge ydeevnen)

Denne kommando bruges sammen med PaintBuffer for at øge visningshastighed på skærmen, når programmet genererer flere grafiske objekter.

Med UseBuffer vises al grafikken først, når den næste PaintBuffer-kommando er udført.

UseBuffer skal kun kaldes en gang i programmet, dvs. hver brug af PaintBuffer behøver ikke en tilsvarende UseBuffer

Se også: [PaintBuffer](#)

## UseBuffer

```
For n,1,10
```

```
x:=randInt(0,300)
```

```
y:=randInt(0,200)
```

```
radius:=randInt(10,50)
```

```
Wait 0,5
```

```
DrawCircle x,y,radius
```

```
EndFor
```

```
PaintBuffer
```

Dette program viser alle de 10 cirkler på én gang.



Hvis kommandoen "UseBuffer" fjernes, vises hver cirkel, som den er tegnet.

## Tomme (ugyldige) elementer

Ved analyse af data fra virkeligheden har du muligvis ikke et komplet datasæt. TI-Nspire™ tillader tomme eller ugyldige dataelementer, så du kan fortsætte med det næsten komplette sæt data i stedet for at skulle starte forfra eller kassere de ukomplette forekomster.


Du kan finde et eksempel på data, der medtager tomme elementer i kapitlet Lister og regneark under "Tegning af regnearkdata".

Med funktionen **delVoid()** kan du fjerne tomme elementer fra en liste. Med funktionen **isVoid()** kan du teste for et tomt element. Nærmere beskrivelse findes under **delVoid()**, side 41, og **isVoid()**, side 78.

**Bemærk:** Du kan indtaste et tomt element manuelt i et matematisk udtryk ved at skrive "\_" eller nøgleordet **void**. Nøgleordet **void** konverteres automatisk til symbolet "\_", når udtrykket evalueres. Du skriver "\_" på den håndholdte ved at trykke  .


### Beregninger, der omfatter ugyldige elementer

De fleste beregninger med et ugyldigt input giver et ugyldigt resultat. Se de specielle tilfælde i det følgende.

	–
$\gcd(100, \_)$	–
$3 + \_$	–
$\{5, \_, 10\} - \{3, 6, 9\}$	$\{2, \_, 1\}$

### Listeargumenter med ugyldige elementer

Følgende funktioner og kommandoer ignorerer (springer over) ugyldige elementer i listeargumenter.

**count**, **countIf**, **cumulativeSum**, **freqTable** , **list**, **frequency**, **max**, **mean**, **median**, **product**, **stDevPop**, **stDevSamp**, **sum**, **sumIf**, **varPop** og **varSamp** samt regressionsberegninger, **OneVar**, **TwoVar**, og **FiveNumSummary** statistik, konfidensintervaller, og statistiske tests

$\text{sum}(\{2, \_, 3, 5, 6, 6\})$	16.6
$\text{median}(\{1, 2, \_, \_, \_, 3\})$	2
$\text{cumulativeSum}(\{1, 2, \_, 4, 5\})$	$\{1, 3, \_, 7, 12\}$
$\text{cumulativeSum} \left( \begin{bmatrix} 1 & 2 \\ 3 & \_ \\ 5 & 6 \end{bmatrix} \right)$	$\begin{bmatrix} 1 & 2 \\ 4 & \_ \\ 9 & 8 \end{bmatrix}$

## Listeargumenter med ugyldige elementer

**SortA** og **SortD** flytter alle ugyldige elementer i det første argument til bunden.

$\{5,4,3,_,1\} \rightarrow list1$	$\{5,4,3,_,1\}$
$\{5,4,3,2,1\} \rightarrow list2$	$\{5,4,3,2,1\}$
SortA <i>list1,list2</i>	Done
<i>list1</i>	$\{1,3,4,5,_\}$
<i>list2</i>	$\{1,3,4,5,2\}$

$\{1,2,3,_,5\} \rightarrow list1$	$\{1,2,3,_,5\}$
$\{1,2,3,4,5\} \rightarrow list2$	$\{1,2,3,4,5\}$
SortD <i>list1,list2</i>	Done
<i>list1</i>	$\{5,3,2,1,_\}$
<i>list2</i>	$\{5,3,2,1,4\}$

I regressioner indfører en ugyldig værdi i en X- eller Y-liste en ugyldig værdi for det tilsvarende element i residualerne.

$l1:=\{1,2,3,4,5\}; l2:=\{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
LinRegMx <i>l1,l2</i>	Done
<i>stat.Resid</i>	$\{0.434286,_, -0.862857, -0.011429, 0.44\}$
<i>stat.XReg</i>	$\{1,_,3,4,5\}$
<i>stat.YReg</i>	$\{2,_,3,5,6,6\}$
<i>stat.FreqReg</i>	$\{1,_,1,1,1,1\}$

En udeladt kategori i regressioner indfører en ugyldig værdi for det tilsvarende element i residualerne.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
$cat:=\{"M", "M", "F", "F"\}; incl:=\{"F"\}$	$\{"F"\}$
LinRegMx <i>l1,l2,1,cat,incl</i>	Done
<i>stat.Resid</i>	$\{_,_,0,0,0\}$
<i>stat.XReg</i>	$\{_,_,4,5\}$
<i>stat.YReg</i>	$\{_,_,5,6,6\}$
<i>stat.FreqReg</i>	$\{_,_,1,1,1\}$

En hyppighed på 0 i regressioner indfører en ugyldig værdi i det tilsvarende element i residualerne.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
LinRegMx <i>l1,l2,\{1,0,1,1\}</i>	Done
<i>stat.Resid</i>	$\{0.069231,_, -0.276923, 0.207692\}$
<i>stat.XReg</i>	$\{1,_,4,5\}$
<i>stat.YReg</i>	$\{2,_,5,6,6\}$
<i>stat.FreqReg</i>	$\{1,_,1,1,1\}$

## Genveje til indtastning af matematiske udtryk

Med genveje kan du indtaste elementer i matematiskeudtryk ved at skrive i stedet for at anvende Katalog eller Symbolpaletten. Du kan for eksempel indtaste udtrykket  $\sqrt{6}$ , ved at skrive `sqrt(6)` i indtastningslinjen. Når du trykker på `Enter`, ændres udtrykket `sqrt(6)` til  $\sqrt{6}$ . Visse genveje er nyttige fra både den håndholdtes og computerens tastatur. Andre er hovedsageligt nyttige på computerens tastatur.

### Fra den håndholdtes eller computerens tastatur

Indsæt dette:	- ved at skrive denne genvej:
$\pi$	<code>pi</code>
$\theta$	<code>theta</code>
$\infty$	<code>uendeligt</code>
$\leq$	<code>&lt;=</code>
$\geq$	<code>&gt;=</code>
$\neq$	<code>/=</code>
$\Rightarrow$ (medfører)	<code>=&gt;</code>
$\Leftrightarrow$ (ensbetydende med)	<code>&lt;=&gt;</code>
$\rightarrow$ (gem operator)	<code>=:</code>
$   $ (absolut værdi)	<code>abs (...)</code>
$\sqrt{\quad}$	<code>sqrt (...)</code>
$\Sigma()$ (Sumskabelon)	<code>sumSeq (...)</code>
$\Pi()$ (Produktskabelon)	<code>prodSeq (...)</code>
$\sin^{-1}()$ , $\cos^{-1}()$ , ...	<code>arcsin (...)</code> , <code>arccos (...)</code> , ...
$\Delta\text{List}()$	<code>deltaList (...)</code>

### På computerens tastatur

Indsæt dette:	- ved at skrive denne genvej:
$i$ (imaginær konstant)	<code>@i</code>
$e$ (naturlig logbase $e$ )	<code>@e</code>
$E$ (videnskabelig notation)	<code>@E</code>
$T$ (transponere)	<code>@t</code>
$r$ (radianer)	<code>@r</code>
$^\circ$ (grader)	<code>@d</code>

Indsæt dette:	- ved at skrive denne genvej:
g (nygrader)	@g
∠ (vinkel)	@<
► (konvertering)	@>
►Decimal, ►approxFraction(), osv.	@>Decimal, @>approxFraction(), osv.

