



TI-Nspire™ CX CAS

Opslagsvejledning

Få mere at vide om TI Technology via online-hjælpen på education.ti.com/eguide.

Vigtige oplysninger

Medmindre andet udtrykkeligt angives i den Licens, der følger med et program, stiller Texas Instruments ingen garantier, hverken udtrykkeligt eller underforstået, herunder men ikke begrænset til underforståede garantier om salgbarhed og egnethed til et bestemt formål, for programmer eller skriftligt materiale, og Texas Instruments stiller udelukkende sådant materiale til rådighed, som det foreligger. Texas Instruments kan under ingen omstændigheder holdes ansvarlig for særlige, indirekte, hændelige eller følgeskader i forbindelse med eller som følge af køb eller brug af dette materiale, og hele Texas Instruments' erstatningsansvar kan, uanset søgsmålets art, ikke overstige det beløb, der fremgår af programlicensen. Derudover kan Texas Instruments ikke holdes ansvarlig for nogen form for krav som følge af en anden parts brug af dette materiale.

© 2006 - 2019 Texas Instruments Incorporated

Indholdsfortegnelse

Udtryksskabeloner	1
Alfabetisk oversigt	8
A	8
B	17
C	21
D	46
E	59
F	69
G	79
I	90
L	98
M	114
N	124
O	131
P	134
Q	143
R	146
S	161
T	186
U	202
V	203
W	204
X	206
Z	207
Symboler	216
Tomme (ugyldige) elementer	243
Genveje til indtastning af matematiske udtryk	245
Hierarkiet i EOS™ ligningsoperativsystemet (Equation Operating System)	247
Konstanter og værdier	249
Fejlkoder og fejlmeddelelser	250
Fejlkoder og -meddelelser	258
Generelle oplysninger	260
Online hjælp	260

Kontakt TI-Support	260
Service og garanti	260
Indeks	261

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 218.

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 219.

Eksempel:

$$2^3 = 8$$

Kvadratrodsskabelon

ctrl **x²** -taster



Bemærk: Se også $\sqrt{}$ (kvadratrod), side 229.

Eksempel:

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

Nte rod-skabelon

ctrl **^** -taster



Bemærk: Se også **root()**, side 157.

Eksempel:

Nte rod-skabelon

ctrl **^** -taster

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

$$\sqrt[3]{\{8, 27, b\}} \quad \left\{ 2, 3, b^{\frac{1}{3}} \right\}$$

e eksponentskabelon

e^x -taster

e

Den naturlige eksponentialfunktion e opløftet til en potens

Bemærk: Se også $e^{\wedge}()$, side 59.

Eksempel:

$$e^1 \quad e$$

$$e^1. \quad 2.71828182846$$

Log-skabelon

ctrl **10^x** -tasten

log

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

Bemærk: Se også $\log()$, side 110.

Eksempel:

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

Stykkevis-skabelon (2 stykker)

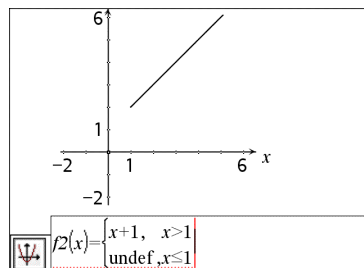
Katalog **>**

{ **}**

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 135.

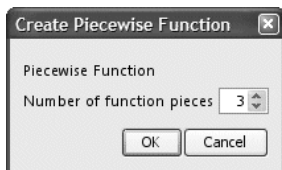
Eksempel:



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 135.

Eksempel:

Se eksemplet med stykkevis-skabelonen (2 stykker).

Skabelon til system med 2 ligninger

Katalog > 



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

Bemærk: Se også `system()`, side 186.

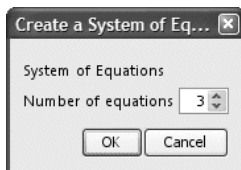
Eksempel:

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

Skabelon til system med N ligninger

Katalog > 

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



Bemærk: Se også `system()`, side 186.

Eksempel:

Se eksemplet med ligningssystemskabelonen (2-ligninger).

Absolut værdi-skabelon

Katalog > 



Bemærk: Se også `abs()`, side 8.

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:

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.

$$30^{\circ}15'10'' \quad \frac{10891 \cdot \pi}{64800}$$

Matrix-skabelon (2 x 2)

Katalog > 

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

Eksempel:

Opretter en matrix 2 x 2.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot a \quad \begin{bmatrix} a & 2 \cdot a \\ 3 \cdot a & 4 \cdot a \end{bmatrix}$$

Matrix-skabelon (1 x 2)

Katalog > 

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

Eksempel:

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

Matrix-skabelon (2 x 1)

Katalog > 

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

Eksempel:

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

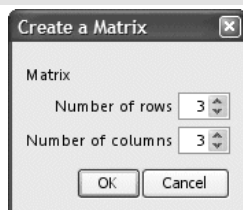
Matrix-skabelon (m x n)

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 (Σ)

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

Eksempel:

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

Bemærk: Se også $\Sigma()$ (**sumSeq**), side 231.

Produkt-skabelon (Π)

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

Eksempel:

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

Bemærk: Se også $\Pi()$ (**prodSeq**), side 230.

Skabelon til differentialkvotient af første orden

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

Eksempel:

$$\frac{d}{dx} (x^3) \quad 3 \cdot x^2$$

$$\frac{d}{dx} (x^3)_{x=3} \quad 27$$

Skabelonerne til differentialkvotienter af første orden kan også anvendes til at beregne differentialkvotienten af første orden i et punkt.

Skabelon til differentialkvotient af første orden

Katalog > 

Bemærk: Se også **d()** (differentialkvotient), side 227.

Skabelon til differentialkvotient af anden orden

Katalog > 

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

Skabelonerne til differentialkvotienter af anden orden kan også anvendes til at beregne differentialkvotienten af anden orden i et punkt.

Bemærk: Se også **d()** (differentialkvotient), side 227.

Eksempel:

$$\frac{d^2}{dx^2}(x^3) \quad 6 \cdot x$$

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

Skabelon til differentialkvotient af nte orden

Katalog > 

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

Skabelonen til differentialkvotienten af nte orden kan anvendes til at beregne differentialkvotienten af nte orden.

Bemærk: Se også **d()** (differentialkvotient), side 227.

Eksempel:

$$\frac{d^3}{dx^3}(x^3)|_{x=3} \quad 6$$

Bestemt integral skabelon

Katalog > 

$$\int_a^b \square dx$$

Bemærk: Se også **∫()** **integral()**, side 216.

Eksempel:

$$\int_a^b x^2 dx \quad \frac{b^3}{3} - \frac{a^3}{3}$$

Ubestemt integralskabelon

Katalog > 

$$\int \square dx$$

Eksempel:

Ubestemt integralskabelon

Katalog > 

Bemærk: Se også `∫()` `integral()`, side 216.

$$\int x^2 dx \quad \frac{x^3}{3}$$

Grænseværdi skabelon

Katalog > 

$$\lim_{x \rightarrow a} f(x)$$

Eksempel:

$$\lim_{x \rightarrow 5} (2 \cdot x + 3) = 13$$

Anvend – eller (–) til grænseværdi fra venstre. Anvend + til grænseværdi fra højre.

Bemærk: Se også `limit()`, side 100.

Alfabetisk oversigt

Elementer, hvis navne ikke er alfabetiske (som f.eks. +, ! og >), er anført sidst i dette afsnit, startende (side 216). 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(*UdtrI*) $\Rightarrow$ *udtryk*

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

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

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

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

$$\begin{bmatrix} |z| & |z| \\ |x+y \cdot i| & \sqrt{x^2+y^2} \end{bmatrix}$$

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*

$$\text{amortTbl}(12,60,10,5000,,12,12)$$

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 200).

- Hvis du udelader *Pmt*, sættes den som standard til *Pmt=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 231 og **bal()**, side 17.

and

Boolsk Udtr1 and Boolsk Udtr2 $\Rightarrow$ Boolsk udtryk

$$\begin{array}{ccc} x \geq 3 \text{ and } x \geq 4 & & x \geq 4 \\ \{x \geq 3, x \leq 0\} \text{ and } \{x \geq 4, x \leq -2\} & & \{x \geq 4, x \leq -2\} \end{array}$$

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:

$$\begin{array}{ccc} 0h7AC36 \text{ and } 0h3D5F & & 0h2C16 \end{array}$$

Vigtigt: Tallet nul, ikke bogstavet O.

I binær tilstand:

$$\begin{array}{ccc} 0b100101 \text{ and } 0b100 & & 0b100 \end{array}$$

I decimal tilstand:

$$\begin{array}{ccc} 37 \text{ and } 0b100 & & 4 \end{array}$$

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 talssystem).

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

angle()

angle(*Udtr1*) ⇒ *udtryk*

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

Bemærk: Alle udefinerede variable behandles som reelle variable.

I vinkeltilstanden Grader:

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

$$\text{angle}(1+i) \quad \frac{\pi}{4}$$

$$\text{angle}(z) \quad \frac{-\pi \cdot (\text{sign}(z)-1)}{2}$$

$$\text{angle}(x+i \cdot y) \quad \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right)$$

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

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 181.)

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 181.)

LevRow=0 for Blok

LevRow=2,3,...,Len-1, for to-faktor, hvor
 $Len = \text{length}(List1) = \text{length}(List2) = \dots = \text{length}(List10)$ 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.F Blok	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.F col	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.F row	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.dfInteract	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 > **approx(*Udtryk*)**⇒*udtryk*

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

►approxFraction()Katalog > *Expr* ►**approxFraction([*Tol*])**⇒*udtryk**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 > **approxRational(*Udtryk*, [*Tol*])**⇒*udtryk***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 32.
-----------------	-----------------------------

arccosh()	Se $\cosh^{-1}()$, side 34.
------------------	------------------------------

arccot()	Se $\cot^{-1}()$, side 35.
-----------------	-----------------------------

arccoth()	Se $\coth^{-1}()$, side 36.
------------------	------------------------------

arccsc()	Se $\csc^{-1}()$, side 38.
-----------------	-----------------------------

arccsch()	Se $\operatorname{csch}^{-1}()$, side 39.
------------------	--

arcLen()	Katalog > 
arcLen (<i>Udtr1</i> , <i>Var</i> , <i>Start</i> , <i>S</i>) $\Rightarrow$ <i>udtryk</i>	$\frac{\operatorname{arcLen}(\cos(x), x, 0, \pi)}{\operatorname{arcLen}(f(x), x, a, b)} = \frac{3.8202}{\int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} \, dx}$
Returnerer buelængden for <i>Udtr1</i> fra <i>Start</i> til <i>Slut</i> med hensyn til variabelen <i>Var</i>. Buelængden beregnes som et integral, der forudsætter er funktionsdefinition.	
arcLen (<i>Liste1</i> , <i>Var</i> , <i>Start</i> , <i>Slut</i>) $\Rightarrow$ <i>liste</i>	$\frac{\operatorname{arcLen}(\{\sin(x), \cos(x)\}, x, 0, \pi)}{\{3.8202, 3.8202\}}$
Returnerer en liste med buelængder for hvert element i <i>Liste1</i> fra <i>Start</i> til <i>Slut</i> med hensyn til <i>Var</i>.	

arcsec()	Se $\sec^{-1}()$, side 161.
-----------------	------------------------------

arcsech()Se $\text{sech}^{-1}()$, side 162.**arcsin()**Se $\sin^{-1}()$, side 172.**arcsinh()**Se $\sinh^{-1}()$, side 173.**arctan()**Se $\tan^{-1}()$, side 187.**arctanh()**Se $\tanh^{-1}()$, side 189.**augment()**Katalog > **augment(Liste1, Liste2) ⇒ liste** $\text{augment}(\{1, -3, 2\}, \{5, 4\}) \quad \{1, -3, 2, 5, 4\}$ 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*.

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

avgRC()	Katalog > 	
avgRC (<i>Udtryk1</i> , <i>Var</i> [=Værdi] [, <i>Trin</i>]) ⇒ <i>udtryk</i>	$\text{avgRC}(f(x), x, h)$	$\frac{f(x+h)-f(x)}{h}$
avgRC (<i>Udtryk1</i> , <i>Var</i> [=Værdi] [, <i>Liste1</i>]) ⇒ <i>liste</i>	$\text{avgRC}(\sin(x), x, h) _{x=2}$	$\frac{\sin(h+2)-\sin(2)}{h}$
avgRC (<i>Liste1</i> , <i>Var</i> [=Værdi] [, <i>Trin</i>]) ⇒ <i>liste</i>	$\text{avgRC}(x^2-x+2, x)$	$2 \cdot (x-0.4995)$
	$\text{avgRC}(x^2-x+2, x, 0.1)$	$2 \cdot (x-0.45)$
avgRC (<i>Matrix1</i> , <i>Var</i> [=Værdi] [, <i>Trin</i>]) ⇒ <i>matrix</i>	$\text{avgRC}(x^2-x+2, x, 3)$	$2 \cdot (x+1)$

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 "I" 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()

Katalog > 

bal(NPmt,N,I,PV,[Pmt],[FV],[PpY],[CpY],[PmtAt],[afrundVærdi])⇒værdi

bal(NPmt,amortTabel)⇒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 200).

NPmt angiver betalingsnummeret, hvorefter du vil have dataene beregnet.

bal(5,6,5.75,5000,,12,12)

tbl:=amortTbl(6,6,5.75,5000,,12,12)

00.0.5000.

1-23.35-825.634174.37

2-19.49-829.493344.88

3-15.62-833.362511.52

4-11.73-837.251674.27

5-7.82-841.16833.11

6-3.89-845.09-11.98

bal(4,tbl)

1674.27

$N, I, PV, Pmt, FV, PpY, CpY$ og $PmtAt$ er beskrevet i tabellen over TVM-argumenter (side 200).

- Hvis du udelader Pmt , bliver den som standard $Pmt=\mathbf{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$.
- 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 8.

Bemærk: Se også $\Sigma Int()$ og $\Sigma Prn()$, side 231.

►Base2

$Heltal1 \rightarrow Base2 \Rightarrow heltal$

256►Base2	0b100000000
0h1F►Base2	0b11111

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

Konverterer $Heltal1$ 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 $Heltal1$ som decimaltal (10- talssystem). 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^{63} 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

Heltal ►Base10⇒*heltal*

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

0b10011►Base10	19
0h1F►Base10	31

Konverterer *Heltal1* 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 hexadecimalt tal kan have op til 16.

Uden præfiks behandles *Heltal1* som decimaltal. Resultatet vises som decimaltal uanset tilstanden for talsystem.

Heltal1 ►Base16 ⇒ *heltal*

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

Konverterer *Heltal1* til et hexadecimalt tal. Binære eller hexadecimal tal har altid henholdsvis 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 hexadecimalt tal kan have op til 16.

Uden præfiks behandles *Heltal1* 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 18.

256►Base16	0h100
0b111100001111►Base16	0hF0F

binomCdf(n, p) $\Rightarrow$ *liste*

binomCdf($n, p, \text{nedreGrænse}, \text{øvreGrænse}$)
 $\Rightarrow$ *tal* hvis *nedreGrænse* og *øvreGrænse* er
 tal, *liste* if *nedreGrænse* og *øvreGrænse* er
 lister

binomCdf($n, p, \text{øvreGrænse}$) 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 binomialfordeling med n antal
 forsøg og sandsynligheden p for succes ved
 hvert forsøg.

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

binomPdf()

binomPdf(n, p) $\Rightarrow$ *liste*

binomPdf($n, p, [XVærdi]$) $\Rightarrow$ *tal* hvis *XVærdi*
 er et tal, *liste* hvis *XVærdi* er en liste

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

C

ceiling()

ceiling(*UdtrI*) $\Rightarrow$ *heltal*

$\text{ceiling}(.456)$	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*

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

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

centralDiff()

centralDiff(Udtr1,Var [=Værdi][,Trin])
⇒udtryk

centralDiff(Udtr1,Var [,Trin])
| Var=Værdi⇒udtryk

centralDiff(Udtr1,Var [=Værdi][,])⇒liste

centralDiff(1,Var [=Værdi][,Trin])⇒liste

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

Returnerer den numeriske differentialkvotient udregnet med formlen for den centrale differenskquotient.

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 **og d()**.

$$\frac{\text{centralDiff}(\cos(x),x,h)}{-\frac{(\cos(x-h)-\cos(x+h))}{2 \cdot h}}$$

$$\lim_{h \rightarrow 0} \left(\text{centralDiff}(\cos(x),x,h) \right) = -\sin(x)$$

$$\frac{\text{centralDiff}(x^3,x,0.01)}{3 \cdot (x^2+0.000033)}$$

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

$$\frac{\text{centralDiff}(x^2,x,\{0.01,0.1\})}{\{2 \cdot x, 2 \cdot x\}}$$

cFactor()

cFactor(Udtr1[,Var])⇒udtryk

cFactor(Liste1[,Var])⇒liste

cFactor(Matrix1[,Var])⇒matrix

cFactor(Udtr1) returnerer *Udtr1* opløst i faktorer med hensyn til alle dens variable over en fællesnævner.

$$\frac{\text{cFactor}(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x)}{a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i)}$$

$$\frac{\text{cFactor}\left(x^2 + \frac{4}{9}\right)}{\frac{(3 \cdot x - 2 \cdot i) \cdot (3 \cdot x + 2 \cdot i)}{9}}$$

$$\frac{\text{cFactor}(x^2 + 3)}{x^2 + 3}$$

$$\frac{\text{cFactor}(x^2 + a)}{x^2 + a}$$

Udtr1 opløses så meget som muligt i faktorer af 1. grad, også selvom dette indfører nye ikke-reelle tal. Denne mulighed er velegnet, hvis du ønsker opløsning i faktorer med hensyn til mere end en variabel.

cFactor(*Udtr1*,*Var*) returnerer *Udtr1* opløst i faktorer med hensyn til variabelen *Var*.

Udtr1 opløses så meget som muligt i faktorer mod faktorer af 1. grad i *Var*, med mulige ikke-reelle konstanter, også selvom det indfører irrationale konstanter eller deludtryk, der er irrationale i andre variable.

Faktorerne og deres led sorteres med *Var* som hovedvariabel. Ens potenser af *var* samles i hver faktor. Medtag *Var*, hvis opløsning i faktorer kun er nødvendig med hensyn til den pågældende variabel, og du er villig til at acceptere irrationale udtryk i alle andre variable for at øge opløsningen i faktorer med hensyn til *Var*. Der kan forekomme en uforudset faktoropløsning med hensyn til andre variable.

For automatisk indstilling af **Auto eller tilnærmet**-tilstanden tillader medtagningen af *Var* også en approksimation med koefficienter med flydende decimal, hvor irrationale koefficienter ikke kan udtrykkes eksplicit og koncist med de indbyggede funktioner. Også når der kun er en variabel, kan en medtagelse af *Var* give en mere komplet opløsning i faktorer.

Bemærk: Se også **factor()**.

$$\begin{aligned} \text{cFactor}(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x) &= a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i) \\ \text{cFactor}(x^2 + 3, x) &= (x + \sqrt{3} \cdot i) \cdot (x - \sqrt{3} \cdot i) \\ \text{cFactor}(x^2 + a, x) &= (x + \sqrt{a} \cdot i) \cdot (x - \sqrt{a} \cdot i) \end{aligned}$$

$$\begin{aligned} \text{cFactor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3) &= x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3 \\ \text{cFactor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3, x) &= (x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x + i) \cdot (x - i) \end{aligned}$$

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

char(*Heltal*) ⇒ *tegn*

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

$$\begin{aligned} \text{char}(38) &= "&" \\ \text{char}(65) &= "A" \end{aligned}$$

charPoly(kvadratMatrix, Var) $\Rightarrow$ polynomielt udtryk**charPoly(kvadratMatrix, Udtr)** $\Rightarrow$ polynomielt udtryk**charPoly(kvadratMatrix1, Matrix2)** $\Rightarrow$

polynomium udtryk

Returnerer det karakteristiske polynomium af *kvadratMatrix*. Det karakteristiske polynomium af en $n \times n$ matrix *A*, betegnet ved $p_A(\lambda)$, er polynomiet defineret ved

$$p_A(\lambda) = \det(\lambda \cdot I - A)$$

hvor *I* betegner $n \times n$ identitetsmatrixen.

kvadratMatrix1 og *kvadratMatrix2* skal have de samme dimensioner.

$m := \begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 3 & 0 \\ 2 & -1 & 0 \\ -2 & 2 & 5 \end{bmatrix}$
charPoly(<i>m</i> , <i>x</i>)	$-x^3 + 5 \cdot x^2 + 7 \cdot x - 35$
charPoly(<i>m</i> , $x^2 + 1$)	$-x^6 + 2 \cdot x^4 + 14 \cdot x^2 - 24$
charPoly(<i>m</i> , <i>m</i>)	0

χ^2 2way

 χ^2 2way *obsMatrix***chi22way** *obsMatrix*

Beregner en χ^2 test til association på tovejstabellen med tællinger i den observerede matrix *obsMatrix*. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 181.)

Oplysninger om effekten af tomme elementer i en matrix findes "Tomme (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat. χ^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

$\chi^2\text{Cdf}(\text{nedreGrænse}, \text{øvreGrænse}, df) \Rightarrow \text{tal}$
 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 χ^2 sandsynlighedsfordelingen
 mellem *nedreGrænse* og *øvreGrænse* for
 de angivne frihedsgrader *df*.

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

Oplysninger om effekten af tomme
 elementer i en liste findes "Tomme
 (ugyldige) elementer," side 243.

$\chi^2\text{GOF}$ *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. *obsList* er en liste med antal,
 og skal indeholde heltal. En sammenfatning
 af resultaterne lagres i *stat.results* variable.
 (side 181.)

Oplysninger om effekten af tomme
 elementer i en liste findes "Tomme
 (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat. χ^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

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

$\text{chi2Pdf}(XVal, df) \Rightarrow tal$ hvis $XVal$ er et tal,
liste, hvis $XVal$ er en liste

Beregner tæthedsfunktionen (pdf) for χ^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 243.

ClearAZ

Katalog >

ClearAZ

$5 \rightarrow b$	5
-------------------	---

Sletter alle enkelttegnvariable i det aktuelle opgaverum.

b	5
-----	---

ClearAZ	Done
---------	------

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock**, side 203.

b	b
-----	-----

ClrErr

Katalog

ClrErr

Se et eksempel på **ClrErr**, i Eksempel 2 under **Try**-kommandoen, side 196.

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 135, og **Try**, side 196.

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*.

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

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

Returnerer antallet af kolonner i *Matrix*.

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

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

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

comDenom()Katalog > **comDenom**(*Udtr1*[, *Var*]) $\Rightarrow$ *udtryk***comDenom**(*Liste1*[, *Var*]) $\Rightarrow$ *liste***comDenom**(*Matrix1*[, *Var*]) $\Rightarrow$ *matrix*

$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right)$
$\frac{x^2 \cdot y^2 + x^2 \cdot y + 2 \cdot x \cdot y^2 + 2 \cdot x \cdot y + 2 \cdot y^2 + 2 \cdot y}{x^2 + 2 \cdot x + 1}$

comDenom(**Udtryk1**) returnerer en forkortet brøk af en fuldt udviklet tæller over en fuldt udviklet nævner.

comDenom(Udtr1, Var) returnerer en brøk med fuldt udviklet tæller og nævner med hensyn til *Var*. Leddene og deres faktorer sorteres med *Var* som hovedvariabel. Ens potenser af *Var* samles. Der kan være tilfældige opløsninger i faktorer i de samlede koefficienter. Sammenlignet med at udelade *Var* sparer dette ofte tid, hukommelse og skærmpads, samtidig med, at det gør udtrykket mere læseligt. Det gør også de efterfølgende operationer på resultatet hurtigere og giver mindre risiko for at fylde hukommelsen op.

Hvis *Var* ikke forekommer i *Udtr1*, **comDenom(Udtr1, returnerer Var)** en brøk med uudviklet tæller og nævner. Sådanne resultater sparer normalt endnu mere tid, hukommelse og skærmpads. Sådanne resultater, der er delvist opløst i faktorer, gør også de efterfølgende operationer på resultatet hurtigere og mindre tilbøjelige til at fylde hukommelsen op.

Selv hvis der ikke er en nævner, er **comden-**funktionen ofte et hurtigt middel til en delvis faktoropløsning, hvis **factor()** er for langsom eller bruger hele hukommelsen.

Tip: Indsæt denne **comden()**-funktionsdefinition og prøv den rutinemæssigt som et alternativ til **comDenom()** og **factor()**.

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y, x\right) \\ \frac{x^2 \cdot y \cdot (y+1) + 2 \cdot x \cdot y \cdot (y+1) + 2 \cdot y \cdot (y+1)}{x^2 + 2 \cdot x + 1}$$

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y, y\right) \\ \frac{y^2 \cdot (x^2 + 2 \cdot x + 2) + y \cdot (x^2 + 2 \cdot x + 2)}{x^2 + 2 \cdot x + 1}$$

Define *comden(exprn)=comDenom(exprn,abc)*
Done

$$\text{comden}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right) \frac{(x^2+2 \cdot x+2) \cdot y \cdot (y+1)}{(x+1)^2}$$

$$\text{comden}(1234 \cdot x^2 \cdot (y^3-y) + 2468 \cdot x \cdot (y^2-1)) \\ 1234 \cdot x \cdot (x \cdot y + 2) \cdot (y^2-1)$$

completeSquare ()

completeSquare(ExprOrEqn, Var)
⇒udtryk eller ligning

completeSquare(ExprOrEqn, Var^Power)
⇒udtryk eller ligning

completeSquare(ExprOrEqn, Var1, Var2 [...])⇒udtryk eller ligning

completeSquare(ExprOrEqn, {Var1, Var2 [...]})⇒udtryk eller ligning

$$\text{completeSquare}(x^2+2 \cdot x+3, x) \quad (x+1)^2+2$$

$$\text{completeSquare}(x^2+2 \cdot x=3, x) \quad (x+1)^2=4$$

$$\text{completeSquare}(x^6+2 \cdot x^3+3, x^3) \quad (x^3+1)^2+2$$

$$\text{completeSquare}(x^2+4 \cdot x+y^2+6 \cdot y+3=0, x, y) \\ (x+2)^2+(y+3)^2=10$$

completeSquare ()

Katalog > 

Konverterer et kvadratisk, polynomielt udtryk af formen $a \cdot x^2 + b \cdot x + c$ til formen $a \cdot (x-h)^2 + k$

— eller —

Konverterer en kvadratisk ligning af formen $a \cdot x^2 + b \cdot x + c = d$ til formen $a \cdot (x-h)^2 = k$

Det første argument skal være et kvadratisk udtryk eller en ligning på standardform med hensyn til det andet argument.

Det andet argument skal være en enkelt variabel eller en enkelt variabel opløftet til en rationel potens, f.eks. x, y^2 eller $z^{(1/3)}$.