# Hierarkiet i EOS™ ligningsoperativsystemet (Equation Operating System)

I dette afsnit beskrives EOS™ (Equation Operating System) der anvendes af TI-Nspire™ læringsteknologien til matematik og naturfag. Tal, variable og funktioner indtastes i en nem og enkel rækkefølge. EOS™ softwaren beregner udtryk og ligninger ved hjælp af parenteser og efter de prioriteringer, der er beskrevet nedenfor.

## Operatorhierarkiet

Niveau	Operator
1	Parenteser ( ), kantparenteser [ ], krølp parenteser { }
2	Henvi sning (nr.)
3	Funktionskald
4	Postoperatorer: grader-minutter-sekunder ( <sup>°</sup> , ' , " ), fakultet (!), procentsats (%), radian ( <sup>∘</sup> ), lavtstillet ( [ ] ), skifte side for lighedstegnet ( <sup>⊤</sup> )
5	Eksponentiering, potensoperator ( <sup>^</sup> )
6	Negation ( - )
7	Strengsammenkædning &
8	Multiplikation ( • ), division ( / )
9	Addition ( + ), subtraktion ( - )
10	Lighedsrelationer: lig med ( = ), forskellig fra ( ≠ eller ≠ ), mindre end ( < ), mindre end eller lig med ( ≤ eller ≤ ), større end ( > ), større end eller lig med ( ≥ eller ≥ )
11	Logisk <b>not</b>
12	logisk <b>and</b>
13	Logisk <b>or</b>
14	<b>xor, nor, nand</b>
15	medfører, ⇒
16	ensbetydende med ( ⇔ )
17	Betingelses-operator ( "   " )
18	Gem ( → )

## Parenteser, kantparenteser og krølp parenteser

Alle beregninger inde i parenteser, kantede parenteser eller krøllede parenteser foretages først. F.eks. beregner EOS™ softwaren i udtrykket  $4(1+2)$  først den del af udtrykket, der er i parentes,  $1+2$ , og multiplicerer derefter resultatet, 3, med 4.



Antallet af start- og slutparenteser, -kantparenteser og -krølparenteser skal være det samme i et udtryk eller en ligning. Hvis ikke, vises en fejlmeddelelse, der viser det manglende element. For eksempel vil  $(1+2)/(3+4)$  udløse fejlmeddelelsen "Mangler."

**Bemærk:** Da du i TI-Nspire™ softwaren kan definere dine egne funktioner, tolkes et variabelnavn fulgt af et udtryk i parentes som et funktionskald i stedet for en underforstået multiplikation. For eksempel er  $a(b+c)$  funktionen  $a$  beregnet af  $b+c$ . For at gange udtrykket  $b+c$  med variabelen  $a$  anvendes explicit multiplikation:  $a*(b+c)$ .

## henvisning

Henvisningsoperatoren (#) konverterer en streng til en variabel eller et funktionsnavn. For eksempel opretter #("x"&"y"&"z") variabelnavnet xyz. Henvisning muliggør også oprettelse og ændring af variable inde i et program. Hvis for eksempel  $10 \rightarrow r$  og  $r \rightarrow s1$ , så  $\#s1=10$ .

## Post operatorer

Postoperatorer er operatorer, der følger direkte efter et argument som  $5!$ ,  $25\%$ , eller  $60^\circ 15' 45''$ . Argumenter efterfulgt af en postoperator, beregnes på fjerde prioritetsniveau. I udtrykket  $4^4 3!$  beregnes  $3!$  først. Resultatet,  $6$ , bliver derefter eksponenten af  $4$ , så det bliver  $4096$ .

## Eksponentiering

Eksponentiering (^) og elementvis eksponentiering (.^ ) beregnes fra højre mod venstre. Udtrykket  $2^4 3^2$  beregnes for eksempel på samme måde som  $2^4(3^2)$  og giver  $512$ . Dette er forskelligt fra  $(2^4 3)^2$ , der giver  $64$ .

## Negation

Negative tal indtastes ved at trykke på  $\boxed{-}$  efterfulgt af tallet. Postoperationer og eksponentiering udføres før negation. For eksempel er resultatet af  $-x^2$  et negativt tal, og  $-9^2 = -81$ . Med parenteser kvadreres et negativt tal som  $(-9)^2$ , så det giver  $81$ .

## Betingelse ("|")

Det argument, der følger efter ("|")-betingelses-operatoren, angiver et sæt betingelser, der skal tages hensyn til under beregningen af det argument, der kommer før operatoren.

# TI-Nspire CX II - TI-Grundlæggende programmeringsfunktioner

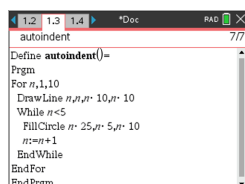
## Automatisk indrykning i programmeringseditoren

TI-Nspire™-programmeditor indrykker nu automatisk sætninger inde i en blokkommando.

Blokkommandoer er If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

Editoren forbereder automatisk mellemrum for programkommandoer inde i en blokkommando. Lukkekommandoen for blokken vil blive justeret med åbningskommandoen.

Eksemplet herunder viser automatisk indrykning i indlejrede blokkommandoer.



```
autoindent 7/7
Define autoindent()=
Prgm
For n,1,10
  DrawLine n,n,n-10,n-10
  While n<5
    FillCircle n-25,n-5,n-10
    n=n+1
  EndWhile
EndFor
EndPrgm
```

Kodefragmenter, der kopieres og indsættes, beholder den originale indrykning.

Åbning af et program, der er oprettet i en tidligere version af softwaren, bevarer den oprindelige indrykning.

---

## Forbedrede fejlmeddelelser til TI-Basic

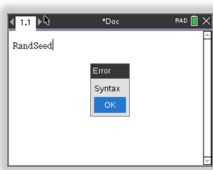
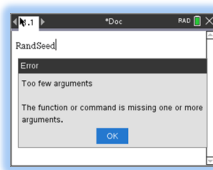
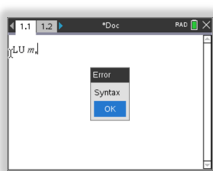
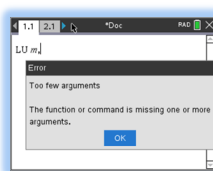
### Fejlmeddelelser

Fejltilstand	Ny meddelelse
Fejl i betingelseserklæring (If/While)	En betinget erklæring løstes ikke til <b>TRUE</b> eller <b>FALSE</b> <b>BEMÆRK:</b> Med ændringen for at placere markøren på linjen med fejlen, behøver vi ikke længere at angive, om fejlen er i en "If"-erklæring eller en "While"-erklæring.
Manglende EndIf	Forventede <b>EndIf</b> , men fandt en anden sluterklæring
Manglende EndFor	Forventede <b>EndFor</b> , men fandt en anden sluterklæring
Manglende EndWhile	Forventede <b>EndWhile</b> , men fandt en anden sluterklæring
Manglende EndLoop	Forventede <b>EndLoop</b> , men fandt en anden

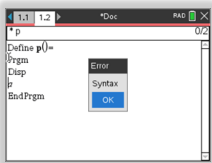
Fejltilstand	Ny meddelelse
	sluterklæring
Manglende <b>EndTry</b>	Forventede <b>EndTry</b> , men fandt en anden sluterklæring
“ <b>Then</b> ” udeladt efter <b>If</b> <condition>	Manglende <b>If..Then</b>
“ <b>Then</b> ” udeladt efter <b>Elseif</b> <condition>	<b>Then</b> mangler i blok: <b>Elseif</b> .
Når “ <b>Then</b> ”, “ <b>Else</b> ” og “ <b>Elseif</b> ” opstod uden for kontrolblokke	<b>Else</b> ugyldig uden for blokke: <b>If..Then..EndIf</b> eller <b>Try..EndTry</b>
“ <b>Elseif</b> ” vises uden for “ <b>If..Then..EndIf</b> ”-blokke	<b>Elseif</b> ugyldig uden for blok: <b>If..Then..EndIf</b>
“ <b>Then</b> ” vises uden for “ <b>If....EndIf</b> ”-blokke	<b>Then</b> ugyldig uden for blok: <b>If..EndIf</b>

## Syntaksfejl

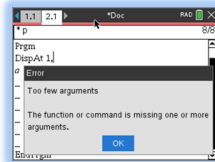
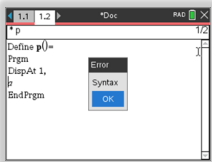
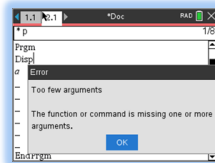
Hvis kommandoer, der forventer en eller flere argumenter, kaldes med en ufuldstændig liste over argumenter, vil en “**Too few argument-fejl**” blive udstedt i stedet for en “**syntax**”-fejl

Aktuel adfærd	Ny CX II-adfærd
 <p>A screenshot of a TI-84 Plus CE calculator interface. The command 'RandSeed' is entered. An error dialog box is displayed with the text 'Error Syntax' and an 'OK' button.</p>	 <p>A screenshot of a TI-84 Plus CE calculator interface. The command 'RandSeed' is entered. An error dialog box is displayed with the text 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.</p>
 <p>A screenshot of a TI-84 Plus CE calculator interface. The command 'L1D m]' is entered. An error dialog box is displayed with the text 'Error Syntax' and an 'OK' button.</p>	 <p>A screenshot of a TI-84 Plus CE calculator interface. The command 'L1D m]' is entered. An error dialog box is displayed with the text 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.</p>

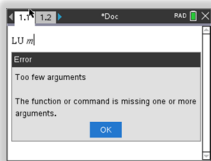
## Aktuel adfærd




## Ny CX II-adfærd



**Bemærk:** Når en ufuldstændig liste over argumenter ikke følges af et komma, er fejlmeddelelsen: "too few arguments". Dette er det samme som tidligere udgivelser.



## Konstanter og værdier

Det følgende skema oplister konstanter og deres værdier, som er tilgængelige, når der udføres enhedskonvertering. De kan indtastes manuelt eller vælges via **Konstant** listen i **Værktøjer > Enhedskonverteringer** (Håndholdt: Tryk  3).



Konstant	Navn	Værdi
_c	Lysets hastighed	299792458 _m/_s
_Cc	Coulombs konstant	8987551792.261 _m/_F
_Fc	Faradays konstant	96485.33212 _coul/_mol
_g	Tyngdeaccelerationen	9.80665 _m/_s <sup>2</sup>
_Gc	Tyngdekonstant	6.6743E-11 _m <sup>3</sup> /_kg/_s <sup>2</sup>
_h	Plancks konstant	6.62607015E-34 _J _s
_k	Boltzmanns konstant	1.380649E-23 _J/_°K
_μ0	Vakuumpermeabiliteten	1.25663706212E-6 _N/_A <sup>2</sup>
_μb	Bohr-magneton	9.274009994E-24 _J _m <sup>2</sup> /_Wb
_Me	Elektronens hvilemasse	9.1093837015E-31 _kg
_Mμ	Myonens masse	1.883531627E-28 _kg
_Mn	Neutronens hvilemasse	1.67492749804E-27 _kg
_Mp	Protonens hvilemasse	1.67262192369E-27 _kg
_Na	Avogadros konstant	6.02214076E23 /_mol
_q	Elementarladningen	1.602176634E-19 _coul
_Rb	Bohr-radius	5.29177210903E-11 _m
_Rc	Gaskonstanten	8.314462618 _J/_mol/_°K
_Rdb	Rydbergs konstant	10973731.568160/_m
_Re	Elektronradius	2.8179403262E-15 _m
_u	Atommasse	1.6605390666E-27 _kg
_Vm	Det molare volumen	2.241396954E-2 _m <sup>3</sup> /_mol
_ε0	Permittivitet for et vakuum (konstant)	8.8541878128E-12 _F/_m
_σ	Stefan-Boltzmanns konstant	5.670367E-8 _W/_m <sup>2</sup> /_°K <sup>4</sup>
_φ0	Magnetisk fluxkvantum	2.067833831E-15 _Wb

## Fejlkode og fejlmeddelelser

Hvis der opstår en fejl, tildeles dens kode variablen `errCode`. Brugerdefinerede programmer og funktioner kan undersøge `errCode` og bestemme årsagen til en fejl. Du finder et eksempel på anvendelse af `errCode` i Eksempel 2 under kommandoen **Try**, side 167.

**Bemærk:** Visse fejltilstande gælder kun for *TI-Nspire™ CAS-produkter*, og andre gælder kun for *TI-Nspire™-produkter*.

Fejlkode	Beskrivelse
10	En funktion returnerede ikke en værdi
20	En test gav ikke resultatet SAND eller FALSK.  Generelt kan udefinerede variable ikke sammenlignes. For eksempel forårsager testen <code>If a&lt;b</code> denne fejl, enten fordi <code>a</code> eller <code>b</code> er udefineret, når sætningen eksekveres.
30	Argumentet kan ikke være et navn på en mappe.
40	Argumentfejl
50	Argumentuoverensstemmelse  To eller flere argumenter skal være af samme type.
60	Argumentet skal være et boolsk udtryk eller et heltal
70	Argumentet skal være et decimaltal
90	Argumentet skal være en liste
100	Argumentet skal være en matrix
130	Argumentet skal være en streng
140	Argumentet skal være et variabelnavn.  Kontroller, at navnet: <ul style="list-style-type: none"><li>• ikke begynder med et tal</li><li>• ikke indeholder mellemrum eller specialtegn</li><li>• ikke anvender understregning eller punktum forkert</li><li>• ikke overskrider begrænsningerne for længden</li></ul> I afsnittet Regner i dokumentationen er der flere oplysninger.
160	Argumentet skal være et udtryk
165	Batterierne er for flade til at kunne sende/modtage  Isæt nye batterier, før du sender eller modtager
170	Grænse

Fejlkode	Beskrivelse
	Nedre grænse skal være mindre end øvre grænse for at definere søgeintervallet.
180	Afbrudt  Der er blevet trykket på  eller  -tasten under en længere beregning eller under en programeksekvierung.
190	Cirkulær definition  Denne meddelelse vises for at undgå at løbe tør for hukommelse under uendelig erstatning af variabelværdier under reduktion. For eksempel vil $a+1 \rightarrow a$ , hvor a er en undefineret variabel give denne fejl.
200	Ugyldigt begrænsningsudtryk  For eksempel vil solve( $3x^2-4=0,x$ )   $x < 0$ or $x > 5$ give denne fejlmeddelelse, fordi begrænsningen er adskilt med "or" i stedet for "and."
210	Ugyldig datatype  Et argument er af forkert datatype.
220	Afhængig grænse
230	Dimension  Et liste- eller et matrixindex er ikke gyldigt. Hvis listen {1,2,3,4} for eksempel gemmes i L1, vil L1[5] være en dimensionsfejl, fordi L1 kun indeholder fire elementer.
235	Dimensionsfejl. Der er ikke nok elementer i listerne.
240	Dimensionsuoverensstemmelse  To eller flere argumenter skal være af samme dimension. For eksempel er [1,2]+[1,2,3] en dimensionsuoverensstemmelse, fordi matrixerne indeholder et andet antal elementer.
250	Division med 0
260	Domænefejl  Et argument skal være i et angivet domæne. For eksempel er rand(0) ikke gyldig.
270	Variabelnavn findes allerede
280	Else og Elseif er ikke gyldige uden for blokken If..EndIf
290	EndTry mangler den tilsvarende Else-erklæring
295	For stor iteration
300	Forventede en liste med 2 eller 3 elementer eller en matrix