Den tredje eller fjerde del af syntaksen forsøger at kvadratiskekomplettere med hensyn til variable *Var1*, *Var2* [...]).

$$\text{completeSquare}\left(3 \cdot x^2 + 2 \cdot y + 7 \cdot y^2 + 4 \cdot x = 3, \{x, y\}\right) \\ 3 \cdot \left(x + \frac{2}{3}\right)^2 + 7 \cdot \left(y + \frac{1}{7}\right)^2 = \frac{94}{21}$$

$$\text{completeSquare}\left(x^2 + 2 \cdot x \cdot y, x, y\right) \quad (x+y)^2 - y^2$$

conj()

Katalog > 

conj(Udtr1) ⇒ *udtryk*

conj(Liste1) ⇒ *liste*

conj(Matrix1) ⇒ *matrix*

Returnerer kompleks konjugerede af argumentet.

Bemærk: Alle udefinerede variable behandles som reelle variable.

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

constructMat()

Katalog > 

constructMat
(*Udtr*, *Var1*, *Var2*, *antalRækker*, *antalKol*)
⇒ *matrix*

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*.

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

Var1 er automatisk forøget fra **1** til *antalRækker*. Inden for hver række, *Var2* er forøget fra **1** til *antalKol*.

CopyVar

Katalog >

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 værdi

CopyVar <i>a</i> , <i>c</i> : $c(4)$	$\frac{1}{4}$
--------------------------------------	---------------

CopyVar <i>b</i> , <i>c</i> : $c(4)$	16
--------------------------------------	----

Hvis *Var1* er navnet på en eksisterende brugerdefineret funktion, kopieres definitionen af denne funktion til funktionen *Var2*. Funktionen *Var1* skal defineres.

Var1 skal opfylde kravene til navngivning af variable, eller være et indirekte udtryk, der kan reduceres til et variabelnavn, der opfylder betingelserne.

CopyVar *Var1*, *Var2*. kopierer alle elementer af *Var1*. variabelgruppen til *Var2*. gruppen, og opretter *Var2*. hvis nødvendigt.

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

<i>aa.b</i> :=6.78	6.78
--------------------	------

CopyVar <i>aa</i> , <i>bb</i> .	Done
---------------------------------	------

Var1. skal være navnet på en eksisterende variabelgruppe, så som statistikken *stat.nn* resultater eller variable dannet ved brug af **LibShortcut()**-funktionen. Hvis *Var2*.

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

allerede eksisterer, vil denne kommando udskifte alle elementer, der er fælles i begge grupper, og tilføje de elementer, som ikke allerede eksisterer. Hvis et eller flere elementer i *Var2*. er låst, efterlades alle elementer i *Var2*. uændret.

corrMat()

Katalog >

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

Beregner korrelationsmatricen for den udvidede matrix [*Liste1* *Liste2* ... *Liste20*].

Udtr ►cos

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

Repræsenterer *Expr* i termer af cosinus. Det er en konverteringsoperator for visning. Denne operator kan kun anvendes ved slutningen af indtastningslinjen.

►cos reducerer alle potenser af

$$\sin(\dots) \text{ modulo } 1 - \cos(\dots)^2$$

således, at alle tilbageværende potenser af $\cos(\dots)$ har eksponenter i området (0, 2). Således vil resultatet være uden $\sin(\dots)$ hvis, og kun hvis $\sin(\dots)$ ku optræder i det givne udtryk med lige potenser.

Bemærk: Konverteringsoperatorer understøttes ikke i vinkeltilstandene Grader eller Nygrader. Før brug skal man sikre at vinkeltilstanden er indstillet til radianer, og at *Udtr* ikke indeholder eksplicit reference til grader eller nygrader.

$$\left(\sin(x)\right)^2 \blacktriangleright \cos \qquad 1 - \left(\cos(x)\right)^2$$

cos()

-tast

cos(*Udtr1*) ⇒ *udtryk*

I vinkeltilstanden Grader:

cos(*Liste1*) ⇒ *list*

cos(*Udtr1*) returnerer cosinus af argumentet som et udtryk.

cos(*Liste1*) returnerer en liste med cosinus 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.

$$\begin{array}{l} \cos\left(\frac{\pi}{4}\right) \qquad \qquad \qquad \sqrt{2} \\ \qquad \qquad \qquad \qquad \qquad \qquad 2 \\ \cos(45) \qquad \qquad \qquad \sqrt{2} \\ \qquad \qquad \qquad \qquad \qquad \qquad 2 \\ \cos(\{0,60,90\}) \qquad \qquad \left\{1, \frac{1}{2}, 0\right\} \end{array}$$

I vinkeltilstanden Nygrader:

$$\cos(\{0,50,100\}) \qquad \qquad \left\{1, \frac{\sqrt{2}}{2}, 0\right\}$$

I vinkeltilstanden Radian:

$\cos\left(\frac{\pi}{4}\right)$	$\frac{\sqrt{2}}{2}$
$\cos(45^\circ)$	$\frac{\sqrt{2}}{2}$

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:

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

cos⁻¹(Udtr1)⇒udtryk

I vinkeltilstanden Grader:

cos⁻¹(Liste1)⇒liste

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

$\cos^{-1}(Udtr1)$ returnerer den vinkel, hvis cosinus er *Udtr1* som et udtryk.

$\cos^{-1}(Liste1)$ returnerer en liste med de inverse cosinusværdier for hvert element af *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 **arccos (...)**.

$\cos^{-1}(kvadratMatrix1) \Rightarrow kvadratMatrix$

Returnerer den matrixinverse cosinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne den inverse cosinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

$$\cos^{-1}\{0,0,2,0,5\} \quad \left\{ \frac{\pi}{2}, 1.36944, 1.0472 \right\}$$

I vinkeltilstanden radian og rektangulært komplekst format:

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

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

$\cosh(Udtr1) \Rightarrow udtryk$

I vinkeltilstanden Grader:

$\cosh(Liste1) \Rightarrow liste$

$$\cosh \left(\left(\frac{\pi}{4} \right)_r \right) \quad \cosh(45)$$

$\cosh(Udtr1)$ returnerer den hyperbolske cosinus af argumentet som et udtryk.

$\cosh(Liste1)$ returnerer en liste med hyperbolsk cosinus for hvert element i *Liste1*.

$\cosh(kvadratMatrix1) \Rightarrow kvadratMatrix$

I vinkeltilstanden Radian:

Returnerer matrix hyperbolsk cosinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne den hyperbolske cosinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

cosh()

Katalog >

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

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

cosh⁻¹()

Katalog >

cosh⁻¹(Udtr1) ⇒ udtryk

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

cosh⁻¹(Liste1) ⇒ liste

$$\cosh^{-1}(\{1, 2, 1, 3\}) = \{0, 1.37286, \cosh^{-1}(3)\}$$

cosh⁻¹(Udtr1) returnerer den inverse hyperbolske cosinus af argumentet som et udtryk.

cosh⁻¹(Liste1) returnerer en liste med de inverse hyperbolske cosinusværdier for hvert element i *Liste1*.

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

cosh⁻¹(kvadratMatrix1) ⇒ kvadratMatrix

Returnerer den matrixinverse hyperbolske cosinus af *kvadratMatrix1*. 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:

$$\cosh^{-1} \left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \right) = \begin{bmatrix} 2.52503+1.73485 \cdot i & -0.009241-1.4908i & 0.486969-0.725533 \cdot i \\ 0.486969-0.725533 \cdot i & 1.66262+0.623491i & -0.322354-2.08316 \cdot i \\ -0.322354-2.08316 \cdot i & 1.26707+1.79018i & \end{bmatrix}$$

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

cot()

-tast

cot(Udtr1) ⇒ udtryk

I vinkeltilstanden Grader:

cot(Liste1) ⇒ liste

$$\cot(45) = 1$$

cot()

-tast

Returnerer cotangens til *Udtr1* 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:

$\cot(50)$	1
------------	---

I vinkeltilstanden Radian:

$\cot(\{1,2,1,3\})$	$\left\{ \frac{1}{\tan(1)}, -0.584848, \frac{1}{\tan(3)} \right\}$
---------------------	--

cot⁻¹()

-tast

cot⁻¹(*Udtr1*)⇒*udtryk*

I vinkeltilstanden Grader:

cot⁻¹(*Liste1*)⇒*liste*

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

Returnerer den vinkel, hvis cotangens er *Udtr1*, eller returnerer en liste med den inverse cotangens til hvert element i *Liste1*.

I vinkeltilstanden Nygrader:

$\cot^{-1}(1)$	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 **arccot (...)**.

$\cot^{-1}(1)$	$\frac{\pi}{4}$
----------------	-----------------

coth()

Katalog > 

coth(*Udtr1*)⇒*udtryk*

$\coth(1.2)$	1.19954
--------------	---------

coth(*Liste1*)⇒*liste*

$\coth(\{1,3,2\})$	$\left\{ \frac{1}{\tanh(1)}, 1.00333 \right\}$
--------------------	--

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

coth⁻¹()	Katalog > 
coth⁻¹(Udtr1) ⇒ <i>udtryk</i>	$\text{coth}^{-1}(3,5)$ 0.293893
coth⁻¹(Liste1) ⇒ <i>liste</i>	$\text{coth}^{-1}(\{-2,2,1,6\})$
Returnerer den inverse hyperbolske cotangens til <i>Udtr1</i> eller returnerer en liste med den inverse hyperbolske cotangens til alle elementer i <i>Liste1</i> .	$\left\{ \frac{-\ln(3)}{2}, 0,518046, \frac{\ln\left(\frac{7}{5}\right)}{2} \right\}$
Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive arccoth (...) .	

count()	Katalog > 
count(Værdi1ellerListe1 [,Værdi2ellerListe2 [...]]) ⇒ <i>værdi</i>	$\text{count}(2,4,6)$ 3
Returnerer det akkumulerede antal af alle elementer i argumenterne, der evalueres til numeriske værdier.	$\text{count}(\{2,4,6\})$ 3
Hvert argument kan være et udtryk, en værdi, en liste eller en matrix. Du kan blande datatyper og anvende argumenter med forskellige dimensioner.	$\text{count}\left(2, \{4,6\}, \begin{bmatrix} 8 & 10 \\ 12 & 14 \end{bmatrix}\right)$ 7
For lister, matricer eller celleområder evalueres hvert element for at bestemme, om det skal inkluderes i tællingen.	$\text{count}\left(\frac{1}{2}, 3+4*i, \text{undef}, "hello", x+5, \text{sign}(0)\right)$ 2
I applikationen Lister og regneark kan du anvende et celleområde i stedet for ethvert argument.	I det sidste eksempel tælles kun 1/2 og $3+4*i$. De resterende argumenter evalueres ikke til numeriske værdier, hvis det antages at x er udefineret.
Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 243.	

countif()	Katalog > 
countif(Liste,Kriterie) ⇒ <i>værdi</i>	$\text{countif}(\{1,3,"abc",\text{undef},3,1\},3)$ 2
Returnerer det akkumulerede antal af alle elementer i <i>Liste</i> , der opfylder de angivne <i>Kriterie</i> .	Tæller antallet af elementer lig med 3.
<i>Kriterie</i> kan være:	$\text{countif}(\{"abc","def","abc",3\}, "def")$ 1
En værdi, et udtryk eller en streng. For	Tæller antallet af elementer lig med "def."

eksempel tæller **3** kun de elementer i *Liste*, der reduceres til værdien 3.

- 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 243.

Bemærk: Se også **sumlf()**, side 185, og **frequency()**, side 77.

$$\text{countIf}\left(\left\{x^{-2}, x^{-1}, 1, x, x^2\right\}, x\right) \quad 1$$

Tæller antallet af elementer lig med x ; dette eksempel antager, at variabelen x er udefineret.

$$\text{countIf}\left(\left\{1, 3, 5, 7, 9\right\}, ? < 5\right) \quad 2$$

Tæller 1 og 3.

$$\text{countIf}\left(\left\{1, 3, 5, 7, 9\right\}, 2 < ? < 8\right) \quad 3$$

Tæller 3, 5 og 7.

$$\text{countIf}\left(\left\{1, 3, 5, 7, 9\right\}, ? < 4 \text{ or } ? > 6\right) \quad 4$$

Tæller 1, 3, 7 og 9.

cPolyRoots()

cPolyRoots(*Poly*, *Var*) $\Rightarrow$ *liste*

cPolyRoots(*ListeAfKoeff*) $\Rightarrow$ *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 en variabel.

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

Bemærk: Se også **polyRoots()**, side 140.

$$\text{polyRoots}\left(y^3 + 1, y\right) \quad \{-1\}$$

$$\text{cPolyRoots}\left(y^3 + 1, y\right) \quad \left\{-1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i\right\}$$

$$\text{polyRoots}\left(x^2 + 2 \cdot x + 1, x\right) \quad \{-1, -1\}$$

$$\text{cPolyRoots}\left(\{1, 2, 1\}\right) \quad \{-1, -1\}$$

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.

crossP(Vektor1, Vektor2) ⇒ vektor

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.

$$\text{crossP}(\{a1, b1\}, \{a2, b2\})$$

$$\{0, 0, a1 \cdot b2 - a2 \cdot b1\}$$

$$\text{crossP}(\{0, 1, 2, 2, -5\}, \{1, -0.5, 0\})$$

$$\{-2.5, -5, -2.25\}$$

$$\text{crossP}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \begin{bmatrix} -3 & 6 & -3 \\ 0 & 0 & -2 \end{bmatrix}\right)$$

csc() -tast**csc(Udtr1) ⇒ udtryk**

I vinkeltilstanden Grader:

csc(Liste1) ⇒ liste

Returnerer cosecansen til *Udtr1* eller returnerer en liste med cosecansen til alle elementer i *Liste1*.

$$\text{csc}(45) \quad \sqrt{2}$$

I vinkeltilstanden Nygrader:

$$\text{csc}(50) \quad \sqrt{2}$$

I vinkeltilstanden Radian:

$$\text{csc}\left(\left\{1, \frac{\pi}{2}, \frac{\pi}{3}\right\}\right) \quad \left\{\frac{1}{\sin(1)}, 1, \frac{2\sqrt{3}}{3}\right\}$$

csc⁻¹() -tast**csc⁻¹(Udtr1) ⇒ udtryk**

I vinkeltilstanden Grader:

csc⁻¹(Liste1) ⇒ liste

$$\text{csc}^{-1}(1) \quad 90.$$

Returnerer den vinkel, hvis cosecans er *Udtr1*, eller returnerer en liste med den inverse cosecans til de enkelte elementer på *Liste1*.

I vinkeltilstanden Nygrader:

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

trig **-tast**

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

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

100.

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

I vinkeltilstanden Radian:

$$\text{csc}^{-1}\left(\left\{1,4,6\right\}\right) \quad \left\{\frac{\pi}{2}, \sin^{-1}\left(\frac{1}{4}\right), \sin^{-1}\left(\frac{1}{6}\right)\right\}$$

$\text{csch}()$

Katalog >

 $\text{csch}(\text{Udtr1}) \Rightarrow \text{udtryk}$

$$\text{csch}(3) \quad \frac{1}{\sinh(3)}$$

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

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

$$\text{csch}\left(\left\{1,2,1,4\right\}\right) \quad \left\{\frac{1}{\sinh(1)}, 0.248641, \frac{1}{\sinh(4)}\right\}$$

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

Katalog >

 $\text{csch}^{-1}(\text{Udtr1}) \Rightarrow \text{udtryk}$

$$\text{csch}^{-1}(1) \quad \sinh^{-1}(1)$$

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

Returnerer den hyperbolske cosecans til *Udtr1* eller returnerer en liste med den inverse hyperbolske cosecans til hvert element i *Liste1*.

$$\text{csch}^{-1}\left(\left\{1,2,1,3\right\}\right) \quad \left\{\sinh^{-1}(1), 0.459815, \sinh^{-1}\left(\frac{1}{3}\right)\right\}$$

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

$\text{cSolve}()$

Katalog >

 $\text{cSolve}(\text{Ligning}, \text{Var}) \Rightarrow \text{Boolsk udtryk}$

$$\text{cSolve}(x^3=1, x)$$

 $\text{cSolve}(\text{Ligning}, \text{Var}=\text{Gæt}) \Rightarrow \text{Boolesk udtryk}$

$$x=\frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i \text{ or } x=\frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i \text{ or } x=1$$

 $\text{cSolve}(\text{Ulighed}, \text{Var}) \Rightarrow \text{Boolsk udtryk}$

$$\text{solve}(x^3=1, x) \quad x=1$$

Returnerer komplekse løsninger til en ligning eller ulighed for *Var*. Målet er at finde alle reelle og ikke-reelle løsninger. Også selvom *Ligning* er reel, tillader **cSolve()** ikke-reelle resultater i den reelle tilstand.

cSolve() indstiller midlertidigt til det komplekse domæne under løsningen, også selvom det aktuelle domæne er det reelle. I det komplekse domæne anvender brøkpotenser med ulige nævnere hovedområdet i stedet for det reelle område. Som følge heraf er løsninger fra **solve()** af ligninger med sådanne brøkpotenser ikke nødvendigvis en delmængde af løsningerne fra **cSolve()**.

cSolve() starter med eksakte symbolske metoder. Med undtagelse af **Eksakt** tilstand anvender **cSolve()** om nødvendigt også en iterativ tilnærmet kompleks polynomiel faktoropløsning.

Bemærk: Se også **cZeros()**, **solve()**, og **zeros()**.

$\text{cSolve}\left(x^{\frac{1}{3}}=1,x\right)$	false
$\text{solve}\left(x^{\frac{1}{3}}=1,x\right)$	$x=1$

I Vis cifre-tilstand for Fast 2:

$\text{exact}\left(\text{cSolve}\left(x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3=0,x\right)\right)$
$x\cdot\left(x^4+4\cdot x^3+5\cdot x^2-6\right)=3$
$\text{cSolve}\left(\text{Ans},x\right)$
$x=-1.11+1.07\cdot i$ or $x=-1.11-1.07\cdot i$ or $x=-2.9$

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

cSolve(*Eqn1* and *Eqn2* [and...],

VarEllerGæt1, *VarEllerGæt2* [, ...])
 $\Rightarrow$ Boolesk udtryk

cSolve(*SystemAfLign*, *VarEllerGæt1*,
VarEllerGæt2 [, ...]) $\Rightarrow$ Boolesk udtryk

Returnerer mulige komplekse løsninger til de sammenhørende algebraiske ligninger, hvor hvert *VarEllerGæt* angiver en variabel, du vil løse for.

Du kan også vælge at angive et initielt gæt til en variabel. Hvert *VarEllerGæt* skal have formen:

variabel

– eller –

variabel = reelt eller ikke-reelt tal

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

Hvis alle ligningerne er polynomier, og hvis du IKKE angiver nogle initiale gæt, benytter **cSolve()** den leksikale Gröbner/Buchberger eliminationsmetode som forsøger på at bestemme **alle** komplekse løsninger.

Komplekse løsninger kan omfatte både reelle og ikke-reelle løsninger som i eksemplet til højre.

$$\text{cSolve}\left(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v\}\right) \\ u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ or } u = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i$$

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

Sammenhørende polynomielle ligninger kan have ekstra variable, der ikke har nogen værdier men repræsenterer givne numeriske værdier, der kan substitueres efterfølgende.

$$\text{cSolve}\left(u \cdot v - u = c \cdot v \text{ and } v^2 = -u, \{u, v\}\right) \\ u = \frac{-\left(\sqrt{4 \cdot c - 1} \cdot i + 1\right)^2}{4} \text{ and } v = \frac{\sqrt{4 \cdot c - 1} \cdot i + 1}{2} \circ$$

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

Du kan også medtage løsningsvariable, der ikke optræder i ligningerne. Disse løsninger viser, hvordan løsningsfamilier kan indeholde arbitrære konstanter af formen c_k , hvor k er et heltalssuffix fra 1 til og med 255.

$$\text{cSolve}\left(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v, w\}\right) \\ u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } w = c_{43} \text{ or } u = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } w = c_{43}$$

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

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbrug stærkt af den rækkefølge løsningsvariablene angives i. Hvis det initielle valg kræver for meget hukommelse eller tålmodighed, skal du prøve at omarrangere variablene i ligningerne og/eller *varEllerGæt* listen.

Hvis du ikke medtager nogen gæt, og hvis en ligning er ikke-polynomiell i en variabel, men alle ligninger er lineære i alle løsningsvariable, anvender **cSolve()** en Gauss-eliminering i et forsøg på at bestemme alle løsninger.

$$\text{cSolve}\left(u + v = e^w \text{ and } u - v = i, \{u, v\}\right) \\ u = \frac{e^w + i}{2} \text{ and } v = \frac{e^w - i}{2}$$

Hvis et system hverken er polynomielt i alle variable eller lineært i sine løsningsvariable, bestemmer **cSolve()** højst en løsning med en iterativ approksimationsmetode. Dette gøres ved at lade antallet af løsningsvariable være lig med antallet af ligninger og reducere alle andre variable i ligningerne til tal.

Et ikke-reelt gæt er ofte nødvendigt for at bestemme en ikke-reel løsning. For at opnå konvergens skal et gæt være meget tæt på en løsning.

$$\text{cSolve}(e^z = w \text{ and } w = z^2, \{w, z\})$$

$w = 0.494866 \text{ and } z = 0.703467$

$$\text{cSolve}(e^z = w \text{ and } w = z^2, \{w, z = 1 + i\})$$

$w = 0.149606 + 4.8919 \cdot i \text{ and } z = 1.58805 + 1.5402 \cdot i$

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

CubicReg

CubicReg $X, Y, [Frekv] [, Kategori, Medtag]$

Beregner polynomielt 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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

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 ²	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>Hyppighed</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>Hyppighed</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()

Katalog > 

cumulativeSum(Liste1) ⇒ liste

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

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.

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

Et tomt (ugyldigt) element i *Liste1* eller *Matrix1* giver et ugyldigt element i den resulterende liste eller matrix. Yderligere oplysninger om tomme elementer findes på side 243.

Cycle

Katalog > 

Cyklus

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

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

Cycle må 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.

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

$g()$	5000
-------	------

►Cylind

Vektor ►Cylind

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.

$\begin{bmatrix} 2 & 2 & 3 \end{bmatrix}$ ►Cylind	$\begin{bmatrix} 2\sqrt{2} & \angle \frac{\pi}{4} & 3 \end{bmatrix}$
---	--

cZeros()

cZeros(Udtr, Var)⇒liste

Returnerer en liste med mulige reelle og ikke-reelle værdier for Var som løser $Udtr=0$. **cZeros()** gør dette ved at beregne **exp►liste(cSolve(Udtr=0,Var),Var)**. Ellers ligner **cZeros()** **zeros()**.

Bemærk: Se også **cSolve()**, **solve()** og **zeros()**.

**cZeros({Udtr1, Udtr2 [, ...] },
{varEllerGæt1,varEllerGæt2 [, ...] })**
⇒matrix

Returnerer mulige positioner, hvor udtrykkene samtidigt er nul. Hver *varEllerGæt* angiver en ubekendt, hvis værdi du søger.

I Vis cifre-tilstand med Fast 3:

$cZeros(x^5+4x^4+5x^3-6x-3,x)$
$\{-1.1138+1.07314i, -1.1138-1.07314i, -2\}$

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

Du kan også vælge at angive et initielt gæt til en variabel. Hvert *varEllerGæt* skal have formen:

variabel

– eller –

variabel = reelt eller ikke-reelt tal

For eksempel er x gyldig, og det er $x=3+iosg\ddot{a}$.

Hvis alle ligningerne er polynomier, og hvis du IKKE angiver nogle initielle gæt, benytter **cZeros()** den leksikale Gröbner/Buchberger eliminationsmetode som forsøg på at bestemme **alle** komplekse nulpunkter.

Komplekse nulpunkter kan omfatte både reelle og ikke-reelle nulpunkter som i eksemplet til højre.

Hver række i den resulterende matrix repræsenterer et nulpunkt med komponenterne arrangeret på samme måde som *varEllerGæt*-listen. Du kan udtrække en række og indeksere matricen efter [*række*].

Sammenhørende polynomier kan have ekstra variable, der ikke har nogen værdier men repræsenterer givne numeriske værdier, der kan substitueres efterfølgende.

Du kan også medtage ubekendte variable, der ikke optræder i udtrykkene. Disse nulpunkter viser, hvordan nulpunktsfamilier kan indeholde arbitrære konstanter af formen c_k , hvor k er et heltalssuffiks fra 1 til og med 255.

$$cZeros\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \{u, v\}\right) \begin{bmatrix} 0 & 0 \\ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \end{bmatrix}$$

Udtræk række 2:

$$Ans[2] \begin{bmatrix} \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \end{bmatrix}$$

$$cZeros\left(\left\{u \cdot v - u - c \cdot v^2, v^2 + u\right\}, \{u, v\}\right) \begin{bmatrix} 0 & 0 \\ -(c-1)^2 & -(c-1) \end{bmatrix}$$

$$cZeros\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \{u, v, w\}\right) \begin{matrix} cZero\left(\left\{u \cdot (v-1) - v, u + v^2\right\}, \{u, v, w\}\right) \\ \begin{bmatrix} 0 & 0 & c\# \\ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & c\# \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & c\# \end{bmatrix} \end{matrix}$$

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbruget stærkt af den rækkefølge, de ubekendte angives i. Hvis det initielle valg kræver for meget hukommelse eller tålmodighed, skal du prøve at omarrangere variablene i udtrykkene og/eller *varEllerGæt* listen.

Hvis du ikke medtager nogen gæt, og hvis en ligning er ikke-polynomiell i en variabel, men alle udtryk er lineære i alle ubekendte, anvender **cZeros()** en Gauss-eliminering i et forsøg på at bestemme alle nulpunkter.

$$\left| \begin{array}{c} \text{cZeros}\left(\left\{u+v-e^w, u-v-i\right\}, \left\{u, v\right\}\right) \\ \left[\frac{e^w+i}{2} \quad \frac{e^w-i}{2} \right] \end{array} \right|$$

Hvis et system hverken er polynomielt i alle variable eller lineært i sine ubekendte, bestemmer **cZeros()** højst et nulpunkt med en iterativ approksimationsmetode. Dette gøres ved at lade antallet af ubekendte være lig med antallet af udtryk og forkorte alle andre variable i udtrykkene til tal.

$$\left| \begin{array}{c} \text{cZeros}\left(\left\{e^z-w, w-z^2\right\}, \left\{w, z\right\}\right) \\ [0.494866 \quad -0.703467] \end{array} \right|$$

Et ikke-reelt gæt er ofte nødvendigt for at bestemme ikke-reelle nulpunkter. For at opnå konvergens skal et gæt være meget tæt på et nulpunkt.

$$\left| \begin{array}{c} \text{cZeros}\left(\left\{e^{-z}-w, w-z^2\right\}, \left\{w, z=1+i\right\}\right) \\ [0.149606+4.8919 \cdot i \quad 1.58805+1.54022 \cdot i] \end{array} \right|$$

D

dbd()

Katalog >

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.

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

MM.DDÅÅ (almindeligt format i USA)

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

DDMM.ÅÅ (almindeligt format i Europa)

►DD

Katalog > *Tal* ►DD⇒*værdi*

I vinkeltilstanden Grader:

Liste ►DD⇒*liste**Matrix* ►DD⇒*matrix*

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

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.