Fejlkode	Beskrivelse
310	Første argument af nSolve skal være en ligning i en enkelt variabel. Det kan ikke indeholde en variabel uden en værdi bortset fra den variabel, der skal undersøges.
320	Første argument for solve eller cSolve skal være en ligning eller en ulighed For eksempel er solve( $3x^4-4$ ,x) ugyldig, fordi det første argument ikke er en ligning.
345	Inkonsistente enheder
350	Indeks uden for området
360	Henvisningsstrengen er ikke et gyldigt variabelnavn
380	Udefineret Ans Enten gav den foregående beregning ikke en Ans, eller der blev ikke indtastet en tidligere beregning.
390	Ugyldig tildeling
400	Ugyldig tildelingsværdi
410	Ugyldig kommando
430	Ugyldigt for de aktuelle indstillinger
435	Ugyldigt gæt
440	Ugyldig implicit multiplikation for eksempel er $x(x+1)$ ugyldig, mens $x*(x+1)$ er den korrekte syntaks. Dette er for at undgå forveksling mellem implicit multiplikation og funktionskald.
450	Ugyldig i en funktion eller i det aktuelle udtryk Kun visse kommandoer er gyldige i en brugerdefineret funktion.
490	Ugyldig i Try..EndTryin Try..EndTry-blok
510	Ugyldig liste eller matrix
550	Ugyldig uden for en funktion eller et program Et antal kommandoer er ugyldige uden for en funktion eller et program. For eksempel kan Local ikke bruges, medmindre det er i en funktion eller program.
560	Ugyldig uden for blokkene Loop..EndLoop, For..EndFor eller While..EndWhile For eksempel er Exit-kommandoen kun gyldig inde i disse to løkkeblokke.
565	Ugyldig uden for et program
570	Ugyldigt stinavn For eksempel er \var ugyldigt.



Fejlkode	Beskrivelse
575	Ugyldigt pol-kompleks
580	Ugyldig programreference Programmer kan ikke kaldes i funktioner eller udtryk som $1+p(x)$ , hvor p er et program.
600	Ugyldig tabel
605	Ugyldig brug af enheder
610	Ugyldigt variabelnavn i en Local-sætning
620	Ugyldigt variabel- eller funktionsnavn
630	Ugyldig variabelhenvisning
640	Ugyldig vektorsyntaks
650	Link-overførsel En overførsel mellem to enheder blev ikke gennemført. Kontroller, at forbindelseskablet er godt tilsluttet i begge ender.
665	Matricen kan ikke diagonaliseres
670	Lav hukommelse 1. Slet nogle data i dette dokument 2. Gem og luk dette dokument Hvis 1 og 2 ikke lykkes, skal batterierne tages ud og sættes på plads igen
672	Resourcer opbrugt
673	Resourcer opbrugt
680	Mangler (
690	Mangler )
700	Mangler “
710	Mangler ]
720	Mangler }
730	Mangler starten eller slutningen af bloksyntaksen
740	Mangler Then i blokken If..EndIf
750	Navnet er ikke en funktion eller et program
765	Ingen funktion er valgt

Fejlkode	Beskrivelse
780	Ingen løsning blev fundet
800	Ikke-reelt resultat Hvis softwaren for eksempel er i indstillingen Reel, er $\sqrt{-1}$ ugyldig. For at kunne tillade komplekse resultater skal du ændre tilstanden Reel eller Kompleks til REKTANGULÆR eller POLÆR
830	Overløb
850	Programmet blev ikke fundet Et programkald inde i et andet program blev ikke fundet i den angivne sti under eksekvering.
855	Rand funktioner er ikke tilladt graftegning
860	For mange rekursioner
870	Reserveret navn eller systemvariabel
900	Argumentfejl Median-median-modellen kunne ikke anvendes på datasættet.
910	Syntaksfejl
920	Teksten blev ikke fundet
930	For få argumenter Funktionen eller kommandoen mangler et eller flere argumenter.
940	For mange argumenter Udtrykket eller ligningen indeholder for mange argumenter og kan ikke beregnes.
950	For mange lavtstillede tegn
955	For mange udefinerede variable
960	Variablen er ikke defineret Der er ikke tildelt en værdi til variablen. Anvend en af følgende kommandoer: <ul style="list-style-type: none"> <li>• <code>sto</code> →</li> <li>• <code>:=</code></li> <li>• <b>Define</b></li> </ul> for at tildele værdier til variable.
965	OS uden licens
970	Variabel i brug, så referencer eller ændringer er ikke tilladt

Fejlkode	Beskrivelse
980	Variablen er beskyttet
990	Ugyldigt variabelnavn Kontroller, at navnet ikke overskrider begrænsningerne for længden
1000	Talområde for Vindue variabel
1010	Zoom
1020	Intern fejl
1030	Overskridelse af beskyttet hukommelse
1040	Funktionen understøttes ikke. Denne funktion kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1045	Operatoren understøttes ikke. Denne operator kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1050	Funktionen understøttes ikke. Denne operator kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1060	Input-argumentet skal være et tal. Der er kun tilladt input med numeriske værdier.
1070	Trig-funktionsargumentet er for stort til en nøjagtig reduktion
1080	Ikke-understøttet anvendelse af Ans. Denne applikation understøtter ikke Ans.
1090	Funktionen er ikke defineret. Anvend en af følgende kommandoer: <ul style="list-style-type: none"> <li>• <b>Define</b></li> <li>• :=</li> <li>• sto →</li> </ul> til at definere en funktion.
1100	Ikke-reel beregning Hvis softwaren for eksempel er i indstillingen Reel, er $\sqrt{-1}$ ugyldig. For at kunne tillade komplekse resultater skal du ændre tilstanden Reel eller Komplex til REKTANGULÆR eller POLÆR
1110	Ugyldige grænser
1120	Intet fortegnsskift
1130	Argument kan ikke være en liste eller matrix
1140	Argumentfejl Det første argument skal være et polynomielt udtryk i det andet argument. Hvis det andet argument udelades, vil softwaren forsøge at vælge en standardværdi.
1150	Argumentfejl

Fejlkode	Beskrivelse
	De første to argumenter skal være polynomielle udtryk i det tredje argument. Hvis det tredje argument udelades, vil softwaren forsøge at vælge en standardværdi.
1160	Ugyldigt biblioteksstinavn  Et stinavn skal være på formen <code>xxx\yyy</code> , hvor: <ul style="list-style-type: none"> <li>• <code>xxx</code>-delen kan have 1 til 16 tegn.</li> <li>• <code>yyy</code>-delen kan have 1 til 15 tegn.</li> </ul> Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1170	Ugyldig brug af biblioteksstinavn <ul style="list-style-type: none"> <li>• En værdi kan ikke tildeles et stinavn med <b>Define</b>, <code>:=</code> eller <code>sto</code> →.</li> <li>• Et stinavn kan ikke erklæres som en lokal variabel eller anvendes som en parameter i en funktion eller en programdefinition.</li> </ul>
1180	Ugyldigt biblioteksvariabelnavn  Kontroller, at navnet: <ul style="list-style-type: none"> <li>• Ikke indeholder et punktum</li> <li>• Ikke begynder med en understregning</li> <li>• Ikke er længere end 15 tegn</li> </ul> Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1190	Biblioteksdokumentet blev ikke fundet. <ul style="list-style-type: none"> <li>• Kontroller, at biblioteket er i MyLib folder.</li> <li>• Opdater biblioteker.</li> </ul> Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1200	Biblioteksvariablen blev ikke fundet. <ul style="list-style-type: none"> <li>• Kontroller, at biblioteksvariablen findes i den første opgave i biblioteket.</li> <li>• Kontroller, at biblioteksvariablen er defineret som LibPub eller LibPriv.</li> <li>• Opdater biblioteker.</li> </ul> Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1210	Ugyldigt navn på biblioteksgenvej  Kontroller, at navnet: <ul style="list-style-type: none"> <li>• Ikke indeholder et punktum</li> <li>• Ikke begynder med en understregning</li> <li>• Ikke er længere end 16 tegn</li> <li>• Ikke er et reserveret navn</li> </ul> Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1220	Domænefejl:

Fejlkode	Beskrivelse
	Funktionerne tangentLine og normalLine understøtter kun reelle funktioner.
1230	Domænefejl. Trigonometriske konverteringsoperatorer understøttes ikke i vinkeltilstandene Grader eller Nygrader.
1250	Argumentfejl Brug et system af lineære ligninger . Eksempel på et system med to lineære ligninger med variablerne x og y: $3x+7y=5$ $2y-5x=-1$
1260	Argumentfejl: Det første argument til nfMin eller nfMax skal være et udtryk i en enkelt variabel. Det kan ikke indeholde en variabel uden en værdi bortset fra den variabel, der skal undersøges.
1270	Argumentfejl Differentialkvotientens orden skal være 1 eller 2.
1280	Argumentfejl Brug et polynomium i udvidet form i en variabel.
1290	Argumentfejl Brug et polynomium i en variabel.
1300	Argumentfejl Koefficienter i polynomiet skal beregnes til talværdier.
1310	Argumentfejl: En funktion kunne ikke beregnes i et eller flere af den argumenter.
1380	Domænefejl Indlejrede funktionskald er ikke tilladt i domain()-funktionen.