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

$\{1.5\}$ ►DD	85.9437°
---------------	-----------------

►Decimal

Katalog > *Udtr* ►Decimal⇒*udtryk**Liste* ►Decimal⇒*udtryk**Matrix* ►Decimal⇒*udtryk*

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.

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

Define

Katalog > **Define** *Var* = *Udtryk*

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

Definerer variablen *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**).

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 49 og **Define LibPub**, side 49.

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

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

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

Define LibPriv *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 47, og **Define LibPub**, side 49.

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 47 og **Define LibPriv**, side 49.

DelVar

Katalog > 

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

2 → a2

DelVar Var.

(a+2)²16

Sletter de angivne variable, eller variabelgruppe fra hukommelse.

DelVar aDone

(a+2)²(a+2)²

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock**, side 203

DelVar Var. sletter alle elementer i Var. variabelgruppe(så som statistikken stat.nn resultater, eller variable dannet ved brug af LibShortcut()-funktionen). Punktummet (.) i denne form af DelVar -kommandoen begrænser den til at slette en variabelgruppe: den simple variabel Var berøres ikke.

aa.a:=4545

aa.b:=5.675.67

aa.c:=78.978.9

getVarInfo()

aa.a "NUM" 

aa.b "NUM" 

aa.c "NUM" 

DelVar aa.Done

getVarInfo() "NONE"

delVoid()	Catalog > 	
delVoid (<i>Liste1</i>)⇒ <i>liste</i>	delVoid({1,void,3})	{1,3}
Returnerer en liste med indholdet i <i>Liste1</i> med alle tomme (ugyldige) elementer fjernet.		
Yderligere oplysninger om tomme elementer findes på side 243.		

deSolve()

Katalog > 

deSolve(1.Eller2.OrdenODE, Var, afhVar)
 $\Rightarrow$ en generel løsning

Returnerer en ligning, der eksplicit eller implicit angiver en generel løsning til en 1. eller anden ordens ordinær differentialligning (ODE). I ODE'en:

- Anvend et mærketegn (tryk på ) for at betegne differentialkvotienten af første orden af den afhængige variabel med hensyn til den uafhængige variable
- Anvend to mærketegn for at betegne den tilsvarende anden afledede.

Mærketegnet ' anvendes kun til differentialkvotienter i deSolve(). I andre tilfælde anvendes d().

Den generelle løsning til en 1. grads ligning indeholder en arbitrær konstant af formen c_k, hvor k er et heltalssuffiks fra 1 til og med 255. Løsningen af en 2. ordens differentialligning indeholder to af disse konstanter.

solve() anvendes på en implicit løsning, hvis du vil prøve at omregne den til en eller flere ækvivalente eksplicitte løsninger.

Ved sammenligning af dine resultater med tekstbogen eller manuelt frembragte løsninger skal du være opmærksom på, at forskellige metoder indfører arbitrære konstanter på forskellige steder i beregningen, hvilket kan frembringe forskellige generelle løsninger.

$$\begin{aligned} &\text{deSolve}(y''+2\cdot y'+y=x^2, x, y) \\ &\quad y=(c3\cdot x+c4)\cdot e^{-x}+x^2-4\cdot x+6 \\ &\text{right}(Ans) \rightarrow temp \quad (c3\cdot x+c4)\cdot e^{-x}+x^2-4\cdot x+6 \\ &\frac{d^2}{dx^2}(temp)+2\cdot \frac{d}{dx}(temp)+temp-x^2 \quad 0 \\ &\text{DelVar temp} \quad Done \end{aligned}$$

$$\begin{aligned} &\text{deSolve}(y'=(\cos(y))^2\cdot x, x, y) \quad \tan(y)=\frac{x^2}{2}+c4 \\ &\text{solve}(Ans, y) \quad y=\tan^{-1}\left(\frac{x^2+2\cdot c4}{2}\right)+n3\cdot \pi \\ &\text{Ans}|c4=c-1 \text{ and } n3=0 \quad y=\tan^{-1}\left(\frac{x^2+2\cdot (c-1)}{2}\right) \end{aligned}$$

deSolve(1.OrdenODEandstartBeting, Var, afhVar) ⇒ en partikulær løsning

Returnerer en partikulær løsning, som opfylder 1.OrdenODE og startBeting. Dette er normalt nemmere end at bestemme en generel løsning, substituere startværdier, løse for den arbitrære konstant og derefter substituere denne værdi ind i den generelle løsning.

startBeting er en ligning på formen:

afhVar (uafhængigStartværdi) = afhængigStartværdi

uafhængigStartværdi og afhængigStartværdi kan være variable som f.eks. x0 og y0, der ikke har nogen lagrede værdier. Implicit differentiation kan hjælpe med at verificere implicitte løsninger.

deSolve

(
2.OrdensODE
andstartBeting1andstartBeting2, Var,
depVa) ⇒ en partikulær løsning

Returnerer en partikulær løsning, der opfylder 2. ordens ODE og har en angivet værdi for den afhængige variable og dens første afledede i et punkt.

For startBeting1, brug formen:

depVar (uafhængigStartværdi) = afhængigStartværdi

For startBeting2, brug formen:

afhVar (uafhængigStartværdi) = start
1.afledede

deSolve

(
2.OrdensODE
andbegrBeting1andbegrBeting2, Var,
depVar) ⇒ en partikulær løsning

$$\sin(y) = (y \cdot e^x + \cos(y)) \cdot y' \rightarrow ode$$

$$\sin(y) = (e^x \cdot y + \cos(y)) \cdot y'$$

$$deSolve(ode \text{ and } y(0)=0, x, y) \rightarrow soln$$

$$\frac{-(2 \cdot \sin(y) + y^2)}{2} = (e^x - 1) \cdot e^{-x} \cdot \sin(y)$$

$$soln|x=0 \text{ and } y=0 \quad \text{true}$$

$$ode|y'=\text{impDif}(soln, x, y) \quad \text{true}$$

$$DelVar \text{ ode, soln} \quad \text{Done}$$

$$deSolve(y''=y^{\frac{-1}{2}} \text{ and } y(0)=0 \text{ and } y'(0)=0, t, y)$$

$$\frac{\frac{3}{2} \cdot y^{\frac{4}{3}}}{3} = t$$

$$solve\left(\frac{\frac{3}{2} \cdot y^{\frac{4}{3}}}{3} = t, y\right)$$

$$y = \frac{3 \cdot 3^{\frac{3}{4}} \cdot 2^{\frac{3}{4}} \cdot t^{\frac{3}{4}}}{4} \text{ and } t \geq 0$$

$$deSolve(y''=x \text{ and } y(0)=1 \text{ and } y'(2)=3, x, y)$$

$$y = \frac{x^3}{6} + x + 1$$

$$deSolve(y''=2 \cdot y' \text{ and } y(3)=1 \text{ and } y'(4)=2, x, y)$$

$$y = e^{2 \cdot x - 8} - e^{-2 + 1}$$

Returnerer en partikulær løsning, der opfylder 2. ordens ODE og har angivne værdier i to forskellige punkter.

$$\text{deSolve}\left(w'' - \frac{2 \cdot w'}{x} + \left(9 + \frac{2}{x^2}\right) \cdot w = x \cdot e^x \text{ and } w\left(\frac{\pi}{6}\right) = 0 \text{ and } w\left(\frac{\pi}{3}\right) = 0, x, w\right)$$

$$w = \frac{x \cdot e^x}{(\ln(e))^2 + 9} + \frac{e^{\frac{\pi}{3}} \cdot x \cdot \cos(3 \cdot x)}{(\ln(e))^2 + 9} - \frac{e^{\frac{\pi}{6}} \cdot x \cdot \sin(3 \cdot x)}{(\ln(e))^2 + 9}$$

det()

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

Returnerer determinanten af *kvadratMatrix*.

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.

- 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:

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

$$\det\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad a \cdot d - b \cdot c$$

$$\det\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad -2$$

$$\det\left(\text{identity}(3) - x \cdot \begin{pmatrix} 1 & -2 & 3 \\ -2 & 4 & 1 \\ -6 & -2 & 7 \end{pmatrix}\right) \quad -(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$$

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

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

$$\det(\text{matI}, 1) \quad 1. \text{E}20$$

diag()

diag(List) ⇒ *matrix*

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

diag(rækkeMatrix) ⇒ *matrix*

diag(kolonneMatrix) ⇒ *matrix*

diag()

Katalog > 

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.

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

dim()

Katalog > 

dim(Liste)⇒heltal

dim({0,1,2}) 3

Returnerer dimensionen af liste.

dim(matrix)⇒liste

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

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

dim(Streng)⇒heltal

dim("Hello") 5

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

dim("Hello "&"there") 11

Disp

Katalog > 

Disp udtrykEllerStreng1 [, udtrykEllerStreng2] ...

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

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.

```
chars(240,243)
-----
240 ø
241 ñ
242 ò
243 ó
-----
Done
```

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

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.

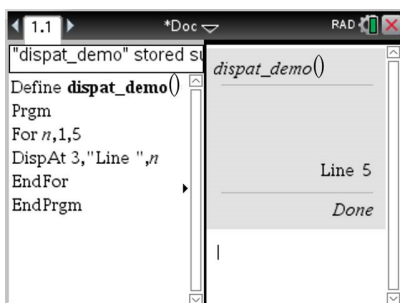
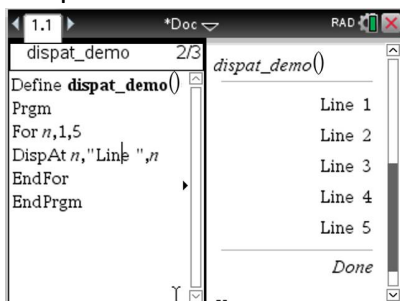
Denne kommando tillader dashboard-lignende output fra programmer, hvor værdien af et udtryk eller fra en sensor-aflæsning bliver opdateret på den samme linje.

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 - så længe linjerne ikke har 2D-matematiske udtryk. Det præcise antal linjer afhænger af indholdet af den viste information.

DispAt

Eksempel



Konkrete eksempler:

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

Fejlbetinger:

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)
Konverteringsoperator: DispAt 2_ft @> _m, "Hallo verden"	CAS: Datatypefejl er fundet ugyldig (hvis tilbagekald er defineret) Numerisk: Konverteringen vil blive evalueret, og hvis resultatet er et gyldigt argument, vil DispAt printe strengen på resultatlinjen.

►DMSKatalog >

Udtr ►DMS

I vinkeltilstanden Grader:

List ►DMS

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

Matrix ►DMS

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 235.

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.

domain()

domain(udtryk1, Var)⇒udtryk

Beregner definitionsområdet for *udtryk1* med hensyn til *Var*.

domain() kan bruges til at undersøge definitionsområdet for funktioner. Den er begrænset til de reelle tal og til et endeligt antal intervaller.

Denne funktionalitet har begrænsninger pga. mangler i computeralgebraens reduktions- og ligningsløsningsalgoritmer.

Visse funktioner kan ikke bruges som argumenter for **domain()**, hvad enten de forekommer eksplicit eller inden for brugerdefinerede variable og funktioner. I det følgende eksempel kan udtrykket ikke reduceres, fordi $f()$ ikke er en tilladt funktion.

$$\text{domain}\left(\left[\begin{array}{c} x \\ \frac{1}{t} \\ 1 \end{array} dt, x\right]\right) \rightarrow \text{domain}\left(\left[\begin{array}{c} x \\ \frac{1}{t} \\ 1 \end{array} dt, x\right]\right)$$

$$\begin{array}{ll} \text{domain}\left(\frac{1}{x+y}, y\right) & -\infty < y < -x \text{ or } -x < y < \infty \\ \text{domain}\left(\frac{x+1}{x^2+2 \cdot x}, x\right) & x \neq -2 \text{ and } x \neq 0 \\ \text{domain}\left(\left(\sqrt{x}\right)^2, x\right) & 0 \leq x < \infty \\ \text{domain}\left(\frac{1}{x+y}, y\right) & -\infty < y < -x \text{ or } -x < y < \infty \end{array}$$

dominantTerm(*Udtr1*, *Var* [, *Punkt*])
 $\Rightarrow$ udtryk

dominantTerm(*Udtr1*, *Var* [, *Punkt*]) |
Var>*Punkt* $\Leftrightarrow$ udtryk

dominantTerm(*Udtr1*, *Var* [, *Punkt*])
Var<*Punkt* $\Rightarrow$ udtryk

Returnerer det dominerende led i potensrækkeudviklingen af *Udtr1* udviklet omkring *Punkt*. Det dominerende led er det led, hvis der vokser hurtigst nær *Var* = *Punkt*. Den resulterende potens af (*Var* - *Punkt*) kan have en negativ og/eller brøkeksponent. Koefficienten af denne potens kan indeholde logaritmer af (*Var* - *Punkt*) og andre funktioner af *Var*, der domineres af alle potenser af (*Var* - *Punkt*) med samme eksponentfortegn.

Punkt er som standard 0. *Punkt* kan være ∞ eller $-\infty$, i hvilke tilfælde det dominerende led vil være det led, der har den største eksponent af *Var* frem for den mindste eksponent for *Var*.

dominantTerm(...) returnerer "dominantTerm(...)" hvis den ikke kan bestemme en sådan repræsentation, for eksempel for væsentlige singulariteter som $\sin(1/z)$ ved $z=0$, $e^{-1/z}$ ved $z=0$, eller e^z ved $z = \infty$ eller $-\infty$.

Hvis rækken eller en af dens afledte afbrydes af et spring ved *Punkt*, indeholder resultatet sandsynligvis deludtryk på formen sign(...) eller abs(...) for en reel udviklingsvariabel eller $(-1)^{\text{floor}(\dots \text{angle}(\dots))}$ for en kompleks udviklingsvariabel, der slutter med "...". Hvis du kun vil bruge det dominerende led til værdier på den ene af *Punkt*, skal du til **dominantTerm(...)** tilføje den korrekte af funktionerne "*| Var* > *Punkt*", "*| Var* < *Punkt*", "*| Var* $\geq$ *Punkt*", eller "*Var* $\leq$ *Punkt*" for at opnå et enklere resultat.

$\text{dominantTerm}(\tan(\sin(x)) - \sin(\tan(x)), x)$	$\frac{x^7}{30}$
$\text{dominantTerm}\left(\frac{1 - \cos(x-1)}{(x-1)^3}, x, 1\right)$	$\frac{1}{2 \cdot (x-1)}$
$\text{dominantTerm}\left(x^{-2} \cdot \tan\left(\frac{1}{3}\right), x\right)$	$\frac{1}{5} x^3$
$\text{dominantTerm}(\ln(x^x - 1) \cdot x^{-2}, x)$	$\frac{\ln(x \cdot \ln(x))}{x^2}$

$\text{dominantTerm}\left(e^{\frac{-1}{z}}, z\right)$	$\text{dominantTerm}\left(e^{\frac{-1}{z}}, z, 0\right)$
$\text{dominantTerm}\left(\left(1 + \frac{1}{n}\right)^n, n, \infty\right)$	e
$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 0\right)$	$\frac{\pi \cdot \text{sign}(x)}{2}$
$\text{dominantTerm}\left(\tan^{-1}\left(\frac{1}{x}\right), x, x > 0\right)$	$\frac{\pi}{2}$

dominantTerm() distribuerer over 1. argument lister og matricer.

dominantTerm() er nyttig, når du vil have det enklost mulige udtryk, der nærmer sig asymptotisk til et andet udtryk som $Var \rightarrow Punkt$. **dominantTerm()** er også nyttig, når det ikke er klart, hvad graden af det første ikke-nul-led i en række vil være, og du ikke ønsker en række interaktive iterative gæt eller med en programløkke.

Bemærk: Se også **series()**, side 165.

dotP()

dotP(Liste1, Liste2)⇒udtryk

Returnerer "prik" produktet af to lister.

$\text{dotP}(\{a,b,c\},\{d,e,f\})$	$a \cdot d + b \cdot e + c \cdot f$
$\text{dotP}(\{1,2\},\{5,6\})$	17

dotP(Vektor1, Vektor2)⇒udtryk

Returner "prik" produktet af to vektorer.

$\text{dotP}([a \ b \ c],[d \ e \ f])$	$a \cdot d + b \cdot e + c \cdot f$
$\text{dotP}([1 \ 2 \ 3],[4 \ 5 \ 6])$	32

Begge skal være rækkevektorer, eller begge skal være kolonnevektorer.

E

e^()

e^(Udtr1)⇒udtryk

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

e^1	e
$e^{1.}$	2.71828
e^{3^2}	e^9

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

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

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

e^(Liste1)⇒liste

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

e^()

e^x-tast

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

e^(kvadratMatrix1)⇒kvadratMatrix

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.

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

eff()

Katalog >

eff(nominelRente,CpY)⇒værdi

eff(5.75,12)

5.90398

Finansfunktion, 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 127.

eigVc()

Katalog >

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 skales af enhver konstantfaktor. Egenvektorerne er normaliseret, dvs. at hvis $V = [x_1, x_2, \dots, x_n]$, så:

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

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

eigVc(m1)

-0.800906	0.767947	(
0.484029	0.573804+0.052258 <i>i</i>	0.5738
0.352512	0.262687+0.096286 <i>i</i>	0.2626

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.

eigVl(*kvadratMatrix*)⇒*liste*

Returnerer en liste med egenverdier 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 egenverdierne beregnes fra øvre Hessenberg-matricen.

I rektangulær kompleks formattilstand:

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

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

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

```
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 75.
EndFunc	Se Func, side 79.
EndIf	Se If, side 90.
EndLoop	Se Loop, side 113.
EndPrgm	Se Prgm, side 141.
EndTry	Se Try, side 196.
EndWhile	Se While, side 206.

euler ()	Katalog > 										
euler (<i>Expr</i> , <i>Var</i> , <i>depVar</i> , { <i>Var0</i> , <i>VarMax</i> }, <i>depVar0</i> , <i>VarStep</i> [, <i>eulerStep</i>]) $\Rightarrow$ <i>matrix</i>	Differentialligning: $y' = 0.001 \cdot y \cdot (100 - y)$ og $y(0) = 10$										
euler (<i>SystemOfExpr</i> , <i>Var</i> , <i>ListOfDepVars</i> , { <i>Var0</i> , <i>VarMax</i> }, <i>ListOfDepVars0</i> , <i>VarStep</i> [, <i>eulerStep</i>]) $\Rightarrow$ <i>matrix</i>	euler(0.001·y·(100−y),t,y,{0,100},10,1) <table><tr><td>0.</td><td>1.</td><td>2.</td><td>3.</td><td>4.</td></tr><tr><td>10.</td><td>10.9</td><td>11.8712</td><td>12.9174</td><td>14.042</td></tr></table>	0.	1.	2.	3.	4.	10.	10.9	11.8712	12.9174	14.042
0.	1.	2.	3.	4.							
10.	10.9	11.8712	12.9174	14.042							
euler (<i>ListOfExpr</i> , <i>Var</i> , <i>ListOfDepVars</i> , { <i>Var0</i> , <i>VarMax</i> }, <i>ListOfDepVars0</i> , <i>VarStep</i> [, <i>eulerStep</i>]) $\Rightarrow$ <i>matrix</i>	Du kan se hele resultatet ved at trykke på  og derefter bruge  og  til at bevæge markøren.										
Anvender Euler-metoden til at løse systemet											

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

med $\text{depVar}(\text{Var0}) = \text{depVar0}$ i intervallet $[\text{Var0}, \text{VarMax}]$. Returnerer en matrix, hvor første række definerer Var -outputværdierne, og anden række definerer værdien af første løsningskomponent ved de tilsvarende Var -værdier, osv.

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

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

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

Var er den uafhængige variable.

ListOfDepVars er en liste med afhængige variable.

$\{\text{Var0}, \text{VarMax}\}$ er en liste med to elementer, der informerer funktionen om, at integrere fra Var0 til VarMax .

ListOfDepVars0 er en liste med startværdier for afhængige variable.

VarStep er et tal forskelligt fra nul, således at $\text{sign}(\text{VarStep}) = \text{sign}(\text{VarMax} - \text{Var0})$ og løsninger returneres i $\text{Var0} + i \cdot \text{VarStep}$ for alle $i=0,1,2,\dots$ således at $\text{Var0} + i \cdot \text{VarStep}$ er i $[\text{var0}, \text{VarMax}]$ (der vil muligvis ikke være en løsningsværdi ved VarMax).

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 $\text{VarStep} / \text{eulerStep}$.

Sammenlign resultatet ovenfor med den eksakte CAS løsning opnået ved brug af $\text{deSolve}()$ og $\text{seqGen}()$:

$$\text{deSolve}\{y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y\}$$

$$y = \frac{100 \cdot \{1.10517\}^t}{\{1.10517\}^t + 9.}$$

$$\text{seqGen}\left(\frac{100 \cdot \{1.10517\}^t}{\{1.10517\}^t + 9.}, t, y, \{0, 100\}\right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189\}$$

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 $y2(0) = 5$

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

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

eval(*Expr*) $\Rightarrow$ *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.

fadein()	Done
-----------------	------

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 80), **GetStr** (side 87), og **Send** (side 162).

<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"

exact()	Katalog > 
exact (<i>Udtr1</i> [, <i>Tolerance</i>])⇒ <i>udtryk</i>	$\text{exact}(0.25) \quad \frac{1}{4}$
exact (<i>Liste1</i> [, <i>Tolerance</i>])⇒ <i>liste</i>	$\text{exact}(0.333333) \quad \frac{333333}{1000000}$
exact (<i>Matrix1</i> [, <i>Tolerance</i>])⇒ <i>matrix</i>	$\text{exact}(0.333333, 0.001) \quad \frac{1}{3}$
Anvender Eksakt-tilstandens aritmetik til at returnere det rationale talækivalent til argumentet, hvis det er muligt.	$\text{exact}(3.5 \cdot x + y) \quad \frac{7 \cdot x}{2} + y$
<i>Tolerance</i> angiver tolerancen for konverteringen; standardindstillingen er 0 (nul).	$\text{exact}(\{0.2, 0.33, 4.125\}) \quad \left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$

Exit	Katalog > 
Exit	Funktionsliste:
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.	<pre> Define g()$\equiv$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 </pre>
	$g()$ 21

exp	Katalog > 
Udtr ▶ exp	$\frac{d}{dx} (e^x + e^{-x}) \quad 2 \cdot \sinh(x)$ $2 \cdot \sinh(x)$ ▶ exp $e^x - e^{-x}$
Repræsenterer <i>Udtr</i> i termer af den naturlige eksponentialfunktion <i>e</i>. Det er en konverteringsoperator til visning. Denne operator kan kun anvendes ved slutningen af indtastningslinjen. Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>exp.	

exp()**e^x-tast****exp(*UdtrI*)**⇒*udtryk*Returnerer **e** opløftet til potensen *UdtrI*.Returnerer **e** opløftet til potensen *VærdiI*.**Bemærk:** Se også **e** eksponentskabelon, side 2.

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

exp(*ListeI*)⇒*liste*Returnerer **e** opløftet til potensen af hvert element i *ListeI*.**exp(*kvadratMatrixI*)**⇒*kvadratMatrix*

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

KvadratMatrixI skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

e^1	e
$e^{1.}$	2.71828
e^{3^2}	e^9

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

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

exp►liste()**Katalog > ****exp►list(*Udtr,Var*)**⇒*liste*

Undersøger *Udtr* for ligninger, der er adskilt af ordet "or", og returnerer en liste med ligningernes højresider på formen *Var*=*udtryk*. Dette er en nem metode til at udtrække løsningsværdier, der er indlejret i resultaterne af funktionerne **solve()**, **cSolve()**, **fMin()**, og **fMax()**.

Bemærk: **exp►list()** er ikke nødvendig med funktionerne **zeros** og **cZeros()**, da de direkte returnerer en liste med løsningsværdier.

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

$\text{solve}(x^2-x-2=0,x)$	$x=-1 \text{ or } x=2$
$\text{exp►list}(\text{solve}(x^2-x-2=0,x),x)$	$\{-1,2\}$

expand()

Katalog > 

expand(*Udtr1* [, *Var*]) $\Rightarrow$ *udtryk*

expand(*Liste1* [, *Var*]) $\Rightarrow$ *liste*

expand(*Matrix1* [, *Var*]) $\Rightarrow$ *matrix*

expand(*Udtr1*) returnerer *Udtr1* på ledform med hensyn til alle variable. Ledformen er en polynomiumsudvikling for polynomier og en udvikling i partialbrøker for polynomiumsbrøker.

Målet for **expand()** er at transformere *Udtr1* til en sum og/eller differens for simple led. Som modsætning er målet for **factor()** at transformere *Udtr1* til et produkt og/eller kvotient af simple faktorer.

expand(*Udtr1*, *Var*) returnerer *Udtr1* på ledform med hensyn til *Var*. Ens potenser af *Var* samles. Leddene og deres faktorer sorteres med *Var* som hovedvariabel. Der kan opstå en vis utilsigtet faktoropløsning eller udvikling af de reducerede koefficienter. Sammenlignet med at udelade *Var* sparer dette ofte tid, hukommelse og skærmplass, samtidig med, at det gør udtrykket mere læseligt.

Selv når der kun er en variabel, kan anvendelsen af *Var* gøre faktoropløsningen af nævneren til delvis udvikling af brøker mere fuldstændig.

Tip: For rationale udtryk er **propFrac()** et hurtigere men mindre vidtgående alternativ til **expand()**.

Bemærk: Se også **comDenom()** vedrørende en tæller på ledform over en nævner på ledform.

$$\begin{aligned} &\text{expand}\left((x+y+1)^2\right) \\ &\quad x^2+2\cdot x\cdot y+2\cdot x\cdot y^2+2\cdot y+1 \\ &\text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}\right) \\ &\quad \frac{1}{x-1}-\frac{1}{x}-\frac{1}{y-1}-\frac{1}{y} \end{aligned}$$

$$\begin{aligned} &\text{expand}\left((x+y+1)^2, y\right) \quad y^2+2\cdot y\cdot (x+1)+(x+1)^2 \\ &\text{expand}\left((x+y+1)^2, x\right) \quad x^2+2\cdot x\cdot (y+1)+(y+1)^2 \\ &\text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2+x\cdot y}, y\right) \\ &\quad \frac{1}{y-1}-\frac{1}{y}+\frac{1}{x\cdot (x-1)} \\ &\text{expand}(\text{Ans}, x) \quad \frac{1}{x-1}-\frac{1}{x}+\frac{1}{y\cdot (y-1)} \end{aligned}$$

$$\begin{aligned} &\text{expand}\left(\frac{x^3+x^2-2}{x^2-2}\right) \quad \frac{2\cdot x}{x^2-2}+x+1 \\ &\text{expand}(\text{Ans}, x) \quad \frac{1}{x-\sqrt{2}}+\frac{1}{x+\sqrt{2}}+x+1 \end{aligned}$$

expand()

[Katalog >](#) 

expand(Udtr1, [Var]) distribuerer også logaritmer og brøkpotenser, uanset *Var*. For øget distribution af logaritmer og brøkpotenser kan grænser for uligheder være nødvendige for at garantere, at visse faktorer er ikke-negative.

expand(Udtr1, [Var]) distribuerer også absolutte værdier, **sign()** og eksponentialfunktioner uanset *Var*.

Bemærk: Se også **tExpand()** vedrørende trigonometriske additionsformler og formler for multipel vinkel.

$\ln(2 \cdot x \cdot y) + \sqrt{2 \cdot x \cdot y}$	$\ln(2 \cdot x \cdot y) + \sqrt{2 \cdot x \cdot y}$
$\text{expand}(\text{Ans})$	$\ln(x \cdot y) + \sqrt{2 \cdot \sqrt{x \cdot y} + \ln(2)}$
$\text{expand}(\text{Ans}) y \geq 0$	
	$\ln(x) + \sqrt{2 \cdot \sqrt{x} \cdot \sqrt{y} + \ln(y)} + \ln(2)$
$\text{sign}(x \cdot y) + x \cdot y + e^{2 \cdot x + y}$	
	$e^{2 \cdot x + y + \text{sign}(x \cdot y) + x \cdot y }$
$\text{expand}(\text{Ans})$	
	$\text{sign}(x) \cdot \text{sign}(y) + x \cdot y + (e^x)^2 \cdot e^y$

expr()

[Katalog >](#) 

expr(Streng)⇒udtryk

Returnerer den tegnstreng, der er indeholdt i *Streng* som et udtryk og og eksekverer den straks.

$\text{expr}("1+2+x^2+x")$	x^2+x+3
$\text{expr}("\text{expand}((1+x)^2)")$	$x^2+2 \cdot x+1$
"Define cube(x)=x^3" → funcstr	
"Define cube(x)=x^3"	
$\text{expr}(\text{funcstr})$	Done
$\text{cube}(2)$	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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder for X og Y data.

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

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot (b)^x$
stat.a, stat.b	Regressionskoefficienter
stat.r ²	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 X -liste, der faktisk bruges i regressionen ud fra begrænsninger af <i>Hyppighed</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 af <i>Hyppighed</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(*Udtr1* [, *Var*]) $\Rightarrow$ *udtryk*

factor(*Liste1* [, *Var*]) $\Rightarrow$ *liste*

factor(*Matrix1* [, *Var*]) $\Rightarrow$ *matrix*

factor(*Udtr1*) returnerer *Udtr1* opløst i faktorer med hensyn til alle dens variable over en fællesnævner.

$\text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a)$	$a \cdot (a-1) \cdot (a+1) \cdot (x-1) \cdot (x+1)$
$\text{factor}(x^2+1)$	x^2+1
$\text{factor}(x^2-4)$	$(x-2) \cdot (x+2)$
$\text{factor}(x^2-3)$	x^2-3
$\text{factor}(x^2-a)$	x^2-a

Udtr1 opløses mest muligt mod lineære rationale faktorer uden at indføre nye ikke-reelle deludtryk. Denne mulighed er velegnet, hvis du ønsker opløsning i faktorer med hensyn til mere end en variabel.

`factor(Udtr1,Var)` returnerer *Udtr1* opløst i faktorer med hensyn til variabelen *Var*.

Udtr1 opløses mest muligt i faktorer, der er lineære i *Var*, også hvis den indfører irrationale konstanter eller deludtryk, der er irrationale i andre variable.

Faktorerne og deres led sorteres med *Var* som hovedvariabel. Ens potenser af *Var* samles i hver faktor. Medtag *Var*, hvis opløsning i faktorer kun er nødvendig med hensyn til den pågældende variabel, og du er villig til at acceptere irrationale udtryk i alle andre variable for at øge opløsningen i faktorer med hensyn til *Var*. Der kan forekomme en uforudset faktoropløsning med hensyn til andre variable.

For Auto-indstillingen af **Auto eller tilnærmet** tilstanden muliggør medtagelsen af *Var* også en approksimering med koefficienter med flydende decimaler, hvor irrationale koefficienter ikke kan udtrykkes med de indbyggede funktioner. Også når der kun er en variabel, kan en medtagelse af *Var* give en mere komplet opløsning i faktorer.

Bemærk: Se også `comDenom()` for en hurtig metode til at opnå partiel faktoropløsning, når `factor()` ikke er hurtig nok, eller den bruger hele hukommelsen.

Bemærk: Se også `cFactor()` om faktoropløsning med komplekse koefficienter for at opnå lineære faktorer.

$$\begin{array}{l} \text{factor}(a^3 \cdot x^2 - a \cdot x^2 - a^3 + a, x) \\ \quad a \cdot (a^2 - 1) \cdot (x - 1) \cdot (x + 1) \\ \text{factor}(x^2 - 3, x) \quad (x + \sqrt{3}) \cdot (x - \sqrt{3}) \\ \text{factor}(x^2 - a, x) \quad (x + \sqrt{a}) \cdot (x - \sqrt{a}) \end{array}$$

$$\begin{array}{l} \text{factor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3) \\ \quad x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3 \\ \text{factor}(x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3, x) \\ \quad (x - 0.964673) \cdot (x + 0.611649) \cdot (x + 2.12543) \cdot (x^2 \end{array}$$

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  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

(
nedreGrænse
,øvreGrænse,fgTæller,fgNævner) ⇒ *tal* hvis
nedreGrænse og *øvreGrænse* er tal, liste
hvis *nedreGrænse* og *øvreGrænse* er lister

FCdf

(
nedreGrænse
,øvreGrænse,fgTæller,fgNævner) ⇒ *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 $P(X \leq \text{øvreGrænse})$ skal du sætte *nedreGrænse* = 0.

Fill

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

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

matrixVar skal eksistere i forvejen.

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

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

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

Listevar skal eksistere i forvejen.

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

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 181.)

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 ≥ 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 243

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

floor()

floor(Udtr1)⇒heltal

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

Returnerer det største heltal, der er $\leq$ 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()**.

fMax()

fMax(Udtr, Var)⇒Boolsk udtryk

$$\text{fMax}\left(1-(x-a)^2-(x-b)^2, x\right) \quad x=\frac{a+b}{2}$$

fMax(Udtr, Var, nedreGrænse)

$$\text{fMax}\left(5 \cdot x^3 - x - 2, x\right) \quad x=\infty$$

fMax(Udtr, Var, nedreGrænse, øvreGrænse)

fMax(Udtr, Var) | nedreGrænse ≤ Var ≤ øvreGrænse

Returnerer et Boolsk udtryk, der angiver mulige værdier for *Var*, der maksimerer *udtryk* eller finder dets mindste øvre grænse.

Med ("|") betingelses-operatoren kan du indskrænke løsningsintervallet og/eller angive andre begrænsninger.

I Tilnærmet-indstillingen af **Auto eller tilnærmet**-tilstanden søger **fMax()** iterativt efter et tilnærmet lokalt maksimum. Dette er ofte hurtigere, især hvis du anvender "|" -operatoren til at begrænse søgningen til et relativt lille interval, der indeholder nøjagtigt et lokalt maksimum.

Bemærk: Se også **fMin()** og **max()**.

$$\text{fMax}(0.5 \cdot x^3 - x - 2, x) | x \leq 1 \quad x = -0.816497$$

fMin(*Udtr*, *Var*) $\Rightarrow$ Boolsk udtryk

fMin(*Udtr*, *Var*, *nedreGrænse*)

fMin(*Udtr*, *Var*, *nedreGrænse*, *øvreGrænse*)

fMin(*Udtr*, *Var*) | *nedreGrænse* $\leq$ *Var* $\leq$ *øvreGrænse*

Returnerer et Boolsk udtryk, der angiver mulige værdier for *var*, der minimerer *Udtr* eller finder dets største nedre grænse.

Med ("|") betingelses-operatoren kan du indskrænke løsningsintervallet og/eller angive andre begrænsninger.

Til Approx-indstillingen af **Auto eller tilnærmet** tilstanden søger **fMin()** iterativt for et enkelt tilnærmet lokalt minimum. Dette er ofte hurtigere, især hvis du anvender "|" -operatoren til at begrænse søgningen til et relativt lille interval, der indeholder nøjagtigt et lokalt minimum.

Bemærk: Se også **fMax()** og **min()**.

$$\text{fMin}(1 - (x-a)^2 - (x-b)^2, x) \quad x = -\infty \text{ or } x = \infty$$

$$\text{fMin}(0.5 \cdot x^3 - x - 2, x) | x \geq 1 \quad x = 1.$$

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 g()
Local tempsum,step,i
0 → tempsum
1 → step
For i,1,100,step
tempsum + i → tempsum
EndFor
EndFunc

Done

g()

5050

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

Returnerer *Udtr* som en tegnstreng baseret på formatskabelonen.

Udtr skal kunne omregnes til et tal.

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.

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"

$G[n][c]$: 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 grundtalpunktet.

fPart()

fPart(*UdtrI*) $\Rightarrow$ *udtryk*

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

fPart(*ListeI*) $\Rightarrow$ *liste*

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

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

Returnerer decimaldelen af argumentet.

For en liste eller matrix returneres decimaldelen af elementerne.

Argumentet kan være et reelt eller komplekst tal.

FPdf()

FPdf(*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(*ListeI*,*frekvHeltalListe*) $\Rightarrow$ *liste*

$freqTable►list(\{1, 2, 3, 4\}, \{1, 4, 3, 1\})$	$\{1, 2, 2, 2, 3, 3, 3, 4\}$
$freqTable►list(\{1, 2, 3, 4\}, \{1, 4, 0, 1\})$	$\{1, 2, 2, 2, 4\}$

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- & Statistikapplikationer.

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 243.

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*.

Hvis *binsListe* er {b(1), b(2), ..., b(n)}, er de specificerede områder { $? \leq b(1)$, $b(1) < ? \leq b(2)$, ..., $b(n-1) < ? \leq b(n)$, $b(n) > ?$ }. 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 {countIf(liste, $? \leq b(1)$), countIf(liste, $b(1) < ? \leq b(2)$), ..., countIf(liste, $b(n-1) < ? \leq b(n)$), countIf(liste, $b(n) > ?$)}.

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

Forklaring af resultatet:

2 elementer fra *Dataliste* er ≤ 2.5

4 elementer fra *Dataliste* er > 2.5 og ≤ 4.5

3 elementer fra *Dataliste* er > 4.5

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

Elementer i *Liste1*, der ikke kan “placeres i en størrelse” ignoreres. Tomme (ugyldige) elementer ignoreres også. Yderligere oplysninger om tomme elementer findes på side 243.

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

Bemærk: Se også `countIf()`, side 36.

FTest_2Samp

FTest_2Samp *Liste1*,*Liste2*[,*Hyppighed1* [,*Hyppighed2*[,*Hypot*]]]

FTest_2Samp *Liste1*,*Liste2*[,*Hyppighed1* [,*Hyppighed2*[,*Hypot*]]]

(Data)listeinput)

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

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

(Sammenfatning, stat input)

Udfører en F test med to målinger. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 181.)

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 243.

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>

Output-variabel	Beskrivelse
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

Katalog >

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.

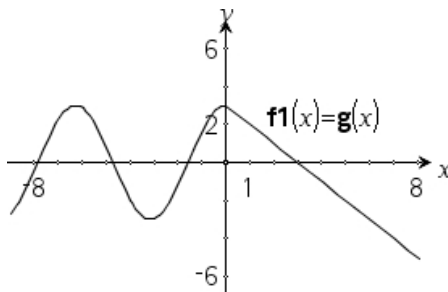
Bemærk indtastning af eksemplet:

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 $3 \cdot \cos(x)$	
Else	
Return $3-x$	
EndIf	
EndFunc	

Tegnet resultat af grafen $g(x)$



G

gcd()

Katalog >

gcd(*Værdi1*, *Værdi2*)⇒*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 **tilnærm**-tilstand er **gcd** af flydende decimalbrøker 1.0.

gcd(*Liste1*, *Liste2*)⇒*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*

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

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

geomCdf()

geomCdf(*p*, *nedreGrænse*, *øvreGrænse*)

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

geomCdf(*p*, *øvreGrænse*) for $P(1 \leq X \leq \text{øvreGrænse}) \Rightarrow$ *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 $P(X \leq \text{øvreGrænse})$, sæt *nedreGrænse*=1.

geomPdf()

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

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

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

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

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*

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 87 og **Send**, side 162.

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

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

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

getDenom()

Katalog >

getDenom(Udtr1) ⇒ udtryk

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

$\text{getDenom}\left(\frac{x+2}{y-3}\right)$	$y-3$
$\text{getDenom}\left(\frac{2}{7}\right)$	7
$\text{getDenom}\left(\frac{1}{x} + \frac{y^2+y}{y^2}\right)$	$x \cdot y$

getKey()

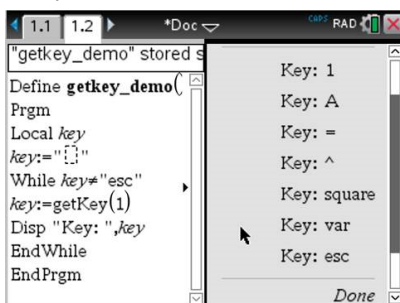
Katalog >

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:**

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"

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
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^2	n/a	"kvadrat"
/ (divisionstast)	/	"/"
* (gangetast)	*	"*"
e^x	n/a	"eksp"
10^x	n/a	"10potens"
+	+	"+"
-	-	"_"
(	(	"("
)	)	")"
.	.	"."
(-)	n/a	"-" (negativ fortegn)
Indtast	Indtast	"indtast"
ee	n/a	"E" (eksponentiel notation E)

Håndholdte enheder/emulatortast	Skrivebord	Returværdi
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

Hændelse	Enhed	Skrivebord - TI-Nspire™ Student Software
		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 variablen *Var*.

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

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

Se **Lock**, side 109, og **unLock**, side 203.

<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)	{ 1,7,2,1,3,1,4,1,5,1,6,1,7,1,8,1 }
getMode(1)	7
getMode(8)	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 166.

Tilstands- navn	Tilstands- heltal	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 kompleks	4	1=Reel, 2=Rektangulær, 3=Polær
Auto eller tilnærmet	5	1=Auto, 2=Tilnærmet, 3=Eksakt
Vektorformat	6	1=Rektangulær, 2=Cylindrisk, 3=Sfærisk
Talsystem	7	1=Decimal, 2=Hex, 3=Binær
Enhedssystem	8	1=SI, 2=Eng/US

getNum()

getNum(*Udtr1*)⇒*udtryk*

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

$\text{getNum}\left(\frac{x+2}{y-3}\right)$	$x+2$
$\text{getNum}\left(\frac{2}{7}\right)$	2
$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$	$x+y$

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 80 og **Send**, side 162.

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"

getVarInfo(*Biblioteksnavnestreng*) 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.

getVarInfo()	"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>												
getVarInfo()	<table><tr><td>x</td><td>"NUM"</td><td>"{"}</td><td>1</td></tr><tr><td>y</td><td>"LIST"</td><td>"LibPriv"</td><td>0</td></tr><tr><td>z</td><td>"FUNC"</td><td>"LibPub"</td><td>0</td></tr></table>	x	"NUM"	"{"}	1	y	"LIST"	"LibPriv"	0	z	"FUNC"	"LibPub"	0
x	"NUM"	"{"}	1										
y	"LIST"	"LibPriv"	0										
z	"FUNC"	"LibPub"	0										
getVarInfo(<i>tmp3</i>)	"Error: Argument must be a string"												
getVarInfo("tmp3")	<table><tr><td><i>volcyl2</i></td><td>"NONE"</td><td>"LibPub"</td><td>0</td></tr></table>	<i>volcyl2</i>	"NONE"	"LibPub"	0								
<i>volcyl2</i>	"NONE"	"LibPub"	0										

getVarInfo()

Katalog > 

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.

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

Goto

Katalog > 

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

►Grad

Katalog > 

Udtr1 ► **Grad** ⇒ *udtryk*

Konverterer *Udtr1* til vinkelmål i nygrader.

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

I vinkeltilstanden Grader:

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

I vinkeltilstanden Radian:

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

identitet()Katalog > **identitet(heltal)** $\Rightarrow$ *matrix*

Returnerer identitetsmatrixen med en dimension af *heltal*.

Heltal skal være et positivt heltal.

identity(4)

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

IfKatalog > 

If *BooleanExpr*
Statement

Define $g(x)$ =Func

Done

If *BooleanExpr* **Then**
Block

If $x < 0$ ThenReturn x^2

EndIf

EndFunc

EndIf $g(-2)$

4

Hvis *BooleanExpr* evalueres som sand, eksekveres enkeltsætningen *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

Define $g(x)$ =Func

Done

ElseIf $x < 0$ ThenReturn $-x$

Else

Return x

EndIf

EndFunc

*Block2***EndIf** $g(12)$

12

 $g(-12)$

12

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

```

Done

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

ifFn()

Katalog >

ifFn(*BooleanExpr*, *Value_If_true* [, *Value_If_false* [, *Value_If_unknown*]]) $\Rightarrow$ *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*.
- 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*.

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

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

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.

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

Value_If_True er en enkelt værdi og svarer til enhver valgt position.

ifFn()

Katalog > 

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.

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

Value_If_False er ikke specificeret. Undef anvendes.

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

Ét element valgt fra *Value_If_True*. Ét element valgt fra *Value_If_unknown*.

imag()

Katalog > 

$\text{imag}(\text{Expr } I) \Rightarrow \text{udtryk}$

Returnerer imaginærdelen af argumentet.

Bemærk: Alle udefinerede variable behandles som reelle variable. Se også reel (), page 149

$\text{imag}(\text{List } l) \Rightarrow \text{liste}$

Returnerer en liste med imaginærdelen af elementerne.

$\text{imag}(\text{Matrix } l) \Rightarrow \text{matrix}$

Returnerer en matrix med imaginærdelene af elementerne.

$\text{imag}(1+2 \cdot i)$	2
$\text{imag}(z)$	0
$\text{imag}(x+i \cdot y)$	y

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

$\text{imag}\left(\begin{bmatrix} a & b \\ i \cdot c & i \cdot d \end{bmatrix}\right)$	$\begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$
--	--

impDif()

Katalog > 

$\text{impDif}(\text{Equation}, \text{Var}, \text{dependVar}[,\text{Ord}]) \Rightarrow \text{udtryk}$

hvor ordenen *Ord* som standard er 1.

Beregner den implicitte differentialkvotient til ligninger, hvor en variabel defineres implicit ud fra en anden.

$\text{impDif}(x^2+y^2=100,x,y)$	$\frac{-x}{y}$
----------------------------------	----------------

Henvisning

Se #(), side 233.

inString()

Katalog > 

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

inString("Hello there", "the")	7
inString("ABCEFG", "D")	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(*Expr*) ⇒ *heltal*

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

int(*List1*) ⇒ *liste*

int(*Matrix1*) ⇒ *matrix*

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*

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

Katalog >

interpoler(*xValue*, *xList*, *yList*, *yPrimeList*) $\Rightarrow$ *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 den interpoleret værdi for **f** (*xValue*); i modsat fald returnerer den **undef**.

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

xValue kan være en ikke-defineret variabel, et tal eller en liste af tal.

Differentialligning:

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

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

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

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

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

inv $\chi^2()$

Katalog >

inv χ^2 (*Areal*, *df*)

invChi2(*Areal*, *df*)

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

invF()

Katalog >

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*) $\Rightarrow$ 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*) $\Rightarrow$ 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*[, μ],[σ]])

Beregner den inverse kumulerede normalfordelingsfunktion for et givet *Areal* under normalfordelingskurven angivet ved μ og σ .

invt()Katalog > **invt**(*Areal*,*df*)

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

iPart()Katalog > **iPart**(*Number*) $\Rightarrow$ *heltal***iPart**(*List1*) $\Rightarrow$ *liste***iPart**(*Matrix1*) $\Rightarrow$ *matrix*

Returnerer heltalsdelen af argumentet.

Returnerer heltalsdelen af hvert element for lister og matrixer.

Argumentet kan være et reelt eller komplekst tal.

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

irr()Katalog > **irr**(*CF0*,*CFList* [*CFFreq*]) $\Rightarrow$ *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*.

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

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.

Bemærk: Se også *mirr()*, side 118.

isPrime()

isPrime(Number) ⇒ *boolsk konstantudtryk*

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

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

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

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

<i>isPrime(5)</i>	true
<i>isPrime(6)</i>	false

Funktion til søgning af det næste primtal efter det angivne tal:

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

isVoid()

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 243.

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

Lbl	Katalog > 
Lbl <i>etiketNavn</i> Definerer en etiket med navnet <i>etiketNavn</i> i en funktion. Du kan anvende en Goto <i>etiketNavn</i>-kommando til at videregive kontrollen til kommandoen lige efter etiketten. <i>EtiketNavn</i> skal opfylde de samme navngivningskrav som et variabelnavn. Bemærk indtastning af eksemplet: For instruktioner til at indtaste programmer over flere linjer og definering af funktioner se Beregninger-afsnittet i din produktvejledning.	Define $g()$=Func Local <i>temp,i</i> $0 \rightarrow temp$ $1 \rightarrow i$ Lbl <i>top</i> $temp+i \rightarrow temp$ If $i<10$ Then $i+1 \rightarrow i$ Goto <i>top</i> EndIf Return <i>temp</i> EndFunc Done $g()$ 55

lcm()	Katalog > 
lcm(<i>Værdi1, Værdi2</i>)⇒<i>udtryk</i> lcm(<i>Liste1, Liste2</i>)⇒<i>liste</i> lcm(<i>Matrix1, Matrix2</i>)⇒<i>matrix</i> 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.	$lcm(6,9)$ 18 $lcm\left(\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}\right)$ $\left\{\frac{2}{3}, 14, 80\right\}$

left()	Katalog > 
left(<i>kildeStreng[, Tal]</i>)⇒<i>streng</i> Returnerer <i>Antal</i>-tegn fra venstre i tegnstrengen <i>kildeStreng</i>. Hvis du udelader <i>Antal</i>, returneres alle <i>kildeStreng</i>-variable.	$left("Hello", 2)$ "He"

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*

$\text{left}(x < 3)$ x

Returnerer venstre af en ligning eller ulighed.

libShortcut()

libShortcut(*BibNavneStreng*,
GenvejNavneStreng

[, *BibPrivFlag*]) $\Rightarrow$ *liste med variable*

Opretter en variabelgruppe i den aktuelle opgave, som indeholder referencer til alle objekter i det specificerede biblioteksdokument *bibNavneStreng*. Tilføjer også gruppemedlemmerne 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 30.

For at slette en variabelgruppe, se **DelVar** på side 50.