## Fejlkode- og meddelelser

Du kan bruge funktionen `warnCodes()` til at gemme fejlkoder genereret ved beregning af et udtryk. Denne tabel viser de numeriske fejlkoder og de tilhørende meddelelser. Se `warnCodes()` for at få et eksempel på lagring af fejlkoder, side 176.

Fejlkode	Meddelelse
10000	Operationen kan måske indføre falske løsninger. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10001	Differentiation af en ligning kan give en falsk ligning.
10002	Tvivlsom løsning Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10003	Tvivlsom præcision Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10004	Operationen kan miste løsninger. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10005	<code>cSolve</code> kan give flere nulpunkter.
10006	<code>Solve</code> kan give flere nulpunkter. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10007	Der kan være flere løsninger. Forsøg at specificere relevante nedre og øvre grænser og/eller et gæt. Eksempler med brug af <code>solve()</code> : <ul style="list-style-type: none"><li><code>solve(Ligning, Var=Gæt)   nedreGrænse&lt;Var&lt;øvreGrænse</code></li><li><code>solve(Ligning, Var)   nedreGrænse&lt;Var&lt;øvreGrænse</code></li><li><code>solve(Ligning,Var=Gæt)</code></li></ul> Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10008	Domænet af resultatet kan være mindre end domænet af inputtet.
10009	Domænet af resultatet kan være større end domænet af inputtet.
10012	Ikke-reel beregning
10013	$\infty^0$ eller $\text{undef}^0$ erstattet med 1
10014	$\text{undef}^0$ erstattes med 1
10015	$1^\infty$ eller $1^\text{undef}$ erstattet med 1
10016	$1^\text{undef}$ erstattes med 1

Fejlkode	Meddelelse
10017	Overløb erstattes med $\infty$ eller $-\infty$
10018	Operationen kræver og returnerer en 64 bit-værdi.
10019	Resursen er udtømt, reduktionen er måske ufuldstændig.
10020	Trig-funktionsargumentet er for stort til en nøjagtig reduktion.
10021	Inputtet indeholder en udefineret parameter. Resultatet er muligvis ikke gyldigt for alle mulige parameterværdier.
10022	Angivelse af korrekte nedre og øvre grænser kan give en løsning.
10023	Skalar er blevet multipliceret med identitetsmatrixen.
10024	Resultatet er opnået ved hjælp af approksimativ aritmetik.
10025	Ækvivalensen kan ikke verificeres i tilstanden EXACT.
10026	Begrænsning ignoreres muligvis. Specificer begrænsning på formen "\" Variable MathTestSymbol Constant" eller en konjunktion af disse formler eksempelvis "x<3 and x>-12"

## Generelle oplysninger

### **Online hjælp**

[education.ti.com/eguide](http://education.ti.com/eguide)

Vælg dit land for at få flere produktoplysninger.

### **Kontakt TI-Support**

[education.ti.com/ti-cares](http://education.ti.com/ti-cares)

Vælg dit land for at finde ressourcer for teknisk support og andre supportressourcer.

### **Service og garanti**

[education.ti.com/warranty](http://education.ti.com/warranty)

Vælg dit land for at få oplysninger om varigheden og betingelserne for garantien, eller om produktservice.

Begrænset reklamationsret. Denne garanti påvirker ikke dine lovbestemte rettigheder.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243



# Indeks

-		-	
-, subtrahere .....	185	_, enhedsbenævnelse .....	204
!			
!, faktulet .....	195	, betingelses-operator .....	205
"		,	
", sekundnotation .....	203	', minutnotation .....	203
#		+	
#, henvisning .....	201	+, adder .....	185
#, henvisningsoperator .....	229	=	
%		≠, forskellig fra .....	192
%, procent .....	191	=, lig med .....	191
&		>	
&, tilføj .....	195	>, større end .....	193
*		∏	
*, multiplicere .....	186	∏, produkt .....	198
.		∑	
., punktum subtraktion .....	189	∑( ), sum .....	198
., punktum multiplikation .....	190	∑Int( ) .....	199
./, punkt division .....	190	∑Prn( ) .....	200
., punktum potens .....	190	√	
., punktum addition .....	189	√, kvadratrodsymbol .....	197
/		∫	
/, dividere .....	187	∫, integral .....	197
:		≤	
:=, tildel .....	207	≤, mindre end eller lig med .....	193
^		≥	
$\wedge^{-1}$ , reciprok .....	205	≥, større end eller lig med .....	194
$\wedge$ , potens .....	188		

	►	Absolut værdi skabelon til .....	3-4
►approxFraction( ) .....	13	adder, + .....	185
►Base10, vise som decimalt heltal ..	18		
►Base16, vise som hexadecimalt ..	19	<b>Æ</b>	
►Base2, vise som binær .....	17	ægte brøk, propFrac .....	121
►Cylind, vise som cylindrisk vektor ..	36		
►DD, vise som decimal vinkel .....	37	<b>A</b>	
►Decimal, vise resultat som decimal	37		
►DMS, vise som grader/minutter/sekunder	44	afrund, round( ) .....	138
►Grad, konvertere til vinkelmål i nygrader .....	71	<b>A</b>	
►Polar, vise som polær vektor .....	118	afslut løkke, EndLoop .....	93
►Rad, konverter til vinkelmål i radianer .....	126	<b>A</b>	
►Rect, vis som rektangulær vektor ..	129	afslut, Exit .....	51
►Sphere, vise som en kuglevektor ..	154		
	→	<b>A</b>	
→, lagring .....	207	afslutte funktion, EndFunc .....	60
	⇒	<b>A</b>	
⇒, medfører[*] .....	194, 226	amortiseringstabel, amortTbl( ) .....	7, 16
	↔	<b>A</b>	
↔, ensbetydende med[*] .....	195	amortTbl( ), amortiseringstabel .....	7, 16
	©	<b>A</b>	
©, kommentar .....	208	and, Boolsk operator .....	8
	°	<b>A</b>	
°, grader/minutter/sekunder .....	203	anden afledede skabelon til .....	6
°, gradnotation .....	202		
	0	<b>A</b>	
0b, binær indikator .....	208	andengradsregression, QuadReg ...	123
0h, hexadecimal indikator .....	208		
	1	<b>A</b>	
10^( ), tiende potens .....	205	angle( ), vinkel .....	9
	<b>A</b>	<b>A</b>	
abs( ), absolut værdi .....	7	Anmodning .....	132