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 }
```

limit(*Udtr1*, *Var*, *Punkt* [, *Retning*])
 $\Rightarrow$ *udtryk*

$$\lim_{x \rightarrow 5} (2 \cdot x + 3) = 13$$

limit(*Liste1*, *Var*, *Punkt* [, *Retning*])
 $\Rightarrow$ *liste*

$$\lim_{x \rightarrow 0^+} \left(\frac{1}{x} \right) = \infty$$

limit(*Matrix1*, *Var*, *Punkt* [, *Retning*])
 $\Rightarrow$ *matrix*

$$\lim_{x \rightarrow 0} \left(\frac{\sin(x)}{x} \right) = 1$$

Returnerer den ønskede grænseværdi.

$$\lim_{h \rightarrow 0} \left(\frac{\sin(x+h) - \sin(x)}{h} \right) = \cos(x)$$

Bemærk: Se også **Grænseværdi skabelon**, side 7.

$$\lim_{n \rightarrow \infty} \left(\left(1 + \frac{1}{n} \right)^n \right) = e$$

Retning: negativ=fra venstre, positiv=fra højre, ellers =begge. (Ved udeladelse er *Retning* som standard begge.)

Grænseværdier i $+\infty$ og $-\infty$ konverteres altid til ensidige grænseværdier fra den begrænsede side.

Afhængigt af omstændighederne returnerer **limit()** sig selv eller undef hvis der ikke kan bestemmes en entydig grænseværdi. Dette betyder ikke nødvendigvis, at der ikke findes en entydig grænseværdi. Undef betyder, at resultatet enten er et ukendt tal af endelig eller uendelig størrelse eller en hel mængde af sådanne tal.

limit() benytter metoder som L'Hopital's regel, så der findes entydige grænseværdier, der ikke kan bestemmes. Hvis *Udtr1* indeholder andre udefinerede variable end *var*, kan det være nødvendigt at begrænse dem for at opnå et mere nøjagtigt resultat.

$$\lim_{x \rightarrow \infty} (a^x) = \text{undef}$$

$$\lim_{x \rightarrow \infty} (a^x) | a > 1 = \infty$$

$$\lim_{x \rightarrow \infty} (a^x) | a > 0 \text{ and } a < 1 = 0$$

Grænseværdier kan være meget følsomme over for afrundingsfejl. Undgå indstillingen **Tilnærmet i Auto eller tilnærmet**-tilstanden og tilnærmede tal ved beregning af grænseværdier, hvor det er muligt. Ellers vil grænseværdier, der skulle være nul eller uendelige og grænseværdier, der skulle have været endelige ikke-nul, sandsynligvis ikke være det.

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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a + b \cdot x$
stat.a, stat.b	Regressionskoefficienter
stat.r ²	Forklaringsgraden
stat.r	Korrelationskoefficient
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>

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 181.)

Alle lister skal have ens dimensioner med 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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $m \cdot x + b$
stat.m, stat.b	Regressionskoefficienter
stat.r ²	Forklaringsgraden
stat.r	Korrelationskoefficient
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>

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 181)

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 ≥ 0 .

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a+b \cdot x$
stat.a, stat.b	Regressionskoefficienter
stat.fg	Frihedsgrader
stat.r ²	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

Output-variabel	Beskrivelse
stat.SESlope	Standardfejl for hældning
stat.s	Standardfejl 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. $\hat{y}$	$a + b \cdot \text{XVærdi}$

LinRegtTest

[katalog](#) > 

LinRegtTest $X, Y[, \text{Frekv}[, \text{Hypot}]]$

Beregner en lineær regression ud fra X og Y listerne og en t test på værdien af hældningen β og korrelationskoefficienten ρ 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 ≥ 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 $\text{Hypot} = 0$

Til $H_a: \beta < 0$ og $\rho < 0$, sæt $Hypot \leq 0$

Til $H_a: \beta > 0$ og $\rho > 0$, sæt $Hypot > 0$

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 181)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a + b \cdot x$
stat.t	<i>t</i> -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 ²	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen

linSolve()

Katalog >

linSolve(SystemafLineæreLigninger, Var1, Var2, ...) $\Rightarrow$ liste

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

linSolve(LineærLign1 og LineærLign2 og ..., Var1, Var2, ...) $\Rightarrow$ liste

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

linSolve({LineærLign1, LineærLign2, ...}, Var1, Var2, ...) $\Rightarrow$ liste

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

linSolve(SystemafLineæreLigninger, {Var1, Var2, ...}) $\Rightarrow$ liste

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

linSolve(LineærLign1 og LineærLign2 og ..., {Var1, Var2, ...}) $\Rightarrow$ liste

linSolve({LineærLign1, LineærLign2, ...}, {Var1, Var2, ...}) $\Rightarrow$

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".

 Δ list()

Δ list(*Liste1*) $\Rightarrow$ liste

Δ 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()

list►mat(*Liste* [, *elementerPrRække*])
 $\Rightarrow$ matrix

list►mat($\{1,2,3\}$)	$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
list►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** (...).

Udtr ►ln⇒*udtryk*

Konverterer inputtet *Udtr* til et udtryk, der kun indeholder naturlige logaritmer (ln).

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @>1n.

$$\left(\log_{10}(x)\right) \rightarrow \ln \frac{\ln(x)}{\ln(10)}$$

ln()

  -taster

ln(*Udtr1*)⇒*udtryk*

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

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:

$$\ln(\{-3, 1.2, 5\}) \quad \{\ln(3) + \pi \cdot i, 0.182322, \ln(5)\}$$

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:

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

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

LnReg

Katalog > 

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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a + b \cdot \ln(x)$
stat.a, stat.b	Regressionskoefficienter
stat.r ²	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 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>

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()*=FuncLocal *i**1* → *i*

Loop

If *randInt(1,6)*=*randInt(1,6)*Goto *end**i*+1 → *i*

EndLoop

Lbl *end*Return *i*

EndFunc

*Done**rollcount()*

16

rollcount()

3

Lock**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 203 og **getLockInfo()**, side 86.

a:=65

65

Lock *a**Done*getLockInfo(*a*)

1

a:=75

"Error: Variable is locked."

DelVar *a*

"Error: Variable is locked."

Unlock *a**Done**a*:=75

75

DelVar *a**Done*

log()

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

log(*Udtr1* [, *Udtr2*]) ⇒ *udtryk*

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

log(*Liste1* [, *Udtr2*]) ⇒ *liste*

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

Returner -*Udtr2*-talslogaritmen til argumentet.

$$\log_3 (10) - \log_3 (5) \quad \log_3 (2)$$

Bemærk: Se også **Log-skabelon**, side 2.

Ved en liste returneres *Udtr2*-talslogaritmen til elementerne.

Hvis kompleks formattilstand er reel:

$$\log_{10} (\{-3, 1.2, 5\}) \quad \text{Error: Non-real result}$$

Hvis *Udtr2* udelades, anvendes 10-talslogaritmen.

Hvis kompleks formattilstand er rektangulær:

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

log(*kvadratMatrix1* [, *Udtr2*]) ⇒ *kvadratMatrix*

I vinkeltilstanden radian og rektangulært komplekst format:

$$\log_{10} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \\ \begin{bmatrix} 0.795387 + 0.753438 \cdot i & 0.003993 - 0.64747 \cdot i \\ 0.194895 - 0.315095 \cdot i & 0.462485 + 0.27079 \cdot i \\ -0.115909 - 0.904706 \cdot i & 0.488304 + 0.77746 \cdot i \end{bmatrix}$$

Returnerer *Udtr2*-talsmatrixlogaritmen til *kvadratMatrix1*. Dette er ikke det samme som at beregne *Udtr2*-talslogaritmen til hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

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

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

Hvis tal-argumentet udelades, anvendes 10-talslogaritmen.

logbase

Katalog >

Udtr1 ▶ **logbase**(*Udtr2*) ⇒ *udtryk*

Omformer inputudtrykket til et udtryk, der anvender *Udtr2*-talslogaritmen.

$$\log_3 (10) - \log_5 (5) \blacktriangleright \logbase(5) \quad \frac{\log_5 \left(\frac{10}{3} \right)}{\log_5 (3)}$$

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

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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

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 <i>rollcount()</i> =Func	
Local <i>i</i>	
$1 \rightarrow i$	
Loop	
If $\text{randInt}(1,6)=\text{randInt}(1,6)$	
Goto <i>end</i>	
$i+1 \rightarrow i$	
EndLoop	
Lbl <i>end</i>	
Return <i>i</i>	
EndFunc	
	Done
<i>rollcount()</i>	16
<i>rollcount()</i>	3

LU

Katalog >

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 rækkeombytninger under beregningen) i *pMatrix*.

$$lMatrix \cdot uMatrix = pMatrix \cdot matrix$$

$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$
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>	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

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 **ctrl** **enter** 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 rækkeombytninger.

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow mI$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
LU $mI, lower, upper, perm$	Done
lower	$\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \end{bmatrix}$
upper	$\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$
perm	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

M

mat▶list()

Katalog >

mat▶list($Matrix$) $\Rightarrow$ liste

Returner en liste bestående af elementerne i $Matrix$. Elementerne kopieres fra $Matrix$ række for række.

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

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

max()

Katalog >

max($Udtr1, Udtr2$) $\Rightarrow$ udtryk

max($Liste1, Liste2$) $\Rightarrow$ liste

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.

$\max\{2, 3, 1, 4\}$	2,3
$\max\{\{1, 2\}, \{-4, 3\}\}$	$\{1, 3\}$

max()Katalog > **max(List)**⇒*udtryk* $\max(\{0,1,-7,1.3,0.5\})$ 1.3Returnerer det største element i *liste*.**max(MatrixI)**⇒*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 *MatrixI*.

Tomme (ugyldige) elementer ignoreres.

Yderligere oplysninger om tomme elementer findes på side 243

Bemærk: Se også **fMax()** og **min()**.**mean()**Katalog > **mean(Liste[, hyppighedsliste])**⇒*udtryk* $\text{mean}(\{0.2,0.1,-0.3,0.4\})$ 0.26Returnerer 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(MatrixI[, Hyppighedsmatrix])**⇒*matrix*

I rektangulært vektorformat:

 $\text{mean}\left(\begin{bmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{bmatrix}\right)$ $\begin{bmatrix} -0.133333 & 0.833333 \end{bmatrix}$ Returnerer en rækkevektor af middelværdierne af alle kolonner i *MatrixI*.Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *MatrixI*. $\text{mean}\left(\begin{bmatrix} \frac{1}{5} & 0 \\ -1 & 3 \\ \frac{2}{5} & \frac{-1}{2} \end{bmatrix}\right)$ $\begin{bmatrix} \frac{-2}{15} & \frac{5}{6} \end{bmatrix}$

Tomme (ugyldige) elementer ignoreres.

Yderligere oplysninger om tomme elementer findes på side 243.

 $\text{mean}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix},\begin{bmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{bmatrix}\right)$ $\begin{bmatrix} \frac{47}{15} & \frac{11}{3} \end{bmatrix}$ **median()**Katalog > **median(Liste[, Hyppighedsliste])**⇒*udtryk* $\text{median}(\{0.2,0.1,-0.3,0.4\})$ 0.2Returnerer medianen af elementerne i *Liste*.

Hvert *Hyppighedsliste*-element tæller antallet af forekomster i ubrudt rækkefølge for de tilsvarende elementer i *Liste*.

median(*MatrixI*[, *Hyppighedsmatrix*])
⇒*matrix*

Returnerer en rækkevektor med medianerne af kolonnerne i *MatrixI*.

Hvert *Hyppighedsmatrix*-element tæller antallet af forekomster i en ubrudt rækkefølge af det tilsvarende element i *MatrixI*.

median	$\begin{bmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{bmatrix}$	$[0.4 \quad -0.3]$
--------	---	--------------------

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 243.

MedMed

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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

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 <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>

mid()

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 ≥ 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*.

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()Katalog > 

Antal skal være ≥ 0 . Hvis antal = 0, returneres en tom liste.

mid(kildeStrengListe, Start[, Antal])
 $\Rightarrow$ liste

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

Returnerer *Antal* strenge fra listen med strenge *kildeStrengListe* begyndende med element nummer *Start*.

min()Katalog > 

min(Udtr1, Udtr2) $\Rightarrow$ udtryk

min(2.3, 1.4)	1.4
---------------	-----

min(Liste1, Liste2) $\Rightarrow$ liste

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

min(Matrix1, Matrix2) $\Rightarrow$ 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) $\Rightarrow$ udtryk

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

Returnerer det mindste element af *Liste*.

min(Matrix1) $\Rightarrow$ matrix

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

Returnerer en rækkevektor med det mindste element i hver kolonne i *Matrix1*.

Bemærk: Se også **fMin()** og **max()**.

mirr()Katalog > 

mirr

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

<i>list1</i> := { 6000, -8000, 2000, -3000 }	{ 6000, -8000, 2000, -3000 }
<i>list2</i> := { 2, 2, 2, 1 }	{ 2, 2, 2, 1 }
mirr(4.65, 12, 5000, <i>list1</i> , <i>list2</i>)	13.41608607

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 96.

mod()

mod(*Udtr1*, *Udtr2*) $\Rightarrow$ *udtryk*

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

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

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

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

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

Returnerer det første argument modulo andet argument som defineret efter definitionen:

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

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

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

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

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

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 153

mRow()

mRow(*Udtr*, *Matrix1*, *Indeks*) $\Rightarrow$ *matrix*

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

mRow()Katalog > 

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks* af *Matrix1* ganget med *Udtr.*

mRowAdd()Katalog > 

mRowAdd(*Udtr*, *Matrix1*, *Indeks1*, *Indeks2*) $\Rightarrow$ *matrix*

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks2* af *Matrix1* erstattet med:

$$\begin{array}{l} \text{mRowAdd}\left(-3, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix} \\ \text{mRowAdd}\left(n, \begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} a & b \\ a \cdot n + c & b \cdot n + d \end{bmatrix} \end{array}$$

$Udtr \times \text{række } Indeks1 + \text{række } Indeks2$
Indeks2

MultRegKatalog > 

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

Beregner multiple lineære regressioner af listen *Y* på listerne *X1*, *X2*, ..., *X10*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 181.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

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 ²	Koefficient af multipel forklaringsgrad
stat.i. $\hat{y}$ List	$\hat{y}$ Liste = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualer fra regressionen

MultRegIntervalsKatalog > 

MultRegIntervals *Y*, *X1*[*...*,*X2*[*...*,*X3*[*...*,*...*[*...*,*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 181.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

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.dfError)	Frihedsgrader for fejl
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	Prædiktionsintervalsmargin for fejl
stat.SEPred	Standardfejl for prædiktion
stat.bList	Liste med regressionskoefficienter, $\{b_0, b_1, b_2, \dots\}$
stat.Resid	Residualer fra regressionen

MultRegTests

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 181.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

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 ²	Koefficient af multipel forklaringsgrad
stat.AdjR ²	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
stat.SSReg	Kvadraternes regressionsum
stat.MSReg	Middelkvadrat af regression
stat.dfFejl	Frihedsgrader for fejl
stat.SSError	fejl, kvadratsum
stat.MSError	fejl, middelkvadrat
stat.bList	{ $b_0, b_1, \dots$ } 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

BoolskUdtryk1 **nand** *BoolskUdtryk2*
returnerer *boolsk udtryk*

$x \geq 3$ and $x \geq 4$	$x \geq 4$
$x \geq 3$ nand $x \geq 4$	$x < 4$

BoolskListe1 **nand** *BoolskListe2* returnerer
Boolsk liste

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.

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

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

nCr()		Katalog > 
nCr(Udtr1, Udtr2)⇒udtryk	$nCr(z, 3)$	$\frac{z \cdot (z-2) \cdot (z-1)}{6}$
For heltal <i>Udtr1</i> og <i>Udtr2</i> med $Udtr2 \geq Udtr2 \geq 0$, nCr() er antallet af kombinationer af <i>Udtr1</i> ting taget <i>Udtr2</i> ad gangen. (Dette kendes også som en binomial koefficient). Begge argumenter kan være heltal eller symbolske udtryk.	$Ans z=5$	10
	$nCr(z, c)$	$\frac{z!}{c! \cdot (z-c)!}$
	Ans	$\frac{1}{c!}$
	$nPr(z, c)$	$c!$
nCr(Udtr, 0)⇒1		
nCr(Udtr, negativtHeltal)⇒0		
nCr(Udtr, positivtHeltal)⇒ Udtr · (Udtr-1) ... (Udtr-positivtHeltal+1)/ positivtHeltal!		
nCr(Udtr, ikkeHeltal)⇒udtryk!/(Udtr-ikkeHeltal)! · ikkeHeltal!		
nCr(Liste1, Liste2)⇒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)⇒matrix	$nCr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$
Returnerer en matrix af kombinationer baseret på de sammenhørende elementpar i de to matricer. Argumenterne skal være matricer af samme størrelse.		

nDerivative()		Katalog > 
nDerivative(Udtr1, Var=Værdi[, Orden]) ⇒værdi	$nDerivative(x , x=1)$	1
	$nDerivative(x , x) x=0$	undef
nDerivative(Udtr1, Var[, Orden]) Var=Værdi⇒værdi	$nDerivative(\sqrt{x-1}, x) x=1$	undef
Returnerer den numeriske differentialkvotient udregnet med en automatisk differentiationsmetode.		

nDerivative()Katalog > 

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

Differentialkvotientens orden skal være **1** eller **2**.

newList()Katalog > 

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

newList (4)	{0,0,0,0}
--------------------	-----------

Returnerer en liste med en dimension af *antalElementer*. Hvert element er nul.

newMat()Katalog > 

newMat(*antalRækker*, *antalKolonner*)
 $\Rightarrow$ *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*) $\Rightarrow$ *værdi*

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

nfMax(*Udtr*, *Var*, *nedreGrænse*) $\Rightarrow$ *værdi*

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

nfMax(*Udtr*, *Var*, *nedreGrænse*,
øvreGrænse) $\Rightarrow$ *værdi*

nfMax(*Udtr*, *Var*) | *nedreGrænse* $\leq$ *Var* $\leq$ *øvreGrænse* $\Rightarrow$ *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.

Bemærk: Se også **fMax()** og **d()**.

nfMin(*Udtr*, *Var*) $\Rightarrow$ *værdi*

nfMin(*Udtr*, *Var*, *nedreGrænse*) $\Rightarrow$ *værdi*

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.

Bemærk: Se også **fMin()** og **d()**.

$\text{nfMin}(x^2 + 2 \cdot x + 5, x)$	-1.
$\text{nfMin}(0.5 \cdot x^3 - x - 2, x, -5.5)$	-5.

nInt()

nInt(*Udtr1*, *Var*, *nedre*, *øvre*) $\Rightarrow$ *udtryk*

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.

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.

Bemærk: Se også $\int()$, side 216.

$\text{nInt}(e^{-x^2}, x, -1, 1)$	1.49365
-----------------------------------	---------

$\text{nInt}(\cos(x), x, \pi, \pi + 1 \cdot 10^{-12})$	-1.04144E-12
$\int_{-\pi}^{\pi + 10^{-12}} \cos(x) dx \cdot \sin\left(\frac{1}{1000000000000}\right)$	

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

nom(*effektivRente*, *CpY*) $\Rightarrow$ *værdi*

nom(5.90398,12)

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 60.

nor  -taster

BoolskUdtryk1 **nor** *BoolskUdtryk2*
returnerer *boolsk udtryk*

 $x \geq 3$ or $x \geq 4$ $x \geq 3$ $x \geq 3$ nor $x \geq 4$ $x < 3$

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* $\Rightarrow$ *heltal*

3 or 4

7

3 nor 4

-8

 $\{1,2,3\}$ or $\{3,2,1\}$ $\{3,2,3\}$ $\{1,2,3\}$ nor $\{3,2,1\}$ $\{-4,-3,-4\}$

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 hexadecimalt indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talsystem)

norm()Katalog > **norm**(*Matrix*) $\Rightarrow$ *udtryk*

$$\text{norm}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) \quad \sqrt{a^2+b^2+c^2+d^2}$$

norm(*Vektor*) $\Rightarrow$ *udtryk*

$$\text{norm}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right) \quad \sqrt{30}$$

Returnerer Frobenius-normen.

$$\text{norm}\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right) \quad \sqrt{5}$$

$$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \quad \sqrt{5}$$

normalLine()Katalog > **normalLine**(*Udtr1*,*Var*,*Punkt*) $\Rightarrow$ *udtryk*

$$\text{normalLine}(x^2, x, 1) \quad \frac{3}{2} \frac{x}{2}$$

normalLine(*Udtr1*,*Var*=*Punkt*) $\Rightarrow$ *udtryk*

$$\text{normalLine}((x-3)^2-4, x, 3) \quad x=3$$

Returnerer normallinjen til kurven repræsenteret ved *Udtr1* i punktet angivet i *Var*=*Punkt*.

$$\text{normalLine}\left(\frac{1}{x^3}, x=0\right) \quad 0$$

Vær sikker på, at den uafhængige variabel ikke er defineret. For eksempel, Hvis *f1*(*x*):=5 og *x*:3, så returnerer **normalLine**(*f1*(*x*),*x*,2) "false."

$$\text{normalLine}(\sqrt{|x|}, x=0) \quad \text{undef}$$

normCdf()Katalog > **normCdf**(*nedreGrænse*,*øvreGrænse*[,*μ*[,*σ*]])

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

Beregner normalfordelingens sandsynlighed mellem *nedreGrænse* og *øvreGrænse* for de angivne *μ* (standard=0) og *σ* (standard=1).

For $P(X \leq \text{øvreGrænse})$, sæt *nedreGrænse*= $-\infty$.

normPdf()Katalog > 

normPdf(*XVærdi*[,*μ*,*σ*]) $\Rightarrow$ *tal* hvis *XVærdi* er et tal, *liste* hvis *XVærdi* er en liste

Beregner tæthedsfunktionen for normalfordelingen i en angivet *XVærdi* for de angivne *μ* og *σ*.

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 18.

$\text{not}(2 \geq 3)$	true
$\text{not}(x < 2)$	$x \geq 2$
not not innocent	<i>innocent</i>

I hexadecimal tilstand:

Vigtigt: Tallet nul, ikke bogstavet O.

not 0h7AC36	0hFFFFFFFFFFFF853C9
-------------	---------------------

I binær tilstand:

0b100101 $\blacktriangleright$ Base10	37
not 0b100101	
0b11111111111111111111111111111111 $\blacktriangleright$	
not 0b100101 $\blacktriangleright$ Base10	-38

Du kan se hele resultatet ved at trykke på $\blacktriangle$ 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(*Udtr1*, *Udtr2*) $\Rightarrow$ *udtryk*

For heltal er *Udtr1* og *Udtr2* med *Udtr1* $\geq$ *Udtr2* ≥ 0 , **nPr()** antallet af permutationer af *Udtr1* ting taget *Udtr2* ad gangen. Begge argumenter kan være heltal eller symbolske udtryk.

nPr(*Udtr*, 0) $\Rightarrow$ 1

nPr(*Udtr*, *negHeltal*) $\Rightarrow 1 / ((Udtr+1) \cdot (Udtr+2) \dots (Udtr-negHeltal))$

nPr(*Udtr*, *posHeltal*) $\Rightarrow Udtr \cdot (Udtr-1) \dots (Udtr-posHeltal+1)$

nPr(*Udtr*, *ikkeHetal*) $\Rightarrow Udtr! / (Udtr-ikkeHetal)!$

$\text{nPr}(z, 3)$	$z \cdot (z-2) \cdot (z-1)$
$\text{Ans} z=5$	60
$\text{nPr}(z, -3)$	$\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$
$\text{nPr}(z, c)$	$\frac{z!}{(z-c)!}$
$\text{Ans} \cdot \text{nPr}(z-c, -c)$	1

nPr()Katalog > **nPr**(*Liste1*, *Liste2*) $\Rightarrow$ *liste* $\text{nPr}(\{5,4,3\}, \{2,4,2\})$ $\{20,24,6\}$

Returnerer en liste med permutationer baseret på de sammenhørende elementpar i de to lister. Argumenterne skal være lister af samme størrelse.

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

$$\text{nPr}\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right) \quad \begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$$

Returnerer en matrix med permutationer baseret på de sammenhørende elementpar i de to matricer. Argumenterne skal være matricer af samme størrelse.

npv()Katalog > **npv**(*RenteSats*, *CFO*, *CFListe*, *CFFrekv*)
$$\begin{aligned} \text{list1} &:= \{6000, -8000, 2000, -3000\} \\ &\quad \{6000, -8000, 2000, -3000\} \\ \text{list2} &:= \{2, 2, 2, 1\} & \{2, 2, 2, 1\} \\ \text{npv}(10, 5000, \text{list1}, \text{list2}) & \quad 4769,91 \end{aligned}$$

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*.

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()Katalog > **nSolve**(*Ligning*, *Var*[=*Gæt*]) $\Rightarrow$ *tal* eller *fejlstreng* $\text{nSolve}(x^2 + 5 \cdot x - 25 = 9, x)$ 3.84429 $\text{nSolve}(x^2 = 4, x = -1)$ -2.**nSolve**(*Ligning*, *Var*[=*Gæt*], *nedreGrænse*) $\Rightarrow$ *tal* eller *fejlstreng* $\text{nSolve}(x^2 = 4, x = 1)$ 2.

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() er ofte meget hurtigere end **solve()**
eller **zeros()**, især hvis "|" operatoren
anvendes til at begrænse søgningen til et
lille interval, der indeholder nøjagtig én
simpler løsning.

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."

Bemærk: Se også **cSolve()**, **cZeros()**, **solve()**
og **zeros()**.

Bemærk: Hvis der er flere løsninger, kan du
anvende et gæt til at finde en partikulær
løsning.

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

O

OneVar

OneVar [1,]*X* [,*Hyppighed*]
[,*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 181.)

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 ≥ 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 243

Output-variabel	Beskrivelse
stat. $\bar{x}$	Gennemsnit af x -værdier
stat. Σx	Summen af x -værdier
stat. Σx^2	Summen af x^2 værdier
stat.sx	Standardafvigelse for målingen for x
stat. x	Populations standardafvigelse for x
stat.n	Antal datapunkter
stat.MinX	Minimum af x -værdier
stat. $Q_1 X$	1. kvartil af x
stat.MedianX	Median af x .
stat. $Q_3 X$	3. kvartil af x .
stat.MaxX	Maksimum af x -værdier.
stat.SSX	Summen af kvadraterne på afvigelser fra middelværdien for x .

BoolskUdtryk1 **or** *BoolskUdtryk2*
returnerer *boolsk udtryk*

$x \geq 3$ or $x \geq 4$	$x \geq 3$
--------------------------	------------

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

Define $g(x)$ = Func	Done
----------------------	------

BoolskMatrix1 **or** *BoolskMatrix2* returnerer
Boolsk matrix

If $x \leq 0$ or $x \geq 5$
Goto end
Return $x \cdot 3$
Lbl end
EndFunc

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

$g(3)$	9
--------	---

Returnerer true, hvis enten et eller begge
udtryk kan reduceres til true. Returnerer kun
false, hvis begge udtryk evalueres til false.

$g(0)$	A function did not return a value
--------	-----------------------------------

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* $\Rightarrow$ *heltal*

I hexadecimal 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
talsystemtilstand.

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

Vigtigt: Tallet nul, ikke bogstavet O.

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

I binær tilstand:

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.

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 18.

Bemærk: Se xor.

ord()

Katalog > 

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

ord("hello") 104

ord(*Liste l*) $\Rightarrow$ *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()

Katalog > 

P►Rx(*rUdtr*, *θUdtr*) $\Rightarrow$ *udtryk*

I vinkeltilstanden Radian:

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

$P\blacktriangleright Rx(r, \theta) \cos(\theta) \cdot r$

P►Rx(*rMatrix*, *θMatrix*) $\Rightarrow$ *matrix*

$P\blacktriangleright Rx(4, 60^\circ) 2$

Returnerer den ækvivalente x-koordinat til parret (r, θ).

$P\blacktriangleright Rx\left(\left\{-3, 10, 1.3\right\}, \left\{\frac{\pi}{3}, \frac{-\pi}{4}, 0\right\}\right) \left\{\frac{-3}{2}, 5 \cdot \sqrt{2}, 1.3\right\}$

Bemærk: Argumentet θ tolkes i grader, nygrader eller radianer afhængigt af den aktuelle vinkeltilstand. Hvis argumentet er et udtryk, kan du anvende °, ^G eller ^r til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

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

P►Ry()

Katalog > 

P►Ry(*rUdtr*, *θUdtr*) $\Rightarrow$ *udtryk*

I vinkeltilstanden Radian:

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

$P\blacktriangleright Ry(r, \theta) \sin(\theta) \cdot r$

P►Ry(*rMatrix*, *θMatrix*) $\Rightarrow$ *matrix*

$P\blacktriangleright Ry(4, 60^\circ) 2 \cdot \sqrt{3}$

Returnerer den ækvivalente y-koordinat til parret (r, θ).

$P\blacktriangleright Ry\left(\left\{-3, 10, 1.3\right\}, \left\{\frac{\pi}{3}, \frac{-\pi}{4}, 0\right\}\right) \left\{\frac{-3 \cdot \sqrt{3}}{2}, -5 \cdot \sqrt{2}, 0\right\}$

Bemærk: Argumentet θ 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@>Ry** (...).

PassErr

PassErr

Se et eksempel på **PassErr** i Eksempel2 under **Try**-kommandoen, side 196.

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 26, og **Try**, side 196.

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

piecewise()

piecewise(*Udtr1* [, *Betingelse1* [, *Udtr2* [, *Betingelse2* [, ...]]]])

Returnerer definitioner for en stykkevis funktion i form af en liste. Du kan også oprette stykkevise definitioner ved hjælp af en skabelon.

Bemærk: Se også **Stykkevis skabelon**, side 3.

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

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

poissCdf(λ , *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 Poisson-distribution med en
 angivet middelværdi λ .

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

poissPdf()

poissPdf(λ , *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 λ .

►Polar

Vektor ►Polar

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 150.

kompleksVærdi ►Polar

$$\begin{array}{l} \begin{bmatrix} 1 & 3 \end{bmatrix} \text{►Polar} \quad \begin{bmatrix} 3.16228 & \angle 1.24905 \end{bmatrix} \\ \begin{bmatrix} x & y \end{bmatrix} \text{►Polar} \quad \left[\sqrt{x^2 + y^2} \angle \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1} \left(\frac{x}{y} \right) \right] \end{array}$$

I vinkeltilstanden Radian:

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.

$$\begin{array}{l} (3+4 \cdot i) \blacktriangleright \text{Polar} \\ e^{i \cdot \left(\frac{\pi}{2} - \tan^{-1} \left(\frac{3}{4} \right) \right)} \cdot 5 \end{array}$$

$$\begin{array}{l} \left(4 \angle \frac{\pi}{3} \right) \blacktriangleright \text{Polar} \\ e^{\frac{i \cdot \pi}{3}} \cdot 4 \end{array}$$

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Grader:

$$(3+4 \cdot i) \blacktriangleright \text{Polar} \quad \left(5 \angle 90 - \tan^{-1} \left(\frac{3}{4} \right) \right)$$

polyCoeffs()

polyCoeffs(Poly [Var]) ⇒ *liste*

Returnerer en liste over koefficienterne i polynomiet *Poly* vedrørende variabelen *Var*.

Poly skal være et polynomielt udtryk i *Var*. Vi anbefaler, at du ikke udelader *Var*, med mindre *Poly* er et udtryk i en enkelt variabel.

$$\text{polyCoeffs}(4 \cdot x^2 - 3 \cdot x + 2, x) \quad \{4, -3, 2\}$$

$$\text{polyCoeffs}((x-1)^2 \cdot (x+2)^3) \quad \{1, 4, 1, -10, -4, 8\}$$

Udvider polynomiet og vælger *x* for den udeladte *Var*.

$$\text{polyCoeffs}((x+y+z)^2, x) \quad \{1, 2 \cdot (y+z), (y+z)^2\}$$

$$\text{polyCoeffs}((x+y+z)^2, y) \quad \{1, 2 \cdot (x+z), (x+z)^2\}$$

$$\text{polyCoeffs}((x+y+z)^2, z) \quad \{1, 2 \cdot (x+y), (x+y)^2\}$$

polyDegree()Katalog > **polyDegree(Poly [,Var])**⇒*værdi*

Returnerer graden af det polynomielle udtryk *Poly* med hensyn til variablen *Var*. Hvis du udelader *Var*, vælger **polyDegree()**-funktionen en standardværdi fra variablene i polynomiet *Poly*.

Poly skal være et polynomielt udtryk i *Var*. Vi anbefaler, at du ikke udelader *Var*, med mindre *Poly* er et udtryk i en enkelt variabel.

$\text{polyDegree}(5)$	0
$\text{polyDegree}(\ln(2) + \pi, x)$	0

Konstante polynomier

$\text{polyDegree}(4 \cdot x^2 - 3 \cdot x + 2, x)$	2
$\text{polyDegree}((x-1)^2 \cdot (x+2)^3)$	5

$\text{polyDegree}((x+y^2+z^3)^2, x)$	2
---------------------------------------	---

$\text{polyDegree}((x+y^2+z^3)^2, y)$	4
---------------------------------------	---

$\text{polyDegree}((x-1)^{10000}, x)$	10000
---------------------------------------	-------

Graden kan også udledes, selvom koefficienterne ikke kan. Det skyldes, at graden kan udtrækkes uden at ekspandere polynomiet.

polyEval()Katalog > **polyEval(Liste1, Udtryk1)**⇒*udtryk***polyEval(Liste1, Liste2)**⇒*udtryk*

Fortolker første argument som koefficienten til et polynomium i faldende grad og returnerer polynomiet beregnet for værdien af det andet argument.

$\text{polyEval}(\{a, b, c\}, x)$	$a \cdot x^2 + b \cdot x + c$
$\text{polyEval}(\{1, 2, 3, 4\}, 2)$	26
$\text{polyEval}(\{1, 2, 3, 4\}, \{2, -7\})$	$\{26, -262\}$

polyGcd()Katalog > **polyGcd(Udtryk1, Udtryk2)**⇒*udtryk*

Returnerer den største fælles divisor af de to argumenter.

Udtryk1 og *Udtryk2* skal være polynomielle udtryk.

Liste-, matrix-, og Boolske argumenter er ikke tilladt.

$\text{polyGcd}(100, 30)$	10
$\text{polyGcd}(x^2 - 1, x - 1)$	$x - 1$
$\text{polyGcd}(x^3 - 6 \cdot x^2 + 11 \cdot x - 6, x^2 - 6 \cdot x + 8)$	$x - 2$

polyQuotient()

Katalog > 

polyQuotient(Poly1, Poly2 [Var]) ⇒ *udtryk*

Returnerer kvotienten af polynomiet *Poly1* divideret med polynomiet *Poly2* med hensyn til den angivne variabel *Var*.

Poly1 og *Poly2* skal være polynomielle udtryk i *Var*. Vi anbefaler, at du ikke udelader *Var*, med mindre *Poly1* og *Poly2* er udtryk i den samme variabel.

$\text{polyQuotient}(x-1, x-3)$	1
$\text{polyQuotient}(x-1, x^2-1)$	0
$\text{polyQuotient}(x^2-1, x-1)$	$x+1$
$\text{polyQuotient}(x^3-6 \cdot x^2+11 \cdot x-6, x^2-6 \cdot x+8)$	x
$\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, x)$	$y-z$
$\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, y)$	$2 \cdot x-y+2 \cdot z$
$\text{polyQuotient}((x-y) \cdot (y-z), x+y+z, z)$	$-(x-y)$

polyRemainder()

Katalog > 

polyRemainder(Poly1, Poly2 [Var])
⇒ *udtryk*

Returnerer resten af polynomiet *Poly1* divideret med polynomiet *Poly2* med hensyn til den angivne variabel *Var*.

Poly1 og *Poly2* skal være polynomielle udtryk i *Var*. Vi anbefaler, at du ikke udelader *Var*, med mindre *Poly1* og *Poly2* er udtryk i den samme enkelte variabel.

$\text{polyRemainder}(x-1, x-3)$	2
$\text{polyRemainder}(x-1, x^2-1)$	$x-1$
$\text{polyRemainder}(x^2-1, x-1)$	0
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, x)$	$-(y-z) \cdot (2 \cdot y+z)$
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, y)$	$-2 \cdot x^2-5 \cdot x \cdot z-2 \cdot z^2$
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, z)$	$(x-y) \cdot (x+2 \cdot y)$

polyRoots()Katalog > **polyRoots**(*Poly*, *Var*) ⇒

$$\text{polyRoots}(y^3+1, y) \quad \{-1\}$$

polyRoots(*ListeAfKoeff*) ⇒

$$\text{cPolyRoots}(y^3+1, y) \\ \left\{-1, \frac{1}{2} - \frac{\sqrt{3}}{2}i, \frac{1}{2} + \frac{\sqrt{3}}{2}i\right\}$$

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 en variabel.

$$\text{polyRoots}(\{1, 2, 1\}) \quad \{-1, -1\}$$

Den anden syntaks, **polyRoots**(*ListeAfKoeff*), returnerer en liste med reelle rødder for koefficienterne i *ListeAfKoeff*.

Bemærk: Se også **cPolyRoots()**, side 37.

PowerRegKatalog > 

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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot (x)^b$
stat.a, stat.b	Regressionskoefficienter
stat.r ²	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

Katalog >

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.

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

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

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

450 60

60 30

30 0

GCD=30

Done

product()

Katalog > **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(MatrixI[, Start[, slut]]) $\Rightarrow$ *matrix*

Returnerer en rækkevektor med produkterne af elementerne i kolonnerne i *MatrixI*. *Start* og *slut* er valgfri. De angiver en serie af rækker.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 243.

$\text{product}\left(\left\{1,2,3,4\right\}\right)$	24
$\text{product}\left(\left\{2,x,y\right\}\right)$	$2 \cdot x \cdot y$
$\text{product}\left(\left\{4,5,8,9\right\},2,3\right)$	40

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

propFrac()

Katalog > **propFrac(UdtrI[, Var])** $\Rightarrow$ *udtryk*

propFrac(rationalt_tal) returnerer *rationalt_tal* som summen af et heltal og en brøk med samme fortegn og en større nævner end tæller.

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.

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

$\text{propFrac}\left(\frac{x^2+x+1}{x+1} + \frac{y^2+y+1}{y+1}, x\right)$	$\frac{1}{x+1} + x + \frac{y^2+y+1}{y+1}$
$\text{propFrac}(\text{Ans})$	$\frac{1}{x+1} + x + \frac{1}{y+1} + y$

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.

For rationale udtryk er **propFrac()** et hurtigere men mindre vidtgående alternativ til **expand()**.

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-triangular.

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

Tallet med flydende komma (9). i m1 får resultaterne beregnet med flydende komma.

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

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

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*.

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow mI \quad \begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

QR *mI,qm,rm* Done

<i>qm</i>	$\frac{m}{\sqrt{m^2+o^2}}$	$\frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}}$
	$\frac{o}{\sqrt{m^2+o^2}}$	$\frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}}$
<i>rm</i>	$\sqrt{m^2+o^2}$	$\frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}}$
	0	$\frac{m \cdot p - n \cdot o}{\sqrt{m^2+o^2}}$

QuadReg

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

Beregner andengrads polynomiel 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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Regressionskoefficienter
stat.r ²	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>

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 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder
kategorikoder 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 243.

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 ²	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►P0()

R►P0 (*xExpr*, *yExpr*) ⇒ *udtryk*

I vinkeltilstanden Grader:

R►P0 (*xList*, *yList*) ⇒ *liste*

R►P0 (*xMatrix*, *yMatrix*) ⇒ *matrix*

$$R \blacktriangleright P0(x, y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Returnerer den ækvivalente θ -koordinat for (*x*,*y*) argumentparret.

I vinkeltilstanden Nygrader:

Bemærk: Resultatet returneres som en vinkel i grader, nygrader eller radian afhængig af den aktuelle vinkeltilstand.

$$R \blacktriangleright P0(x, y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

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

I vinkeltilstanden Radian:

$$R \blacktriangleright P0(3, 2) \quad \tan^{-1}\left(\frac{2}{3}\right)$$

$$R \blacktriangleright P0\left(\left[3 \quad -4 \quad 2\right], \left[0 \quad \frac{\pi}{4} \quad 1.5\right]\right) \quad \left[0 \quad \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} \quad 0.643501\right]$$

R► Pr()Katalog > **R► Pr** (*xExpr*, *yExpr*) ⇒ *udtryk*

I vinkeltilstanden Radian:

R► Pr (*xList*, *yList*) ⇒ *liste***R► Pr** (*xMatrix*, *yMatrix*) ⇒ *matrix*Returnerer den ækvivalente r-koordinat for (*x*,*y*) argumentparret.**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **R@>Pr** (...).

$$\begin{array}{l} \text{R►Pr}(3,2) \qquad \qquad \qquad \sqrt{13} \\ \text{R►Pr}(x,y) \qquad \qquad \qquad \sqrt{x^2+y^2} \\ \text{R►Pr}\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right) \\ \qquad \qquad \qquad \begin{bmatrix} 3 & \frac{\sqrt{\pi^2+256}}{4} & 2.5 \end{bmatrix} \end{array}$$

► RadKatalog > *Expr1* ► **Rad** ⇒ *udtryk*

I vinkeltilstanden grader:

Konverterer argumentet til vinkelmåling i radian.

$$(1.5) \text{►Rad} \qquad \qquad \qquad (0.02618)^r$$

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

I vinkeltilstanden Nygrader:

$$(1.5) \text{►Rad} \qquad \qquad \qquad (0.023562)^r$$

rand()Katalog > **rand()** ⇒ *udtryk***rand**(*#Trials*) ⇒ *liste***rand()** returnerer en vilkårlig værdi mellem 0 og 1.

Angiv seed-værdien for et vilkårligt tal.

$$\begin{array}{l} \text{RandSeed } 1147 \qquad \qquad \qquad \text{Done} \\ \text{rand}(2) \qquad \qquad \qquad \{0.158206, 0.717917\} \end{array}$$

rand(*#Trials*) returnerer en liste med *#Trials* vilkårlig værdier mellem 0 og 1.**randBin()**Katalog > **randBin**(*n*, *p*) ⇒ *udtryk***randBin**(*n*, *p*, *#Trials*) ⇒ *liste***randBin**(*n*, *p*) returnerer et vilkårlig reelt tal fra en given binomialfordeling.

$$\begin{array}{l} \text{randBin}(80,0.5) \qquad \qquad \qquad 42 \\ \text{randBin}(80,0.5,3) \qquad \qquad \qquad \{41, 32, 39\} \end{array}$$

randBin(*n*, *p*, *#Trials*) returnerer en liste med *#Trials* vilkårlige reelle tal fra en given binomialfordeling.

randInt()

Katalog > 

randInt
(*lowBound, upBound*)
⇒ *udtryk*
randInt
(*lowBound, upBound*,
#Trials) ⇒ *liste*

randInt (3,10)	5
randInt (3,10,4)	{ 9,7,5,8 }

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)	8	-3	6
	-2	3	-6
	0	4	-6

Bemærk: Værdierne i denne matrix ændres,
hver gang du trykker .

randNorm()

Katalog > 

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

randNorm(μ , σ) returnerer et decimalt tal
fra den angivne normalfordeling. Det kan
være ethvert reelt tal, men vil være kraftigt
koncentreret i intervallet [$\mu-3\cdot\sigma$, $\mu+3\cdot\sigma$].

randNorm(μ , σ , *#Trials*) returnerer en liste
med *#Trials* decimaltal fra den angivne
normalfordeling.

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

randPoly()	Katalog > 	
randPoly (<i>Var</i> , <i>Order</i>) ⇒ <i>udtryk</i>	RandSeed 1147	Done
Returnerer et polynomium i <i>Var</i> af den angivne <i>Order</i> . Koefficienterne er vilkårlige heltal i området −9 til 9. Koefficienten af højeste grad vil ikke være nul.	randPoly(<i>x</i> ,5)	$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$
<i>Order</i> skal være 0–99.		

randSamp()	Katalog > 	
randSamp (<i>List</i> ,# <i>Trials</i> [, <i>noRepl</i>]) ⇒ <i>liste</i>	Define <i>list3</i> = $\{1,2,3,4,5\}$	Done
Returnerer en liste med en vilkårlig stikprøve af # <i>Trials</i> målinger fra <i>List</i> med mulighed for tilbagelægning (<i>noRepl</i> =0) eller ingen tilbagelægning (<i>noRepl</i> =1). Standardindstillingen er med tilbagelægning.	Define <i>list4</i> =randSamp(<i>list3</i> ,6)	Done
	<i>list4</i>	$\{2,3,4,3,1,2\}$

RandSeed	Katalog > 	
RandSeed <i>Number</i>	RandSeed 1147	Done
Hvis <i>Number</i> = 0, indstilles startværdien til fabriksindstillingerne for vilkårlig-tal-generatoren. Hvis <i>Number</i> ≠ 0, anvendes det til at generere to startværdier, der lagres i systemvariableerne seed1 og seed2.	rand()	0.158206

real()	Katalog > 	
real (<i>Expr1</i>) ⇒ <i>udtryk</i>	real($2+3 \cdot i$)	2
Returnerer den reelle del af argumentet.	real(<i>z</i>)	<i>z</i>
Bemærk: Alle udefinerede variable behandles som reelle variable. Se også imag() , page 92.	real($x+i \cdot y$)	<i>x</i>
real (<i>List1</i>) ⇒ <i>liste</i>	real($\{a+i \cdot b,3,i\}$)	$\{a,3,0\}$
Returnerer realdelen af alle elementer.		
real (<i>Matrix1</i>) ⇒ <i>matrix</i>	real($\begin{bmatrix} a+i \cdot b & 3 \\ c & i \end{bmatrix}$)	$\begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$
Returnerer realdelen af alle elementer.		

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 136.

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.

$$\left(3 \angle \frac{\pi}{4} \angle \frac{\pi}{6}\right) \text{►Rect}$$

$$\begin{bmatrix} 3\sqrt{2} & 3\sqrt{2} & 3\sqrt{3} \\ 4 & 4 & 2 \end{bmatrix}$$

$$\begin{bmatrix} a \cos(b) \sin(c) & a \sin(b) \sin(c) & a \cos(c) \end{bmatrix}$$

I vinkeltilstanden Radian:

$$\left(4 \cdot e^{\frac{\pi}{3}}\right) \text{►Rect} \quad 4 \cdot e^{\frac{\pi}{3}}$$

$$\left(4 \angle \frac{\pi}{3}\right) \text{►Rect} \quad 2+2\sqrt{3} \cdot i$$

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden grader:

$$\left((4 \angle 60)\right) \text{►Rect} \quad 2+2\sqrt{3} \cdot i$$

Bemærk: Du kan skrive $\angle$ ved at vælge det i symbollisten i kataloget.

ref()

ref(Matrix1[, Tol]) ⇒ *matrix*

Returnerer række-echelonformen af *Matrix1*.

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

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 har fået 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:
 $5E-14 \cdot \max(\dim(Matrix\ I)) \cdot \text{rowNorm}(Matrix\ I)$

Undgå udefinerede elementer i *Matrix I*.
De kan føre til uventede resultater.

Hvis for eksempel *a* er udefineret i følgende udtryk, vil en advarsel blive vist, og resultatet vises som:

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

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.

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

Bemærk: Se også **rref()**, page 160.

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow mI$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$
ref(<i>mI</i>)	$\begin{bmatrix} 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix}$



RefreshProbeVars

Giver dig adgang til sensordata fra alle tilkoblede sensorprober i dit TI-Basic-program.

StatusVar-værdi**Status**

statusVar =0 Normal (fortsæt med programmet)

Vernier DataQuest™-applikation er i dataindsamlingstilstand.

statusVar =1 **Bemærk:** Vernier DataQuest™-applikation skal være i målerstilstand for at denne

kommando virker. 

statusVar =2 Vernier DataQuest™-applikation blev ikke startet.

Vernier DataQuest™-

statusVar =3 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 temperatur>30 Then
```

```
Disp "For varmt"
```

```
EndIf
```

```
© Vent i 1 sekund mellem  
stikprøver
```

```
Wait 1
```

```
EndFor
```

```
Else
```

```
Disp "Ikke klar. Prøv igen  
senere"
```

```
EndIf
```

```
EndPrgm
```

Bemærk: Dette kan også bruge med TI-Innovator™-hubben.

remain()

Katalog > 

remain(*Expr1*, *Expr2*) $\Rightarrow$ *udtryk*

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

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

Returnerer resten af det første argument med hensyn til det andet argument som defineret af identiteterne:

remain(*x*,0) *x*

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

Bemærk, at som følge heraf, **remain**(-*x*,*y*) = **remain**(*x*,*y*). Resultatet er enten nul, eller det har samme fortegn som det første argument.

Bemærk: Se også **mod**(), side 119.

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

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

Request

Katalog > 

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.

- 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 i historikken.

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

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 variabelen *statusVar* til en værdi på **1**.
- I modsat fald indstilles variabelen *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 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.

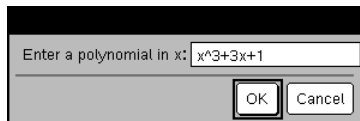
Bemærk: Se også **RequestStr**, page 154.

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:

polynomial()



Resultat efter indtastning af x^3+3x+1 og valg af **OK**:

Reelle rødder er: $\{-0.322185\}$

RequestStr *promptString*, *var*[, *DispFlag*]

Definer et program:

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 153.

```
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

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()Katalog > **right(List1[, Num])** ⇒ *liste* $\text{right}(\{1, 3, -2, 4\}, 3)$ $\{3, -2, 4\}$

Returnerer *Num*-elementer længst til højre i *List1*.

Hvis du udelader *Num*, returneres hele *List1*.

right(sourceString[, Num]) ⇒ *string* $\text{right}(\text{"Hello"}, 2)$ "lo"

Returnerer *Num*-tegn længst til højre i tegnstrengen *sourceString*.

Hvis du udelader *Num*, returneres hele *sourceString*.

right(Comparison) ⇒ *udtryk* $\text{right}(x < 3)$ 3

Returnerer højre side af en ligning eller ulighed.

rk23 ()Katalog > **rk23(Expr, Var, depVar, {Var0, VarMax}, depVar0, VarStep[, diftol])** ⇒ *matrix*

Differentialligning:

 $y' = 0.001 \cdot y \cdot (100 - y)$ og $y(0) = 10$ **rk23(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol])** ⇒ *matrix*

$$\text{rk23}(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9493	13.042	14.2

rk23(ListOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep[, diftol]) ⇒ *matrix*

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

Anvender Runge-Kutta-metoden til at løse systemet

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

Samme ligning med *diftol* sat til $1.E-6$

med $\text{depVar}(\text{Var0}) = \text{depVar0}$ i intervallet $[\text{Var0}, \text{VarMax}]$. 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.

$$\text{rk23}(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1, 1.E-6)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

Sammenlign resultatet ovenfor med den eksakte CAS løsning opnået ved brug af *deSolve()* og *seqGen()*:

$$\text{deSolve}(y' = 0.001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y)$$

$$y = \frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}$$

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 variable.

{*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 variable.

Hvis *VarStep* evalueres til et tal forskelligt fra nul: $\text{sign}(\text{VarStep}) = \text{sign}(\text{VarMax} - \text{Var0})$ og løsninger returneres ved $\text{Var0} + i \cdot \text{VarStep}$ for alle $i=0,1,2,\dots$ således at $\text{Var0} + i \cdot \text{VarStep}$ er i $[\text{var0}, \text{VarMax}]$ (der vil muligvis ikke være en løsningsværdi ved *VarMax*).

Hvis *VarStep* evalueres til nul, returneres løsningerne i "Runge-Kutta" *Var*-værdier.

diffTol er fejltolerancen (som standard 0.001).

$$\text{seqGen} \left(\frac{100 \cdot (1.10517)^t}{(1.10517)^t + 9}, t, y, \{0, 100\} \right)$$

$$\{10., 10.9367, 11.9494, 13.0423, 14.2189, 15.4\}$$

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

root(Expr) $\Rightarrow$ rod

root(Expr1, Expr2) $\Rightarrow$ rod

root(Expr) returnerer kvadratroden af *Expr*.

root(Expr1, Expr2) returnerer *Expr2*-roden af *Expr1*. *Expr1* kan være en reel eller kompleks konstant med flydende decimaler, et heltal eller en kompleks rational konstant eller et generelt symbolsk udtryk.

Bemærk: Se også **N-te rod-skabelon**, side 1.

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

$$\sqrt[3]{3} \quad \frac{1}{3}$$

$$\sqrt[3]{3.} \quad 1.44225$$

I binær tilstand:

Roterer bittene i et binært heltal. Du kan indtaste *Integer1* i ethvert talsystem; det konverteres automatisk til en 64-bit binær form med fortegn. Hvis *Integer1* er for stort til denne form, bringer en symmetrisk modulo-operation værdien ind i det korrekte område. Der er flere oplysninger under ► **Base2**, side 18.

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.

0b000000000000001111010110000110101

Bitten længst til højre roterer længst mod venstre.

giver:

0b100000000000000111101011000011010

Resultatet vises i den valgte tilstand for talsystem.

$$\text{rotate}(\text{List1}[, \#ofRotations]) \Rightarrow \text{liste}$$

Returnerer en kopi af *List1* roteret til højre eller venstre med *#of Rotations*-elementer. Ændrer ikke *List1*.

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#ofRotations* er negativt, kører rotationen mod højre. Standardindstillingen er -1 (roter en bit til højre).

$$\text{rotate}(\text{String}I[, \#ofRotations]) \Rightarrow \text{string}$$

Returnerer en kopi af *String1* roteret til højre eller venstre med *#ofRotations*-tegn. Ændrer ikke *String1*.

[illegible]

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

I hexadecimal tilstand:

rotate(0h78E)	0h3C7
rotate(0h78E,-2)	0h80000000000001E3
rotate(0h78E,2)	0h1E38

Vigtigt: Til binære eller hexadecimale indtastninger skal du som præfiks altid benytte henholdsvis 0b eller 0h (nul, ikke bogstavet O).

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"

rotate()

Katalog >

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#ofRotations* er negativt, kører rotationen mod højre. Standardindstillingen -1 (rotér en bit til højre).

round()

Katalog >

round(*Expr1*[, *digits*]) $\Rightarrow$ *udtryk*

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

Returnerer argumentet afrundet til det angivne antal cifre efter decimalpunktet.

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*]) $\Rightarrow$ *liste*

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

Returnerer en liste med elementerne afrundet til det angivne antal cifre.

round(*Matrix1*[, *digits*]) $\Rightarrow$ *matrix*

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

Returnerer en matrix med elementerne afrundet til det angivne antal cifre.

rowAdd()

Katalog >

rowAdd(*Matrix1*, *rIndex1*, *rIndex2*) $\Rightarrow$ *matrix*

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

Returnerer en kopi af *Matrix1* med rækken *rIndex2* udskiftet med summen af rækker *rIndex1* og *rIndex2*.

$\text{rowAdd}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$
--	--

rowDim()

Katalog >

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

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

Returnerer antallet af rækker i *Matrix*.

Bemærk: Se også **colDim()**, side 27.

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

rowNorm()

Katalog > 

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

Returnerer maksimum for summen af de absolutte værdier for elementerne i *Matrix*-rækkerne.

Bemærk: Alle matricens elementer skal kunne reduceres til tal. Se også **colNorm()**, side 27.

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

rowSwap()

Katalog > 

rowSwap(*Matrix1*, *rIndex1*, *rIndex2*) $\Rightarrow$ *matrix*

Returnerer *Matrix1* med rækkerne *rIndex1* og *rIndex2* byttet.

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

rref()

Katalog > 

rref(*Matrix1*, *Tol*) $\Rightarrow$ *matrix*

Returnerer den reducerede række-echelonform af *Matrix1*.

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

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*.

$$\text{rref}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- 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:
 $5\text{E}-14 \cdot \max(\dim(\text{Matrix1})) \cdot \text{rowNorm}(\text{Matrix1})$

Bemærk: Se også **ref()**, page 150.

sec() **-tast****sec**(*Udtr1*) $\Rightarrow$ *udtryk*

I vinkeltilstanden Grader:

sec(*Liste1*) $\Rightarrow$ *list*

Returnerer sekans til *Udtr1* 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 $^{\circ}$, $^{\text{G}}$ eller $^{\text{r}}$ til midlertidigt at ignorere vinkeltilstanden.

$$\begin{array}{l} \text{sec}(45) \qquad \qquad \qquad \sqrt{2} \\ \text{sec}(\{1,2,3,4\}) \qquad \left\{ \frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)} \right\} \end{array}$$

sec⁻¹() **-tast****sec⁻¹**(*Udtr1*) $\Rightarrow$ *udtryk*

I vinkeltilstanden Grader:

sec⁻¹(*Liste1*) $\Rightarrow$ *liste*

Returnerer den vinkel, hvis sekans er *Udtr1*, eller returnerer en liste med de inverse sekanser til de enkelte elementer 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 **arcsec (...)**.

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

$$\text{sec}^{-1}(\{1,2,5\}) \qquad \qquad \qquad \left\{ 0, \frac{\pi}{3}, \cos^{-1}\left(\frac{1}{5}\right) \right\}$$

sech()**Katalog** > **sech**(*Udtr1*) $\Rightarrow$ *udtryk*

$$\text{sech}(3) \qquad \qquad \qquad \frac{1}{\cosh(3)}$$

sech(*Liste1*) $\Rightarrow$ *list*

Returnerer den hyperbolske sekans til *Udtr1* eller returnerer en liste med den hyperbolske sekans til elementerne i *Liste1*.

$$\begin{array}{l} \text{sech}(\{1,2,3,4\}) \\ \left\{ \frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)} \right\} \end{array}$$

sech⁻¹(*Udtr1*) $\Rightarrow$ *udtryk*

sech⁻¹ (*Liste1*) $\Rightarrow$ *liste*

Returnerer den inverse hyperbolske sekans til *Udtr1* 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 (...)**.

I vinkeltilstanden radian og tilstanden rektangulært kompleks:

sech ⁻¹ (1)	0
sech ⁻¹ { { 1, -2, 2, 1 } }	
$\left\{ 0, \frac{2 \cdot \pi}{3} \cdot i, 8, 8 \cdot 10^{-15} + 1, 07448 \cdot i \right\}$	

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 80), **GetStr** (side 87), og **eval()** (side 64).

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.

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

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

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

seqGen()

Katalog >

seqGen(Expr, Var, depVar, {Var0, VarMax}[, ListOfInitTerms [, VarStep [, CeilingValue]]]) ⇒ liste

Genererer en liste med led for sekvensen $\text{depVar}(Var) = Expr$ 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 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.

Genererer de første fem led i sekvensen $u(n) = u(n-1)^2/2$ med $u(1) = 2$ og $VarStep = 1$.

$$\begin{array}{l} \text{seqGen}\left(\frac{u(n-1)^2}{n}, n, u, \{1, 5\}, \{2\}\right) \\ \left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\} \end{array}$$

Eksempel, hvor $Var0 = 2$:

$$\begin{array}{l} \text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right) \\ \left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\} \end{array}$$

Eksempel, hvor den startværdien er symbolsk:

$$\begin{array}{l} \text{seqGen}\left(u(n-1)+2, n, u, \{1, 5\}, \{a\}\right) \\ \{a, a+2, a+4, a+6, a+8\} \end{array}$$

Det oprindelige indhold af Var er uændret efter **seqGen()** er gennemført .

Standardværdien for $VarStep = 1$.

System med to sekvenser:

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u2(n-1)}{2} + u1(n-1)\right\}, n, \{u1, u2\}, \{1, 5\}, \left[\begin{array}{c} - \\ 2 \end{array}\right]\right)$$

$$\left[\begin{array}{ccccc} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{array}\right]$$

Bemærk: Void () i den oprindelige ledmatrix ovenfor bruges for at angive, at det oprindelige led for $u1(n)$ er beregnet vha. den eksplicite sekvensformel $u1(n) = 1/n$.

seqn(Expr(u , n [, ListOfInitTerms[, $nMax$ [, CeilingValue]]]) $\Rightarrow$ 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.

seqn(Expr(n [, $nMax$ [, CeilingValue]]) $\Rightarrow$ 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.

Hvis $nMax$ mangler, sættes $nMax$ til 2500

Hvis $nMax=0$, sættes $nMax$ til 2500

Bemærk: seqn() kalder seqGen() med $n0=1$ og $nstep=1$

Genererer de første fem led i sekvensen $u(n) = u(n-1)/2$ med $u(1)=2$.

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

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

series(Udtr1, Var, Orden [, Punkt])
 $\Rightarrow$ udtryk

series(Udtr1, Var, Orden [, Punkt]) |
Var > Punkt $\Rightarrow$ udtryk

series(Udtr1, Var, Orden [, Punkt]) |
Var < Punkt $\Rightarrow$ udtryk

Returnerer en generel afkortet repræsentation af en potensrække af *Udtr1* udviklet omkring *Punkt* til graden *Orden*. *Ordenen* kan være et rationelt tal. De resulterende potenser af (*Var* – *Punkt*) kan indeholde negative og/eller brøkeksponenter. Koefficienterne af disse potenser kan indeholde logaritmer af (*Var* – *Punkt*) og andre funktioner af *Var*, der er domineret af alle potenser af (*Var* – *Punkt*), der har det samme eksponentfortegn.

Punkt er som standard 0. *Punkt* kan være ∞ eller $-\infty$, i hvilke tilfælde udviklingen er gennem graden *Orden* i $1/(Var - Punkt)$.

series(...) returnerer “**series(...)**” hvis der ikke kan bestemmes en sådan repræsentation, som for eksempel for væsentlige singulariteter som $\sin(1/z)$ ved $z=0$, $e^{-1/z}$ ved $z=0$, eller e^z ved $z = \infty$ eller $-\infty$.

Hvis rækken eller en af dens afledte afbrydes af et spring ved *Punkt*, indeholder resultatet sandsynligvis deludtryk på formen $\text{sign}(\dots)$ eller $\text{abs}(\dots)$ for en reel udviklingsvariabel eller $(-1)^{\text{floor}(\dots \cdot \text{angle}(\dots))}$ for en kompleks udviklingsvariabel, der slutter med “_”. Hvis du kun vil bruge rækken til værdier på den ene side af *Punkt*, skal du tilføje den korrekte blandt funktionerne “| *Var* > *Punkt*”, “| *Var* < *Punkt*”, “| “*Var* $\geq$ *Punkt*”, eller “*Var* $\leq$ *Punkt*” for at opnå et enklere resultat.

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1\right) \quad \frac{1}{2} \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$$

$$\text{series}\left(\frac{-1}{e^z}, z, 1\right) \quad z - 1$$

$$\text{series}\left(\left(1 + \frac{1}{n}\right)^n, n, 2, \infty\right) \quad e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$$

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5\right) | x > 0 \quad \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$$

$$\text{series}\left(\int \frac{\sin(x)}{x} dx, x, 6\right) \quad x - \frac{x^3}{18} + \frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7\right) \quad \frac{x^3}{2} - \frac{x^5}{24} - \frac{29 \cdot x^7}{720}$$

$$\text{series}\left(\left(1 + e^x\right)^2, x, 2, 1\right) \quad (e+1)^2 + 2 \cdot e \cdot (e+1) \cdot (x-1) + e \cdot (2 \cdot e+1) \cdot (x-1)^2$$

series() kan give symbolske tilnærmelser til ubestemte og bestemte integraler, for hvilke, der ellers ikke kan opnås symbolske løsninger.

series() fordeler sig over lister og matricer med 1.-argument.

series() er en generaliseret version af **taylor()**.

Som vist med det sidste eksempel til højre kan visningsrutinerne flytte rundt på leddene i det resultat **taylor(...)** giver,, så det dominerende led ikke er det længst til venstre.

Bemærk: Se også **dominantTerm()**, side 58.

setMode()

setMode(tilstandNavnHeltal, indstilHeltal) $\Rightarrow$ heltal **setMode(liste) $\Rightarrow$ 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.

Viser en tilnærmet værdi af π med standardindstillingen for Viste cifre, og viser derefter π med en indstilling på Fix2. Sørg for, at standardindstillingen gendannes efter programmet er eksekveret.

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

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 86.

Bemærk: De aktuelle tilstandsindstillinger videresendes til kaldte underrutiner. Hvis en underrutine ændrer en tilstandsindstilling, mistes tilstandsændringen, når kontrollen returnerer til den kaldende rutine.

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

Tilstands- navn	Tilstands- heltal	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 kompleks	4	1=Reel, 2=Rektangulær, 3=Polær
Auto eller tilnærmet	5	1=Auto, 2=Tilnærmet, 3=Eksakt
Vektorformat	6	1=Rektangulær, 2=Cylindrisk, 3=Sfærisk
Talsystem	7	1=Decimal, 2=Hex, 3=Binær
Enhedssystem	8	1=SI, 2=Eng/US

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 18.

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 en flytning til højre droppes bittene længst mod højre, og 0 eller 1 indsættes for at matche bittene længst til venstre. I en flytning til venstre droppes bittene længst mod venstre, og 0 indsættes som bittene længst til højre.

For eksempel i en højreflytning:

Hver bit flytter til højre.

0b00000000000000111101011000011010

Indsætter 0, hvis bittene længst til venstre er 0,

eller 1, hvis bittene længst til venstre er 1.

giver:

00b00000000000000011110101100001101

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*.

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 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}

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(*StrengI* [,*antalFlytninger*])⇒*streng*

Returnerer en kopi af *StrengI* flyttet til højre eller venstre med *antalflytninger* tegn. Ændrer ikke *StrengI*.

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.

shift("abcd")	" abc "
shift("abcd",-2)	" ab "
shift("abcd",1)	"bcd "

sign()

sign(*Udtrl*)⇒*udtryk*

sign(*ListeI*)⇒*liste*

sign(*MatrixI*)⇒*matrix*

For reelle og komplekse *Udtrl*, returneres *Udtrl/abs(Udtrl)*, når *Udtrl*≠0.

Returnerer 1, hvis *Udtrl* er positivt.

Returnerer -1, hvis *Udtrl* 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.

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

Hvis kompleks formattilstand er real:

sign($\begin{bmatrix} -3 & 0 & 3 \end{bmatrix}$)	$\begin{bmatrix} -1 & \pm 1 & 1 \end{bmatrix}$
--	--

simult(koeffMatrix, konstVektor[, Tol])
 $\Rightarrow$ matrix

Returnerer en kolonnevektor, der indeholder løsningerne til et system af lineære ligninger.

Bemærk: Se også **linSolve()**, side 105.

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 matricelement 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:
 $5E-14 \cdot \max(\dim(koeffMatrix)) \cdot \text{rækkeNorm}(koeffMatrix)$

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 for x og y:

$$x + 2y = 1$$

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

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

Løsningen er $x=-3$ og $y=2$.

Løs:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{array}{l} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \text{matx1} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\ \text{simult}\left(\text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \quad \begin{bmatrix} -(2 \cdot b - d) \\ a \cdot d - b \cdot c \\ 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix} \end{array}$$

Løs:

$$x + 2y = 1$$

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

$$x + 2y = 2$$

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

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

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

Udtr sin

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

Repræsenterer *Udtr* i termer af sinus. Det er en konverteringsoperator for visning. Denne operator kan kun anvendes ved slutningen af indtastningslinjen.

sin reducerer alle potenser af

$$\cos(\dots) \text{ modulo } 1 - \sin(\dots)^2$$

således, at alle tilbageværende potenser af $\sin(\dots)$ har eksponenter i området (0, 2). Således vil resultatet være uden $\cos(\dots)$ hvis, og kun hvis $\cos(\dots)$ optræder i det givne udtryk, kun med lige potenser.

Bemærk: Konverteringsoperatorer understøttes ikke i vinkeltilstand, Grader eller Nygrader. Før brug skal man sikre at vinkeltilstanden er indstillet til radianer, og at *Udtr* ikke indeholder eksplicit reference til grader eller nygrader.

$$\left(\cos(x)\right)^2 \rightarrow \sin \quad 1 - \left(\sin(x)\right)^2$$

sin()



sin(Udtr I) ⇒ udtryk

I vinkeltilstanden Grader:

sin(Liste I) ⇒ liste

$$\sin\left(\frac{\pi_r}{4}\right) \quad \frac{\sqrt{2}}{2}$$

sin(Udtr I) returnerer sinus af argumentet som et udtryk.

$$\sin(45) \quad \frac{\sqrt{2}}{2}$$

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

$$\sin\left(\left\{0, 60, 90\right\}\right) \quad \left\{0, \frac{\sqrt{3}}{2}, 1\right\}$$

Bemærk: Argumentet fortolkes som en vinkel i enten grader eller radianer afhængigt af den aktuelle vinkeltilstand. Du kan bruge $^\circ$, G , eller r til midlertidigt at tilsidesætte den indstillede vinkeltilstand.

I vinkeltilstanden Nygrader:

$$\sin(50) \quad \frac{\sqrt{2}}{2}$$

I vinkeltilstanden Radian:

$$\sin\left(\frac{\pi}{4}\right) = \frac{\sqrt{2}}{2}$$

$$\sin(45^\circ) = \frac{\sqrt{2}}{2}$$

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.

I vinkeltilstanden Radian:

$$\sin\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

sin⁻¹(UdtrI)⇒udtryk

sin⁻¹(ListeI)⇒liste

sin⁻¹(UdtrI) returnerer den vinkel, hvis sinus er *UdtrI* som et udtryk.

sin⁻¹(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.

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

sin⁻¹(kvadratMatrixI)⇒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 Grader:

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

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

I vinkeltilstanden radian og tilstanden rektangulært komplekst format:

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

sinh()Katalog > **sinh**(*Udtr1*) $\Rightarrow$ *udtryk* $\sinh(1.2)$ 1.50946**sinh**(*Liste1*) $\Rightarrow$ *liste* $\sinh(\{0,1.2,3\})$ $\{0,1.50946,10.0179\}$ **sinh** (*Udtr1*) returnerer den hyperbolske sinus af argumentet som et udtryk.**sinh** (*Liste1*) returnerer en liste af de hyperbolske sinuser af hvert element af *Liste1*.**sinh**(*kvadratMatrix1*) $\Rightarrow$ *kvadratMatrix*

I vinkeltilstanden Radian:

Returnerer den matrixhyperbolske sinus af *kvadratMatrix1*. Dette er ikke det samme som at beregne den hyperbolske sinus for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.
$$\sinh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$
KvadratMatrix1 skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.**sinh⁻¹()**Katalog > **sinh⁻¹**(*Udtr1*) $\Rightarrow$ *udtryk* $\sinh^{-1}(0)$ 0**sinh⁻¹**(*Liste1*) $\Rightarrow$ *liste* $\sinh^{-1}(\{0,2.1,3\})$ $\{0,1.48748,\sinh^{-1}(3)\}$ **sinh⁻¹**(*Udtr1*) returnerer den inverse hyperbolske sinus af argumentet som et udtryk.**sinh⁻¹**(*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⁻¹**(*kvadratMatrix1*) $\Rightarrow$ *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 X, Y [, [*Iterationer*],[*Periode*] [, *Kategori*, *Medtag*]]

Beregner sinusregressionen på listerne X og Y . En sammenfatning af resultaterne lagres i *stat.results* variable. (side 181.)