<b>A</b>		<b>A</b>	
ANOVA, envejs variansanalyse .....	9	arcsin() .....	14
<b>A</b>		<b>A</b>	
ANOVA2way, to-vejs variansanalyse	10	arcsinh() .....	14
<b>A</b>		<b>A</b>	
ans, sidste resultat .....	12	arctan() .....	15
<b>A</b>		<b>A</b>	
approx( ), tilnærmet .....	13	arctanh() .....	15
<b>A</b>		<b>A</b>	
approxRational( ) .....	13	arcuscosinus, $\cos^{-1}()$ .....	27
<b>A</b>		<b>A</b>	
arccos() .....	14	arcussinus, $\sin^{-1}()$ .....	150
<b>A</b>		<b>A</b>	
arccosh() .....	14	arcustangens, $\tan^{-1}()$ .....	162
<b>A</b>		<b>A</b>	
arccot() .....	14	argumenter i TVM-funktioner .....	171
<b>A</b>		<b>A</b>	
arccoth() .....	14	augment( ), udvid/sammenkæd ....	15
<b>A</b>		<b>A</b>	
arccsc() .....	14	avgRC( ), gennemsnitlig ændringshastighed .....	15
<b>A</b>		<b>B</b>	
arccsch() .....	14	beregne polynomium, polyEval( ) ...	118
<b>A</b>		beregning, rækkefølge af .....	228
arcsec() .....	14	Bestemt integral	
<b>A</b>		skabelon til .....	6
arcsech() .....	14	betingelses-operator " " .....	205
		betingelses-operator, rækkefølge for en beregning .....	228
		Bibliotek (Library)	
		danner genveje til objekter .....	80

binær			
indikator, 0b	208		
vise, ►Base2	17		
binomCdf()	19, 76		
binomPdf()	20		
Boolske operatører			
⇒	194, 226		
⇔	195		
and	8		
nand	103		
nor	108		
not	110		
or	114		
xor	177		
brøker			
ægteBrøk	121		
skabelon til	1		
brugerdefinerede funktioner	38		
brugerdefinerede funktioner og programmer	39		
<b>C</b>			
Cdf()	54		
ceiling(), rund op	20		
centralDiff()	20		
char(), tegnstring	21		
ClearAZ	23		
ClrErr, slet fejl	23		
colAugment	24		
colDim(), matrix kolonnedimension	24		
colNorm(), matrix kolonnenorm	24		
conj(), komplekst konjugat	25		
constructMat(), konstruer matrix	25		
corrMat(), korrelationsmatrix	26		
cos <sup>-1</sup> , arcuscosinus	27		
cos(), cosinus	26		
cosh <sup>-1</sup> (), hyperbolsk arcuscosinus	29		
cosh(), hyperbolsk cosinus	28		
cosinus, cos()	26		
cot <sup>-1</sup> (), arcuscotangens	30		
cot(), cotangens	29		
cotangens, cot()	29		
coth <sup>-1</sup> (), hyperbolsk arcuscotangens	30		
coth(), hyperbolsk cotangens	30		
count(), tælle elementer i en liste	31		
countif(), tælle elementer i en liste betinget	31		
cPolyRoots()	32		
crossP(), vektorprodukt	32		
csc <sup>-1</sup> (), invers cosecans	33		
csc(), cosecans	33		
csch <sup>-1</sup> (), invers hyperbolsk cosecans	34		
csch(), hyperbolsk cosecans	34		
CubicReg, tredjegradsregression	34		
cumulative sum, cumulativeSum()	35		
cumulativeSum(), cumulative sum	35		
Cycle, cyklus	36		
cyklus, Cycle	36		
cylindrisk vektorvisning, ►Cylind	36		
<b>D</b>			
d(), første differentialkvotient	196		
dage mellem datoer, dbd()	36		
dbd(), dage mellem datoer	36		
decimal			
heltal vise, ►Base10	18		
vinkelvisning, ►DD	37		
Define	38		
Define LibPriv	39		
Define LibPub	39		
Define, definere	38		
definere			
offentlig funktion eller program	39		
privat funktion eller program	39		
definere, Define	38		
delmatrix, subMat()	159-160		
deltaList()	40		
DelVar, slette variabel	40		
delVoid(), fjerner ugyldige elementer	41		
det(), matrixdeterminant	41		
diag(), matrixdiagonal	41		
differentialkvotienter			
første differentialkvotient, d()	196		
numerisk differentialkvotient, nDeriv()	106-107		
numerisk differentialkvotient, nDerivative()	105		
dim(), dimension	42		
dimension, dim()	42		
Disp, vis data	42, 142		
DispAt	43		
distributionsfunktioner			
binomCdf()	19		
binomPdf()	20		
normCdf()	109		

normPdf( )	110	Exit, afslut	51
poissCdf( )	117	exp( ), e til en potens	51
poissPdf( )	117	expr( ), streng til udtryk	52
tCdf( )	164	ExpReg, eksponentiel regression	52
tPdf( )	166		
$\chi^2$ 2way( )	21	<b>F</b>	
$\chi^2$ Cdf( )	22	F-test- med 2 målinger	59
$\chi^2$ GOF( )	22	factor( ), opløse i faktorer	53
$\chi^2$ Pdf( )	23	fakultet, !	195
dividere, P	187	fejlkoder og -meddelelser	242
division af heltal, intDiv( )	74	fejlmdelelser og fejlfinding	
dot		slette fejl, ClrErr	23
produkt, dotP( )	45	videresende fejl, PassErr	116
dotP( ), prikprodukt	45	Fill, matrixudfyldning	54
<b>E</b>		finansfunktioner, tvmFV( )	169
e eksponent		finansfunktioner, tvml( )	170
skabelon til	2	finansfunktioner, tvmN( )	170
e til en potens, e^( )	45, 51	finansfunktioner, tvmPmt( )	170
E, eksponent	201	finansfunktioner, tvmPV( )	171
e^( ), e til en potens	45	FiveNumSammendrag	55
eff ), omregn nominal til effektiv		fjerdegradsregression, QuartReg	124
rente	46	floor( ), runde ned	56
effektiv rente, eff( )	46	flyt, shift( )	146
egenværdi, eigVl( )	47	For	56
egenvektor, eigVc( )	46	for, For	56
eigVc( ), egenvektor	46	For, for	56
eigVl( ), egenværdi	47	fordelingsfunktioner	
eksponent, E	201	binomCdf( )	76
eksponenter		invNorm( )	76
skabelon til	1	invt( )	77
eksponentiel regression, ExpReg	52	Invx <sup>2</sup> ( )	75
ellers hvis, Elseif	47	format( ), formater streng	57
ellers, Else	71	formater streng, format( )	57
Elseif, ellers hvis	47	forskellig fra, ≠	192
end		første orden	
for, EndFor	56	skabelon til	5
løkke...EndLoop	93	fortegn, sign( )	148
EndWhile, slut while-kompleks	177	fpart( ), funktionsdel	57
enhedsvektor, unitV( )	173	freqTable( )	58
ensbetydende med, ⇔	195	frequency( )	58
EOS ligningsoperativsystem		Frobenius-norm, norm( )	109
(Equation Operating System)	228	Func, funktion	60
Equation Operating System (EOS)		Func, programfunktion	60
ligningsoperativsystem	228	funktioner	
etiket, Lbl	78	brugerdefinerede	38
euler( ), Euler function	48	del, fpart( )	57
		programfunktion, Func	60
		funktioner og variable	
		kopiere	25

fyld ..... 216-217

## G

g, nygrader ..... 201  
gå til, Goto ..... 70  
gcd(), største fælles divisor ..... 61  
gennemsnitlig ændringshastighed,  
avgRC() ..... 15  
genveje, tastatur ..... 226  
geomCdf() ..... 61  
geomPdf() ..... 61  
Get ..... 62, 218  
getDenom(), hente/returnere  
nævner ..... 63  
getKey() ..... 63  
getLangInfo(), hent/returner  
sproginformation ..... 66  
getLockInfo(), tester låsestatus af en  
variabel eller  
variabelgruppe ..... 67  
getMode(), hente  
tilstandsindstillinger ..... 67  
getNum(), hente/returnere tal ..... 68  
GetStr ..... 68  
getType(), get type of variable ..... 69  
getVarInfo(), hent/returner  
variabeloplysninger ..... 69  
Goto, gå til ..... 70  
grader/minutter/sekunder-notation  
gradnotation, g ..... 203  
grupper, låse og låse op ..... 89, 174  
grupper, teste låsestatus ..... 67