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 kategorikoder 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 243.

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

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>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>

solve()

Katalog > 

solve(*Ligning*, *Var*) $\Rightarrow$ Boolesk udtryk

solve(*Ligning*, *Var*=*Gæt*) $\Rightarrow$ Boolesk udtryk

solve(*Ulighed*, *Var*) $\Rightarrow$ Boolesk udtryk

Returnerer reelle løsninger til en ligning eller ulighed for *Var*. Målet er at vise alle løsninger. Der kan dog være ligninger eller uligheder, for hvilke, antallet af løsninger er uendeligt.

Løsninger kan være ikke-reelle for visse kombinationer af værdier for udefinerede variable.

For Auto-indstillingen af

Auto eller tilnærmet-tilstanden er målet at producere eksakte løsninger, når de er concise, og suppleret af iterative søgninger med approksimerende aritmetik, når nøjagtige løsninger er upraktiske

På grund standardforkortelse med den største fælles divisor af tæller og nævner i brøker, kan løsninger kun være ensidige løsninger fra en eller to sider.

For uligheder af typerne $\geq$, $\leq$, $<$, eller $>$, er eksplícitte løsninger usandsynlige, medmindre uligheden er lineær og kun indeholder *Var*.

For Eksakt-indstillingen af

Auto eller tilnærmet-tilstanden returneres dele, der ikke kan løses som en implicit ligning eller ulighed.

$$\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x)$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a} \text{ or } x = \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a}$$

$$\text{Ans} | a=1 \text{ and } b=1 \text{ and } c=1$$

$$x = \frac{-1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ or } x = \frac{-1}{2} - \frac{\sqrt{3}}{2} \cdot i$$

$$\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x)$$

$$x=a \text{ or } x=0.567143$$

$$(x+1) \cdot \frac{x-1}{x-1} + x-3$$

$$2 \cdot x-2$$

$$\text{solve}(5 \cdot x-2 \geq 2 \cdot x, x)$$

$$x \geq \frac{2}{3}$$

$$\text{exact}(\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x))$$

$$e^x + x = 0 \text{ or } x=a$$

Brug operatoren ("|") til at afgrænse løsningsintervallet og/eller andre variable, der optræder i ligningen eller uligheden. Når du finder en løsning i et interval, kan du anvende ulighedsoperatorene til at udelukke dette interval fra efterfølgende søgninger.

False returneres, når der ikke bliver fundet nogen reelle løsninger. True returneres, hvis **solve()** kan bestemme, at enhver endelig reel værdi af *var* opfylder ligningen eller uligheden.

Da **solve()** altid returnerer et Boole'sk resultat, kan du anvende "and", "or", og "not" til at kombinere resultaterne fra **solve()** med hinanden eller med andre Boole'ske udtryk.

Løsningerne kan indeholde en unik, ny udefineret variabel af formen n_j , hvor j er et heltal i intervallet 1–255. Sådanne variable angiver et arbitrært heltal.

I Reel tilstand beregnes kun den reelle gren af brøkpotenser med ulige nævnere. Ved flertydige udtryk, som fx brøkpotenser, logaritmer og inverse trigonometriske funktioner, beregnes kun hovedgrenen. Derfor giver **solve()** kun løsninger, der svarer til den ene reelle gren eller hovedgrenen.

Bemærk: Se også **cSolve()**, **cZeros()**, **nSolve()** og **zeros()**.

solve(*Lign1* and *Lign2* [*and*...],
VarEllerGæt1, *VarEllerGæt2* [, ...])
⇒ *Boolesk* - udtryk

solve(*SystemAfLign*, *VarEllerGæt1*,
VarEllerGæt2 [, ...]) ⇒ *Boolesk* - udtryk

solve(*{Lign1, Lign2* [,...]} *{VarEllerGæt1*,
VarEllerGæt2 [, ...]) ⇒ *Boolesk* - udtryk

I vinkeltilstanden Radian:

$$\text{solve}\left(\tan(x)=\frac{1}{x}, x\right) | x>0 \text{ and } x<1$$

$$x=0.860334$$

$$\text{solve}(x=x+1, x) \quad \text{false}$$

$$\text{solve}(x=x, x) \quad \text{true}$$

$$2 \cdot x - 1 \leq 1 \text{ and } \text{solve}(x^2 \neq 9, x) \quad x \neq -3 \text{ and } x \leq 1$$

I vinkeltilstanden Radian:

$$\text{solve}(\sin(x)=0, x) \quad x=n1 \cdot \pi$$

$$\text{solve}\left(x^{\frac{1}{3}}=-1, x\right) \quad x=-1$$

$$\text{solve}(\sqrt{x}=2, x) \quad \text{false}$$

$$\text{solve}(-\sqrt{x}=2, x) \quad x=4$$

$$\text{solve}(y=x^2-2 \text{ and } x+2 \cdot y=-1, \{x, y\})$$

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

Returnerer mulige reelle løsninger til de sammenhørende algebraiske ligninger, hvor hvert *VarEllerGæt* angiver en variabel, du vil løse for.

Du kan adskille ligningerne med **and**-operatoren, eller du kan indtaste et *SystemAfLign* med en skabelon fra Katalog. Antallet af *VarEllerGæt*-argumenter skal svare til antallet af ligninger. Du kan også vælge at angive et initielt gæt til en variabel. Hvert *VarEllerGæt* skal have formen:

variabel

- eller -

variabel = reel eller ikke-reelt tal

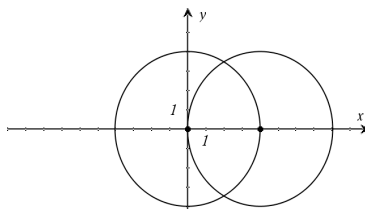
For eksempel er x gyldig, og det er $x=3$ også.

Hvis alle ligningerne er polynomier, og hvis du IKKE angiver nogle initielle gæt, benytter **solve()** den leksikale Gröbner/Buchberger eliminationsmetode som forsøg på at bestemme alle reelle løsninger.

Antag for eksempel, at du har en cirkel med radius r og centrum i origo og en anden cirkel med radius r og centrum, hvor den første cirkel skærer den positive x -akse. Anvend **solve()** til at finde skæringspunkterne.

Som illustreret ved r i eksemplet til højre kan sammenhørende polynomielle ligninger have ekstra variable, der ikke har nogen værdier, men repræsenterer givne numeriske værdier, der kan erstattes senere.

Du kan også (eller i stedet) medtage løsningsvariable, der ikke optræder i ligningerne. For eksempel kan du medtage z som en løsningsvariabel for at udvide det foregående eksempel til to parallelle skærende cylindre med radius r .



$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2}$$

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y=-\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1$$

Cylinderløsningerne illustrerer, hvordan familier af løsninger kan indeholde arbitrære konstanter af formen c_k , hvor k er et heltalssuffiks fra 1 til og med 255

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbrug stærkt af den rækkefølge løsningsvariablene angives i. Hvis det initielle valg kræver for meget hukommelse eller tålmodighed, skal du prøve at omarrangere variablene i ligningerne og/eller *varEllerGæt* listen.

Hvis du ikke medtager nogen gæt, og hvis en ligning er ikke-polynomiell i en variabel, men alle ligninger er lineære i løsningsvariablene, anvender **solve()** en Gauss-eliminering i et forsøg på at bestemme alle reelle løsninger.

Hvis et system hverken er polynomiell i alle variable eller lineært i sine løsningsvariable, bestemmer **solve()** højst en løsning med en iterativ approksimationsmetode. Dette gøres ved at lade antallet af løsningsvariable være lig med antallet af ligninger og reducere alle andre variable i ligningerne til tal.

Hver løsningsvariabel starter ved sin gættede værdi, hvis der findes en; Ellers starter den ved 0.0.

Anvend gæt til at søge flere løsninger, én ad gangen. For at opnå konvergens skal et gæt være meget tæt på en løsning.

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

$$\text{solve}\left(x + e^z \cdot y = 1 \text{ and } x - y = \sin(z), \{x, y\}\right)$$

$$x = \frac{e^z \cdot \sin(z) + 1}{e^z + 1} \text{ and } y = \frac{-(\sin(z) - 1)}{e^z + 1}$$

$$\text{solve}\left(e^z \cdot y = 1 \text{ and } -y = \sin(z), \{y, z\}\right)$$

$$y = 2.812e-10 \text{ and } z = 21.9911 \text{ or } y = 0.001871$$

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

$$\text{solve}\left(e^z \cdot y = 1 \text{ and } -y = \sin(z), \{y, z = 2 \cdot \pi\}\right)$$

$$y = 0.001871 \text{ and } z = 6.28131$$

SortA

Katalog >

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 243.

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 *list1*, *list2* Done

list1 $\{4,3,2,1\}$

list2 $\{3,4,1,2\}$

tomme (ugyldige) elementer i det første argument flyttes til bunden. Yderligere oplysninger om tomme elementer findes på side 243.

►Sphere

Vektor ►**Sphere**

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

Viser række- eller kolonnevektoren i sfærisk form [ρ $\angle\theta$ $\angle\phi$].

Vektor skal have dimensionen 3 og kan være enten en række- eller kolonnevektor.

Bemærk: ►**Sphere** er en displayformat-kommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje.

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på  .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

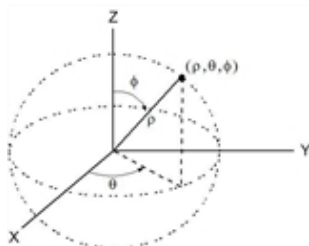
$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$ ►Sphere
 $\begin{bmatrix} 3.74166 & \angle 1.10715 & \angle 0.640522 \end{bmatrix}$

$\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix}$ ►Sphere
 $\begin{bmatrix} 3.60555 & \angle 0.785398 & \angle 0.588003 \end{bmatrix}$

Tryk .

$$\left(\left[2 \quad \angle \frac{\pi}{4} \quad 3 \right] \right) \text{Sphere}$$

$$\left[\sqrt{13} \quad \angle \frac{\pi}{4} \quad \angle \sin^{-1} \left(\frac{2 \cdot \sqrt{13}}{13} \right) \right]$$



sqrt()

sqrt(*Udtryk*) $\Rightarrow$ *udtryk*

$$\sqrt{4} \quad 2$$

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

$$\sqrt{\{9, a, 4\}} \quad \{3, \sqrt{a}, 2\}$$

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

Bemærk: Se også **Kvadratrodsskabelon**, side 1.

stat.results

Viser resultater fra en statistikberegning.

Resultaterne vises som en mængde af navn-værdi-par. De viste specifikke navne afhænger af den senest beregnede statistiske funktion eller kommando.

Du kan kopiere et navn eller en værdi og sætte den ind andre steder.

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.

$xlist = \{1, 2, 3, 4, 5\}$	$\{1, 2, 3, 4, 5\}$
$ylist = \{4, 8, 11, 14, 17\}$	$\{4, 8, 11, 14, 17\}$

LinRegMx $xlist, ylist, 1$: stat.results

"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r ² "	0.996109
"r"	0.998053
"Resid"	"{...}"

stat.values	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	"{-0.4, 0.4, 0.2, 0., -0.2}"

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBlock
stat.AdjR ²	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r ²	stat.SSY
stat.b1	stat.dflInteract	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.σ _x	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.σ _y	stat.tList
stat.b6	stat.e	stat.MSReg	stat.σ _{x1}	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.σ _{x2}	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.̄x
stat.b9	stat.FBlock	stat.̂p	stat.Σx ²	stat.̄x ₁
stat.b10	stat.Fcol	stat.̂p ₁	stat.Σxy	stat.̄x ₂
stat.bList	stat.FInteract	stat.̂p ₂	stat.Σy	stat.̄xDiff
stat.χ ²	stat.FreqReg	stat.̂pDiff	stat.Σy ²	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. $\bar{y}$
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat. $\hat{y}$
stat.CookDist	stat.MaxX	stat.PValRow	stat.SESlope	stat. $\hat{y}$ List
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

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()

Katalog >

`stDevPop(Liste[, Hyppighedsliste])`
 $\Rightarrow$ udtryk

Returnerer population standardafvigelsen af elementerne i *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster af tilsvarende elementer i *Liste*.

Bemærk: *Liste* skal have mindst to elementer. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 243

I vinkeltilstanden Radian og tilstanden Auto:

$$\text{stDevPop}\left(\left\{a, b, c\right\}\right) = \frac{\sqrt{2 \cdot \left(a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2\right)}}{3}$$

$$\text{stDevPop}\left(\left\{1, 2, 5, -6, 3, -2\right\}\right) = \frac{\sqrt{465}}{6}$$

$$\text{stDevPop}\left(\left\{\left\{1, 3, 2, 5, -6, 4\right\}, \left\{3, 2, 5\right\}\right\}\right) = 4.11107$$

stDevPop()Katalog > 

stDevPop(*Matrix1*[, *Hyppighedsmatrix*])
 $\Rightarrow$ 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 243.

$$\text{stDevPop}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right) \left[\begin{array}{ccc} 4 \cdot \sqrt{6} & \sqrt{78} & 2 \cdot \sqrt{6} \\ 3 & 3 & 3 \end{array} \right]$$

$$\text{stDevPop}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right) \left[\begin{array}{cc} 2.52608 & 5.21506 \end{array} \right]$$

stDevSamp()Katalog > 

stDevSamp(*Liste*[, *Hyppighedsliste*])
 $\Rightarrow$ 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 243

stDevSamp(*Matrix1*[, *Hyppighedsmatrix*])
 $\Rightarrow$ 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 243.

$$\text{stDevSamp}\left(\{a, b, c\}\right) \frac{\sqrt{3 \cdot \left(a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2\right)}}{3}$$

$$\text{stDevSamp}\left(\{1, 2, 5, -6, 3, -2\}\right) \frac{\sqrt{62}}{2}$$

$$\text{stDevSamp}\left(\{1.3, 2.5, -6.4\}, \{3, 2, 5\}\right) 4.33345$$

$$\text{stDevSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right) \left[\begin{array}{cc} 4 & \sqrt{13} & 2 \end{array} \right]$$

$$\text{stDevSamp}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right) \left[\begin{array}{cc} 2.7005 & 5.44695 \end{array} \right]$$

Stop	Katalog > 	
Stop	<i>i</i> :=0	0
Programmeringskommando: Afslutter programmet.	Define <i>prog1()</i> =Prgm	Done
	For <i>i</i> ,1,10,1	
	If <i>i</i> =5	
	Stop	
	EndFor	
	EndPrgm	
Stop er ikke tilladt i funktioner.	<i>prog1()</i>	Done
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>	5

Store	Se → (store), side 240.
-------	-------------------------

string()	Katalog > 	
string (<i>Udtr</i>)⇒ <i>streng</i>	string(1.2345)	"1.2345"
Reducerer <i>Udtr</i> og returnerer resultatet som en tegnstreng.	string(1+2)	"3"
	string(cos(<i>x</i>)+√3)	"cos(x)+√(3)"

subMat()	Katalog > 	
subMat (<i>Matrix1</i> [, <i>Startrække</i>] [, <i>Startkolonne</i>] [, <i>Slutrække</i>] [, <i>Slutkolonne</i>]) ⇒ <i>matrix</i>	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
Returnerer den angivne delmatrix af <i>Matrix1</i> .	subMat(<i>m1</i> ,2,1,3,2)	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
Standardindstillinger: <i>Startrække</i> =1, <i>Startkolonne</i> =1, <i>Slutrække</i> =sidste række, <i>Slutkolonne</i> =sidste kolonne.	subMat(<i>m1</i> ,2,2)	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)	Se G(), side 231.
-------------	-------------------

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 *Liste* ignoreres. Yderligere oplysninger om tomme elementer findes på side 243.

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 243.

$\text{sum}\left(\{1,2,3,4,5\}\right)$	15
$\text{sum}\left(\{a,2\cdot a,3\cdot a\}\right)$	$6\cdot a$
$\text{sum}\left(\text{seq}(n,n,1,10)\right)$	55
$\text{sum}\left(\{1,3,5,7,9\},3\right)$	21

$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},2,3\right)$	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

sumIf()

Katalog > 

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.

$\text{sumIf}\left(\{1,2,e,3,\pi,4,5,6\},2.5<?<4.5\right)$	$e+\pi+7$
$\text{sumIf}\left(\{1,2,3,4\},2<?<5,\{10,20,30,40\}\right)$	70

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 243.

Bemærk: Se også **countIf()**, side 36.

system(*Udtr1* [, *Udtr2* [, *Udtr3* [, ...]]])

$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right) \quad x=4 \text{ and } y=-4$$

system(*Ligning1* [, *Ligning2* [, *Ligning3* [, ...]]])

Returnerer et system af ligninger formateret som en liste. Du kan også oprette et system ved hjælp af en skabelon.

Bemærk: Se også **System af Ligninger**, side 3.

T

Matrix1^T ⇒ *matrix*

Returnerer den komplekse konjugerede transponerede *Matrix1*.

Bemærk: Du kan indsætte denne operator fra computerens tastatur ved at skrive @T.

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

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T \quad \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

$$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^T \quad \begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$$

tan()



tan(*Udtr1*)⇒*udtryk*

I vinkeltilstanden Grader:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45)$	1
$\tan(\{0,60,90\})$	$\{0,\sqrt{3},\text{undef}\}$

tan(*Liste1*)⇒*liste*

tan(*Udtr1*) returnerer tangens af argumentet som et udtryk.

tan(*Liste1*) returnerer en liste med tangens til alle elementer i *Liste1*.

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:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(50)$	1
$\tan(\{0,50,100\})$	$\{0,1,\text{undef}\}$

I vinkeltilstanden Radian:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45^\circ)$	1
$\tan\left(\left\{\pi, \frac{\pi}{3}, \pi, \frac{\pi}{4}\right\}\right)$	$\{0,\sqrt{3},0,1\}$

tan(*kvadratMatrix1*)⇒*kvadratMatrix*

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.

I vinkeltilstanden Radian:

$\tan\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$	$\begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$
--	--

tan⁻¹()



tan⁻¹(*Udtr1*)⇒*udtryk*

I vinkeltilstanden Grader:

$\tan^{-1}(1)$	45
----------------	----

tan⁻¹(*Liste1*)⇒*liste*

tan⁻¹(*Udtr1*) returnerer vinklen, hvis tangens er *Udtr1*, som et udtryk.

tan⁻¹(*Liste1*) returnerer en liste med den inverse tangens til hvert element af *Liste1*.

I vinkeltilstanden Nygrader:

$\tan^{-1}()$

 **-tast**

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 **arctan (...)**.

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

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

KvadratMatrixI skal være diagonaliserbar. Resultatet indeholder altid tal med flydende decimaler.

$\tan^{-1}(1)$ 50

I vinkeltilstanden Radian:

$\tan^{-1}(\{0,0,2,0,5\})$ $\{0,0.197396,0.463648\}$

I vinkeltilstanden Radian:

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

-0.083658	1.26629	0.62263
0.748539	0.630015	-0.070012
1.68608	-1.18244	0.455126

tangentLine()

Katalog > 

tangentLine(UdtrI,Var,Punkt) $\Rightarrow$ udtryk

tangentLine(UdtrI,Var,Punkt) $\Rightarrow$ udtryk

Returnerer tangentlinjen til kurven repræsenteret ved *UdtrI* i punktet angivet i *Var=Punkt*.

Vær sikker på, at den uafhængige variabel ikke er defineret. For eksempel, Hvis $f1(x) := 5$ og $x := 3$, så returnerer **tangentLine(f1(x),x,2)** "false."

$\text{tangentLine}(x^2, x, 1)$ $2 \cdot x - 1$

$\text{tangentLine}((x-3)^2 - 4, x=3)$ -4

$\text{tangentLine}\left(x^{\frac{1}{3}}, x=0\right)$ $x=0$

$\text{tangentLine}(\sqrt{x^2-4}, x=2)$ undef

$x:=3: \text{tangentLine}(x^2, x, 1)$ 5

tanh()

Katalog > 

tanh(UdtrI) $\Rightarrow$ udtryk

tanh(ListeI) $\Rightarrow$ liste

tanh(UdtrI) returnerer den hyperbolske tangens til argumentet som et udtryk.

tanh(ListeI) returnerer en liste med den hyperbolske tangens til hvert enkelt element i *ListeI*.

tanh(kvadratMatrixI) $\Rightarrow$ kvadratMatrix

$\tanh(1.2)$ 0.833655

$\tanh(\{0,1\})$ $\{0, \tanh(1)\}$

I vinkeltilstanden Radian:

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.

$$\tanh \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} = \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

tanh⁻¹(Udtr1)⇒udtryk

tanh⁻¹(Liste1)⇒liste

tanh⁻¹(Udtr1) returnerer den inverse hyperbolske tangens af argumentet som et udtryk.

tanh⁻¹(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⁻¹(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:

$$\begin{array}{l} \tanh^{-1}(0) \quad 0 \\ \tanh^{-1}(\{1, 2, 1, 3\}) \\ \left\{ \text{undef}, 0.518046 - 1.5708 \cdot i, \frac{\ln(2)}{2} - \frac{\pi}{2} \cdot i \right\} \end{array}$$

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

I vinkeltilstanden Radian og rektangulært komplekst format:

$$\tanh^{-1} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} = \begin{bmatrix} -0.099353 + 0.164058 \cdot i & 0.267834 - 1.4908 \\ -0.087596 - 0.725533 \cdot i & 0.479679 - 0.94730 \\ 0.511463 - 2.08316 \cdot i & -0.878563 + 1.7901 \end{bmatrix}$$

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

taylor()Katalog > 

taylor(*Udtr1*, *Var*, *Orden*[, *Punkt*])
 $\Rightarrow$ udtryk

Returnerer det ønskede Taylor-polynomium. Polynomiet medtager led, som er forskellige fra nul, og hvis grad er et helt tal fra nul til *Orden* i (*Var* minus *Punkt*). **taylor()** returnerer sig selv, hvis der ikke er nogen afkortet potensserie af denne orden, eller hvis den ville kræve negative eksponenter eller brøkeksponenter. Anvend substitutions- og/eller midlertidig multiplikation med en potens af (*Var* minus *Punkt*) for at bestemme mere generelle potensserier.

Punkt er som standard i nul og er ekspansionspunktet.

$$\begin{array}{l} \text{taylor}(e^{\sqrt{x}}, x, 2) \qquad \text{taylor}(e^{\sqrt{x}}, x, 2, 0) \\ \text{taylor}(e^t, t, 4) | t = \sqrt{x} \qquad \frac{3}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1 \\ \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right) \qquad \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right) \\ \text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}, x\right) \\ -x^3 - x^2 - x - \frac{1}{x} - 1 \end{array}$$

tCdf()Katalog > 

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 $P(X \leq \text{øvreGrænse})$, sæt *nedreGrænse* =
 $-\infty$.

tCollect()Katalog > 

tCollect(*Udtr1*) $\Rightarrow$ udtryk

Returnerer et udtryk hvor produkter og
 heltalspotenser af sinus- og cosinus
 konverteres til en linearkombination af
 sinus- og cosinus af flere vinkler,
 vinkelsummer og vinkeldifferenser.
 Transformation konverterer trigonometriske
 polynomier til en linearkombination af
 deres harmoniske svingninger.

$$\begin{array}{l} \text{tCollect}((\cos(\alpha))^2) \qquad \frac{\cos(2 \cdot \alpha) + 1}{2} \\ \text{tCollect}(\sin(\alpha) \cdot \cos(\beta)) \qquad \frac{\sin(\alpha - \beta) + \sin(\alpha + \beta)}{2} \end{array}$$

Undertiden kan **tCollect()** give det ønskede resultat, når den trigonometriske standardreduktion ikke gør det. **tCollect()** er tilbøjelig til at reversere transformationer udført af **tExpand()**. Nogen gange forenkler det et udtryk at anvende **tExpand()** på et resultat fra **tCollect()**, eller omvendt, i to separate trin.

tExpand

tExpand(*Udtryk*) $\Rightarrow$ *udtryk*

Returnerer et udtryk hvori sinus- og cosinus af multiple vinkler, vinkelsummer og vinkeldifferenser ekspanderes. På grund af identiteten $(\sin(x))^2 + (\cos(x))^2 = 1$ er der mange mulige ækvivalente resultater. Derfor kan et resultat adskille sig fra et resultat vist i andre publikationer.

Undertiden kan **tExpand()** give det ønskede resultat, når den trigonometriske standardreduktion ikke gør det. **tExpand()** er tilbøjelig til at reversere transformationer udført af **tCollect()**. Nogen gange forenkler det et udtryk at anvende **tCollect()** på et resultat fra **tExpand()**, eller omvendt, i to separate trin.

Bemærk: Skalering i gradtilstand med $\pi/180$ forstyrrer funktionen i **tExpand()**, der genkender ekspanderbare former. For at få de bedste resultater skal **tExpand()** anvendes i Radian-tilstand.

$$\begin{aligned} \text{tExpand}(\sin(3 \cdot \phi)) &= 4 \cdot \sin(\phi) \cdot (\cos(\phi))^2 - \sin(\phi) \\ \text{tExpand}(\cos(\alpha - \beta)) &= \cos(\alpha) \cdot \cos(\beta) + \sin(\alpha) \cdot \sin(\beta) \end{aligned}$$

Text

TextpromptStreng[, *DispFlag*]

Programmeringskommando: Standser programmet midlertidigt og viser tegnstrengen *promptStreng* i en dialogboks.

Når en bruger vælger **OK**, fortsætter programudførelsen.

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**.

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 154 eller **RequestStr**, side 154.

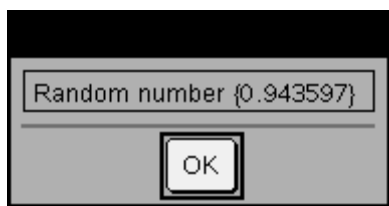
Bemærk: Du kan bruge denne kommando i et brugerdefineret program men ikke i en funktion.