## H

heltal, int() ..... 74  
heltalsdel, iPart() ..... 77  
hente/returnere  
nævner, getDenom() ..... 63  
tal, getNum() ..... 68  
variabeloplysninger, getVarInfo()  
) ..... 66, 69  
henvi sning, # ..... 201  
henvi sningsoperator (#) ..... 229  
hexadecimal  
indikator, Oh ..... 208  
vise, ►Base16 ..... 19  
højre, right() ..... 75, 135

hvis, If ..... 71  
hvisFn() ..... 72  
hyperbolsk  
arctangens, tanh<sup>-1</sup>() ..... 163  
arcuscosinus, cosh<sup>-1</sup>() ..... 29  
arcussinus, sinh<sup>-1</sup>() ..... 151  
cosinus, cosh() ..... 28  
sinus, sinh() ..... 151  
tangens, tanh() ..... 162

## I

i streng, inString() ..... 74  
identitetsmatrix, identity() ..... 71  
identity(), identitetsmatrix ..... 71  
If, hvis ..... 71  
imag(), imaginær del ..... 73  
imaginær del, imag() ..... 73  
indstille  
tilstand, setMode() ..... 145  
indstillinger, hente aktuelle ..... 67  
inString(), i streng ..... 74  
int(), heltal ..... 74  
intDiv(), division af heltal ..... 74  
integral, f ..... 197  
interpolate(), interpolation ..... 75  
invers kumuleret normalfordeling  
(invNorm()) ..... 76  
invers, <sup>-1</sup> ..... 205  
invF() ..... 75  
invNorm(), invers kumuleret  
normalfordeling) ..... 76  
invf() ..... 77  
Invχ<sup>2</sup>() ..... 75  
iPart(), heltalsdel ..... 77  
irr(), intern rente  
intern rente, irr() ..... 77  
isPrime(), primtalstest ..... 78  
isVoid(), test for ugyldighed ..... 78

## K

kombinationer, nCr() ..... 104  
kommentar, © ..... 208  
komplekst  
konjugat, conj() ..... 25  
Konstruer matrix, constructMat() ..... 25  
konverter  
►Rad ..... 126



rækker, mRowAdd( ) ..		mindste fælles multiplum, lcm .....	79
ny, newMat( ) .....	106	minimum, min( ) .....	98
opløsning i øvre-nedre, LU .....	94	minutnotation, .....	203
produkt, product( ) .....	121	mirr( ), modificeret intern	
punktum addition, .+ .....	189	forrentning .....	99
punktum division, ./ .....	190	mod( ), modulo .....	100
punktum multiplikation, .* .....	190	modificeret intern forrentning, mirr(	
punktum potens, .^ .....	190	) .....	99
punktum subtraktion, .- .....	189	modulo, mod( ) .....	100
QR opløse i faktorer, QR .....	122	mRow( ), matrixrækkeoperation .....	100
rækkeoperation, mRow( ) .....	100	mRowAdd( ), multiplikation og	
reduceret række-echelonform,		addition af matrixrækker ..	100
rref( ) .....	140	Multiple lineær regression ttest .....	102
summation, sum( ) .....	159-160	multiplacere, * .....	186
transponere, T .....	161	MultReg .....	100
udfylde, Fill .....	54	MultRegIntervals( ) .....	101
udvid/sammenkæd, augment( )	15	MultRegTests( ) .....	102
vilkårlige, randMat( ) .....	127		
matrix (1 × 2)		<b>N</b>	
skabelon til .....	4	nand, Boolsk operator .....	103
matrix (2 × 1)		når, when( ) .....	176
skabelon til .....	4	naturlig logaritme, ln( ) .....	86
matrix (2 × 2)		nCr( ), kombinationer .....	104
skabelon til .....	4	nDerivative( ), numerisk	
matrix (m × n)		differentialkvotient .....	105
skabelon til .....	4	negation, indtaste negative tal .....	229
matrix til liste, matHist( ) .....	94	nettonutidsværdi, npv( ) .....	111
matrixer		newList( ), ny liste .....	106
række-echelonform, ref( ) .....	130	newMat( ), ny matrix .....	106
rækkeaddition, rowAdd( ) .....	139	nfMax( ), numerisk	
række dimension, rowDim( ) .....	139	funktionsmaksimum .....	106
rækkenorm, rowNorm( ) .....	139	nfMin( ), numerisk	
rækkeskift, rowSwap( ) .....	140	funktionsminimum .....	107
max( ), maksimum .....	95	nInt( ), numerisk integral .....	107
mean( ), middel .....	95	nom ), omregn effektiv til nominel	
med,   .....	205	rente .....	108
medfører, ⇒ .....	194, 226	nominel rente, nom( ) .....	108
median( ), median .....	96	nor, Boolsk operator .....	108
median, median( ) .....	96	norm( ), Frobenius-norm .....	109
medium-medium linjeregression,		normalsandsynlighedsfordeling,	
MedMed .....	97	normCdf( ) .....	109
MedMed, medium-medium		normCdf( ) .....	109
linjeregression .....	97	normPdf( ) .....	110
mens, While .....	177	not, Boolsk operator .....	110
mid( ), midt-streng .....	98	nPr( ), permutationer .....	110
middel, mean( ) .....	95	npv( ), nettonutidsværdi .....	111
midt-streng, mid( ) .....	98	nSolve( ), numerisk løsning .....	112
min( ), minimum .....	98		
mindre end eller lig med, { .....	193		





reduceret række-echelonform, rref( ) .....	140	<b>S</b>	
ref( ), række-echelonform .....	130	sammenhørende ligninger, simult( ) .....	148
RefreshProbeVars .....	131	sandsynlighedsdensitivitet, normPdf( ) .....	110
regressioner		sec <sup>-1</sup> ( ), invers sekans .....	141
andengrads, QuadReg .....	123	sec( ), sekans .....	140
eksponentiel, ExpReg .....	52	sech <sup>-1</sup> ( ), invers hyperbolsk sekans ..	141
fjerdegrads, QuartReg .....	124	sech( ), hyperbolsk sekans .....	141
lineær regression, LinRegAx .....	81	sekundnotation, " .....	203
lineær regression, LinRegBx .....	80	sekvens, seq( ) .....	143
logistiske, Logistik .....	91	seq( ), sekvens .....	143
potensregression, PowerReg ...	119, 132, 134, 164	seqGen( ) .....	143
sinus, SinReg .....	152	seqn( ) .....	144
tredjegrads, CubicReg, CubicReg	34	sequence, seq( ) .....	143-144
Regressioner		setMode( ), indstil tilstand .....	145
lineær regression, LinRegBx .....	82	shift( ), flyt .....	146
logaritmisk, LnReg .....	87	sign( ), fortegn .....	148
Logistisk .....	90	simult( ), sammenhørende ligninger	148
medium-medium linje, MedMed	97	sin <sup>-1</sup> ( ), arcsin .....	150
MultReg .....	100	sin( ), sinus .....	149
rektangulær vektorvisning, ►Rect ...	129	sinh <sup>-1</sup> ( ), hyperbolsk arcsin .....	151
rektangulært x-koordinat, ►►Rx( ) ..	115	sinh( ), hyperbolsk sinus .....	151
rektangulært y-koordinat, ►►Ry( ) ..	116	SinReg, sinusregression .....	152
remain( ), rest .....	132	sinus, sin( ) .....	149
RequestStr .....	134	sinusregression, SinReg .....	152
rest, remain( ) .....	132	skabeloner	
resultat (sidste), ans .....	12	Absolut værdi .....	3-4
resultater, statistiske .....	155	anden afledede .....	6
resultatværdier, statistik .....	156	bestemt integral .....	6
Return, returner .....	135	brøk .....	1
returner, Return .....	135	e eksponent .....	2
right, right( ) .....	48, 176	eksponent .....	1
rk23( ), Runge Kutta-funktion .....	135	første orden .....	5
rotate( ), roter .....	137	kvadratrod .....	1
roter, rotate( ) .....	137	ligningssystem (2-ligninger) .....	3
round( ), afrund .....	138	ligningssystem (N-ligning) .....	3
rowAdd( ), addition af matrixrækker	139	Log .....	2
rowDim( ), matrixrækkedimension ..	139	matrix (1 × 2) .....	4
rowNorm( ), matrixrækkenorm .....	139	matrix (2 × 1) .....	4
rowSwap( ), matrixrækkeskift .....	140	matrix (2 × 2) .....	4
rref( ), reduceret række-echelonform .....	140	matrix (m × n) .....	4
rund op, ceiling( ) .....	20	nte rod .....	1
runde ned, floor( ) .....	56	produkt Π( ) .....	5
runde op, ceiling( ) .....	20, 32	stykkevis-skabelon (N stykker) ..	3
Ryd .....	212	stykkevis funktion (2 stykker) ..	2
		sum Σ( ) .....	5
		slet	
		fejl, ClrErr .....	23

slætte		
ugyldige elementer fra liste	41	
variabel, DelVar	40	
slut		
funktion, EndFunc	60	
hvis, EndIf	71	
mens, EndWhile	177	
slut på hvis, EndIf	71	
slut while-kompleks, EndWhile	177	
som af rentebetalinger	199	
SortA, sorter stigende	153	
SortD, sortere faldende	154	
sortere		
faldende, SortD	154	
stigende, SortA	153	
sprog		
hent sproginformation	66	
sqrt(), kvadratrod	155	
standardafvigelse, stdDev()	157, 174	
stat.results	155	
stat.values	156	
statistik		
faktuel, !	195	
kombinationer, nCr()	104	
median, median()	96	
middelværdi, mean()	95	
permutationer, nPr()	110	
standardafvigelse, stdDev()	157, 174	
statistik med en variabel,		
OneVar	113	
to-variabel-resultater, TwoVar	171	
variens, variance()	174	
vilkårlig norm, randNorm()	127	
vilkårligt tal-seed, RandSeed	128	
statistik med en variabel, OneVar	113	
stdDevPop(), population		
standardafvigelse	157	
stdDevSamp(), standardafvigelse for måling	157	
Stop-kommando	158	
større end eller lig med,	194	
større end, >	193	
største fælles divisor, gcd()	61	
streng		
dimension, dim()	42	
længde	42	
strenge		
anvende til at skabe variabelnavne	229	
flyt, shift()	146	
formaterer	57	
formaterer, format()	57	
henvisning, #	201	
højre, right()	75, 135	
i, inString	74	
midt-streng, mid()	98	
roter, rotate()	137	
streng til udtryk, expr()	52	
tegnkode, ord()	115	
tegnstreng, char()	21	
tilføj, &	195	
udtryk til streng, string()	158	
venstre, left()	79	
string(), udtryk til streng	158	
strings		
right, right()	48, 176	
student-t sandsynlighedsfordeling, tCdf()	164	
student-t sandsynlighedstæthed, tPdf()	166	
stykkevis-skabelon (N stykker)		
skabelon til	3	
stykkevis funktion (2 stykker)		
skabelon til	2	
subMat(), delmatrix	159-160	
substitution med " " -operator	205	
subtrahere, -	185	
sum $\sum()$		
skabelon til	5	
sum af hovedstolsbetalinger	200	
sum(), summation	159	
sum, S()	198	
sumif()	160	
summation, sum()	159	
sumSeq()	160	
<b>T</b>		
T (transponere)	161	
t test, tTest	168	
tælle dage mellem datoer, dbd()	36	
tælle elementer i en liste betinget, countif()	31	
tælle elementer i en liste, count()	31	
tan <sup>-1</sup> (), arcustangens	162	
tan(), tangens	161	
tangens, tan()	161	
tanh <sup>-1</sup> (), hyperbolsk arcustangens	163	

tanh( ), hyperbolsk tangens .....	162		
tastaturgenveje .....	226		
tCdf(), student t sandsynlighedsfordeling .....	164		
tegn .....	213-215		
numerisk kode, ord( ) .....	115		
streng, char( ) .....	21		
tegnstreng, char( ) .....	21		
test for ugyldighed, isVoid( ) .....	78		
Test_2S, F test med 2 målinger .....	59		
Text-kommando .....	164		
tidsdiskonterede pengeværdier, aktuel værdi .....	171		
tidsdiskonterede pengeværdier, antal betalinger .....	170		
tidsdiskonterede pengeværdier, betalingsbeløb .....	170		
tidsdiskonterede pengeværdier, Fremtidig værdi .....	169		
tidsdiskonterede pengeværdier, Rente .....	170		
tiende potens, 10^( ) .....	205		
tilføj, & .....	195		
tilnærmet, approx( ) .....	13		
tilstande indstille, setMode( ) .....	145		
tilstandsindstillinger, getMode( ) .....	67		
tInterval, t konfidensinterval .....	165		
tInterval_2Samp, tomålingers t konfidensinterval .....	165		
to-variabel-resultater, TwoVar .....	171		
tomme (ugyldige) elementer .....	224		
tPdf(), student sandsynlighedstæthed .....	166		
transponere, T .....	161		
tredjegradsregression, CubicReg .....	34		
Try, fejlhåndteringskommando .....	167		
tTest, t test .....	168		
tTest_2Samp, to-målingers t test .....	168		
TVM-argumenter .....	171		
tvmFV( ) .....	169		
tvml( ) .....	170		
tvmN( ) .....	170		
tvmPmt( ) .....	170		
tvmPV( ) .....	171		
TwoVar, to-variabel-resultater .....	171		
		<b>U</b>	
		udelukkelse med " " -operator .....	205
		udtryk streng til udtryk, expr( ) .....	52
		udvid/sammenkæd, augment( ) .....	15
		ugyldige elementer .....	224
		ugyldige elementer, fjerne .....	41
		ugyldighed, test for .....	78
		understregning, _ .....	204
		unitV( ), enhedsvektor .....	173
		unlock, oplåse variabel eller variabel gruppe .....	174
		<b>V</b>	
		variabel oprette navn fra en tegnstreng .....	229
		slette alle enkeltbogstaver .....	23
		variable lokal, Local .....	89
		slette, DelVar .....	40
		variable og funktioner kopiere .....	25
		variable, låse og låse op .....	67, 89, 174
		varians, variance( ) .....	174
		varPop( ) .....	174
		varSamp( ), målingsvarians .....	174
		vektorer cylindrisk vektorvisning, ►Cylind .....	36
		enhed, unitV( ) .....	173
		prikprodukt, dotP( ) .....	45
		vektorprodukt, crossP( ) .....	32
		vektorprodukt, crossP( ) .....	32
		venstre, left( ) .....	79
		Vent på kommando .....	175
		videresende fejl, PassErr .....	116
		vilkårlig matrix, randMat( ) .....	127
		vilkårlig norm, randNorm( ) .....	127
		vilkårlig stikprøve .....	128
		vilkårligt tal-seed, RandSeed .....	128
		vinkel, angle( ) .....	9
		vis data, Disp .....	42, 142
		vis som rektangulær vektor, ►Rect .....	129
		vis som binær, ►Base2 .....	17
		cylindrisk vektor, ►Cylind .....	36

decimal vinkel, ►DD .....	37
decimalt heltal, ►Base10 .....	18
grader/minutter/sekunder, ►DMS .....	44
hexadecimal, ►Base16 .....	19
kuglevektor, ►Sphere .....	154
polær vektor, ►Polar .....	118
Viser grader/minutter/sekunder, ►DMS .....	44

## W

warnCodes( ), Warning codes .....	176
when( ), når .....	176
While, mens .....	177

## X

$x^2$ , kvadrat .....	189
XNOR .....	195
xor, Boolsk exclusive or .....	177

## Z

zInterval, z konfidensinterval .....	178
zInterval_1Prop, en-proportionalt z konfidensinterval .....	179
zInterval_2Prop, to-proportionalt z konfidensinterval .....	179
zInterval_2Samp, z konfidensinterval med to-målinger .....	180
zTest .....	181
zTest_1Prop, enproportional z test .	182
zTest_2Prop, toproportional z test .	182
zTest_2Samp, to-prøvers z test .....	183

## Δ

Δlist( ), listedifferens .....	86
--------------------------------	----

## χ

$\chi^2$ 2way .....	21
$\chi^2$ Cdf( ) .....	22
$\chi^2$ GOF .....	22
$\chi^2$ Pdf( ) .....	23