```
Define text_demo()=Prgm
  For i,1,5
    strinfo:="Random number " & string
    (rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Kør programmet:

```
text_demo()
```

Eksempel på en dialogboks:



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 181.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

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. σ_x	Stikprøve standardafvigelse
stat.n	Længde på datasekvensen med målingsgennemsnit

tInterval_2Samp

Katalog > 

tInterval_2Samp *Liste1, Liste2*
 $[, Hyppighed1[, Hyppighed2[, CNiveau$
 $[, Puljet]]]$

(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 181.)

Puljet=1 puljer varianser. *Puljet=0* puljer ikke varianser.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
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 standardafvigelser for <i>Liste 1</i> og <i>Liste 2</i>

Output-variabel	Beskrivelse
stat.n1, stat.n2	Antal målinger i datasekvenserne
stat.sp	Den puljede standardafvigelse. Beregnes, når <i>Puljet</i> = 1.

tmpCnv()

Katalog >

tmpCnv(*Udtr* °*tempEnhed*, _
°*tempEnhed2*) ⇒ *udtryk* °*tempEnhed2*

Konverterer en temperaturværdi med *Udtr* mellem enhederne. De gyldige enheder for temperatur er:

_ °C Celsius

_ °F Fahrenheit

_ °K Kelvin

_ °R Rankine

Skriv ° ved at vælge den blandt Katalogsymboler.

Skriv _ ved at trykke på .

For eksempel konverteres 100_ °C til 212_ °F.

Et temperaturområde konverteres i stedet med **ΔtmpCnv**().

tmpCnv(100_ °C, _ °F)	212_ °F
tmpCnv(32_ °F, _ °C)	0_ °C
tmpCnv(0_ °C, _ °K)	273.15_ °K
tmpCnv(0_ °F, _ °R)	459.67_ °R

Bemærk: Med Katalog kan du vælge temperaturenheder.

ΔtmpCnv()

Katalog >

ΔtmpCnv(*Udtryk* °*tempEnhed*, _
°*tempEnhed2*) ⇒ *udtryk* °*tempEnhed2*

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

Konverterer et temperaturområde (forskellen mellem to temperaturværdier) angivet med *Udtryk* fra en enhed til en anden. De gyldige enheder for temperatur er:

_ °C Celsius

Skriv Δ ved at vælge den blandt Katalogsymboler.

ΔtmpCnv(100_ °C, _ °F)	180_ °F
ΔtmpCnv(180_ °F, _ °C)	100_ °C
ΔtmpCnv(100_ °C, _ °K)	100_ °K
ΔtmpCnv(100_ °F, _ °R)	100_ °R
ΔtmpCnv(1_ °C, _ °F)	1.8_ °F

Bemærk: Med Katalog kan du vælge temperaturenheder.

Δ tmpCnv()

Katalog > 

_ °F Fahrenheit

_ °K Kelvin

_ °R Rankine

Du kan skrive ° ved at vælge det på symbolpaletten eller skrive @d.

Skriv _ ved at trykke på  .

1_ °C og 1_ °K har samme størrelse som 1_ °F og 1_ °R. 1_ °C er dog 9/5 større end 1_ °F.

For eksempel er et 100_ °C område (fra 0_ °C til 100_ °C) ækvivalent med et område 180_ °F-område.

For at konvertere en bestemt temperaturværdi i stedet for et område skal du anvende **tmpCnv()**.

tPdf()

Katalog > 

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()

Katalog > 

trace(kvadratMatrix)⇒*udtryk*

Returner sporet (sum af alle elementer på hoveddiagonalen) af *kvadratMatrix*.

$\text{trace}\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	15
$\text{trace}\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}$	$2 \cdot a$

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 250.

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.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

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

Bemærk: Se også **ClrErr**, side 26, og **PassErr**, side 135.

```
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** μ_0 ,*Liste*[,*Hyppighed*[,*Hypot*]]

(Datalisteinput)

tTest μ_0 , $\bar{x}$,*sx*,*n*,[*Hypot*]

(Sammenfatning, stat input)

Udfører en hypotesetest for et enkelt ubekendt populationsgennemsnit μ , når populationens standardafvigelse er ubekendt. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 181.)

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 243.

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*
[,*Hyppighed2*[,*Hypot*[,*Puljet*]]]]

(Datalisteinput)

tTest_2Samp $\bar{x}_1, s_{x1}, n_1, \bar{x}_2, s_{x2}, n_2$, [Hypot
[,Puljet]]

(Sammenfatning, stat input)

Beregner en to-prøvers t test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 181.)

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 243.

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.s $_{x1}$, stat.s $_{x2}$	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*])
⇒ 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 200. Se også **amortTbl()**, side 8.

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 200. Se også **amortTbl()**, side 8.

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 200. Se også **amortTbl()**, side 8.

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 200. Se også **amortTbl()**, side 8.

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 200. Se også **amortTbl()**, side 8.

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

TwoVar *X, Y, [Frekv] [, Kategori, Medtag]*

Beregner statistik med to variable. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 181.)

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 ≥ 0 .

Kategori er en liste, der indeholder kategorikoder 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 243.

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 ²	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
stat.MinX	Minimum af x-værdier

Output-variabel	Beskrivelse
stat.Q ₁ X	1. kvartil af x
stat.MedianX	Median af x
stat.Q ₃ X	3. kvartil af x
stat.MaxX	Maksimum af x-værdier
stat.MinY	Minimum af y-værdier
stat.Q ₁ Y	1. kvartil af y
stat.MedY	Median af y
stat.Q ₃ Y	3. kvartil af y
stat.MaxY	Maksimum af y-værdier
stat. (x-) ²	Summen af kvadraterne på afvigelser fra middelværdien for x
stat. (y-) ²	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.

$$\text{unitV}\left(\begin{bmatrix} a & b & c \end{bmatrix}\right) = \begin{bmatrix} \frac{a}{\sqrt{a^2+b^2+c^2}} & \frac{b}{\sqrt{a^2+b^2+c^2}} & \frac{c}{\sqrt{a^2+b^2+c^2}} \end{bmatrix}$$

$$\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{3} & \frac{\sqrt{6}}{6} \end{bmatrix}$$

$$\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{14}}{14} \\ \frac{2\sqrt{14}}{14} \\ \frac{3\sqrt{14}}{14} \end{bmatrix}$$

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

unLock	Catalog > 	
unLock <i>Var1</i> [, <i>Var2</i>] [, <i>Var3</i>] ...	<i>a</i> :=65	65
unLock <i>Var</i> .	Lock <i>a</i>	Done
Oplåser de angivne variable eller variabelgrupper. Låste variable kan ikke redigeres eller slettes.	getLockInfo(<i>a</i>)	1
Se Lock , side 109 og getLockInfo() , side 86.	<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

V

varPop()	Katalog > 	
varPop (<i>Liste</i> [, <i>hyppighedsliste</i>])⇒ <i>udtryk</i>	varPop({ 5,10,15,20,25,30 })	875
Returnerer populationsvariansen af <i>Liste</i> .		12
Hvert <i>hyppighedsliste</i> -element tæller antallet af konsekutive forekomster de tilsvarende elementer i <i>Liste</i> .	Ans: 1.	72.9167
Bemærk: <i>Liste</i> 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 243.		

varSamp()	Katalog > 	
varSamp (<i>Liste</i> [, <i>Hyppighedsliste</i>])⇒ <i>udtryk</i>	varSamp({ { <i>a</i> , <i>b</i> , <i>c</i> } })	
Returnerer stikprøvevariansen for <i>Liste</i> .	$\frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$	
Hvert <i>hyppighedsliste</i> -element tæller antallet af konsekutive forekomster de tilsvarende elementer i <i>Liste</i> .	varSamp({ { 1,2,5, 6,3, -2 } })	31
		2
	varSamp({ { 1,3,5 }, { 4,6,2 } })	68
		33
Bemærk: <i>Liste</i> 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 243.

varSamp(*Matrix1*[, *Hyppighedsmatrix*])
 $\Rightarrow$ *matrix*

Returnerer en rækkevektor med stikprøvevariansen for hver kolonne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

Bemærk: *Matrix1* 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 243.

varSamp	$\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{pmatrix}$	$\begin{bmatrix} 4.75 & 1.03 & 4 \end{bmatrix}$
varSamp	$\begin{pmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{pmatrix}, \begin{pmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{pmatrix}$	$\begin{bmatrix} 3.91731 & 2.08411 \end{bmatrix}$

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. Kommandoen 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

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 variablen *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"

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: kan du bruge kommandoen **Wait** i et brugerdefineret program, men ikke i en funktion.

warnCodes ()

warnCodes(*Expr1*, *StatusVar*) $\Rightarrow$ udtryk

Beregner udtrykket *Expr1*, returnerer resultater 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 258 for at få en oversigt over fejlkoder og tilhørende meddelelser.

	$\text{warnCodes}\left(\text{solve}\left(\sin\left(10 \cdot x\right)=\frac{x^2}{x}, x\right), \text{warn}\right)$
	$x=-0.84232 \text{ or } x=-0.706817 \text{ or } x=-0.2852$
<i>warn</i>	$\{10007, 10009\}$

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

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.

$\text{when}(x < 0, x + 3)$	$x = 5$	undef
-----------------------------	---------	-------

when()

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

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 tempsum	
EndFunc	
	Done
$\text{sum_of_recip}(3)$	$\frac{11}{6}$

X

xor

BoolskUdtryk1 **xor** *BoolskUdtryk2*
returnerer *Boolsk udtryk*

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 18.

Bemærk: Se **or**, side 133.

Heltal1 xor Heltal2 $\Rightarrow$ 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 talssystem).

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 133.

I hexadecimal tilstand:

Vigtigt: Tallet nul, ikke bogstavet O.

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

zeros()

zeros(Udtr, Var) $\Rightarrow$ liste

zeros(Udtr, Var=Gæt) $\Rightarrow$ liste

Returnerer en liste med reelle løsninger for *Var* som gør *Udtr*=0. **zeros()** gør dette ved beregning **explist(solve(Udtr=0, Var), Var)**.

$$\text{zeros}(a \cdot x^2 + b \cdot x + c, x) = \left\{ \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}, \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a} \right\}$$

$$a \cdot x^2 + b \cdot x + c | x = \text{Ans}[2] \quad 0$$

Til visse formål er resultatformatet for **zeros()** mere hensigtsmæssigt end det for **solve()**. Men resultatformatet af **zeros()** kan ikke udtrykke implicitte løsninger, løsninger, der kræver uligheder, eller løsninger, der ikke omfatter *Var*.

Bemærk: Se også **cSolve()**, **cZeros()** og **solve()**.

zeros({*Udtr1*, *Udtr2*}, {*VarEllerGæt1*, *VarEllerGæt2* [, ...]}) $\Rightarrow$ *matrix*

Returnerer de mulige reelle nulpunkter for de sammenhørende algebraiske udtryk, hvor hvert *VarEllerGæt* angiver en ubekendt, du søger værdien for.

Du kan også vælge at angive et initielt gæt til en variabel. Hvert *VarEllerGæt* skal have formen:

variabel

– eller –

variabel = *reelt eller ikke-reelt tal*

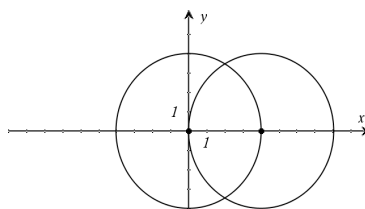
For eksempel er *x* gyldig, og det er *x=3* også.

Hvis alle ligningerne er polynomier, og hvis du IKKE angiver nogle initielle gæt, benytter **zeros()** den leksikale Gröbner/Buchberger eliminationsmetode som forsøg på at bestemme alle komplekse nulpunkter.

Antag for eksempel, at du har en cirkel med radius *r* og centrum i origo og en anden cirkel med radius *r* og centrum, hvor den første cirkel skærer den positive *x*-akse. Anvend **zeros()** til at finde skæringspunkterne.

Som illustreret ved *r* i eksemplet til højre kan sammenhørende polynomielle udtryk have ekstra variable, der ikke har nogen værdier, men repræsenterer givne numeriske værdier, der kan erstattes senere.

$$\begin{array}{l} \text{exact}\left(\text{zeros}\left(a \cdot (e^x + x) \cdot (\text{sign}(x) - 1), x\right)\right) \quad \{\emptyset\} \\ \text{exact}\left(\text{solve}\left(a \cdot (e^x + x) \cdot (\text{sign}(x) - 1) = 0, x\right)\right) \\ e^x + x = 0 \text{ or } x > 0 \text{ or } a = 0 \end{array}$$



$$\text{zeros}\left(\left\{x^2 + y^2 - r^2, (x-r)^2 + y^2 - r^2\right\}, \{x, y\}\right) \begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \\ \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \end{bmatrix}$$

Hver række i den resulterende matrix repræsenterer et alternativt nulunkt med komponenterne arrangeret på samme måde som *varEllerGæt*-listen. Du kan udtrække en række ved at indeksere matricen efter *[række]*.

Du kan også (eller i stedet) medtage ubekendte, der ikke optræder i udtrykkene. For eksempel kan du medtage *z* som en ubekendt for at udvide det foregående eksempel til to parallelle skærende cylindre med radius *r*. Cylindrenes nul illustrerer, hvordan familier af nulpunkter kan indeholde arbitrære konstanter af formen @*k*, hvor *k* er et heltalssuffiks fra 1 til og med 255.

For polynomielle systemer afhænger beregningstiden eller hukommelsesforbruget stærkt af den rækkefølge, de ubekendte angives i. Hvis det initiale valg kræver for meget hukommelse eller tålmodighed, skal du prøve at omarrangere variablene i udtrykkene og/eller *varEllerGæt* listen.

Hvis du ikke medtager nogen gæt, og hvis en ligning er ikke-polynomiel i en variabel, men alle udtryk er lineære i alle ubekendte, anvender **zeros()** en Gauss-eliminering i et forsøg på at bestemme alle nulpunkter.

Hvis et system hverken er polynomielt i alle variable eller lineært i sine ubekendte, bestemmer **zeros()** højst et nul med en iterativ approksimationsmetode. Dette gøres ved at lade antallet af ubekendte være lig med antallet af udtryk og reducere alle andre variable i udtrykkene til tal.

Hver ubekendte starter ved sin gættede værdi, hvis der findes en. Ellers starter den ved 0.0.

Anvend gæt til at søge flere nulpunkter, ét ad gangen. For at opnå konvergens skal et gæt være meget tæt på et nulpunkt.

Udtræk række 2:

$$\text{Ans}[2] \quad \left[\begin{array}{cc} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \end{array} \right]$$

$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x, y, z\}\right) \quad \left[\begin{array}{cc} \frac{r}{2} & \frac{-\sqrt{3} \cdot r}{2} & c1 \\ \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} & c1 \end{array} \right]$$

$$\text{zeros}\left(\left\{x+e^z \cdot y-1, x-y-\sin(z)\right\}, \{x, y\}\right) \quad \left[\begin{array}{cc} \frac{e^z \cdot \sin(z)+1}{e^z+1} & \frac{-(\sin(z)-1)}{e^z+1} \end{array} \right]$$

$$\text{zeros}\left(\left\{e^z \cdot y-1, y-\sin(z)\right\}, \{y, z\}\right) \quad \left[\begin{array}{cc} 0.041458 & 3.18306 \\ 0.001871 & 6.28131 \\ 4.76\text{E-}11 & 1796.99 \\ 2.\text{E-}13 & 254.469 \end{array} \right]$$

$$\text{zeros}\left(\left\{e^z \cdot y-1, y-\sin(z)\right\}, \{y, z=2 \cdot \pi\}\right) \quad \left[\begin{array}{cc} 0.001871 & 6.28131 \end{array} \right]$$

zInterval $\sigma, \text{Liste}, [\text{Hyppighed}, \text{CNiveau}]$

(Datalisteinput)

zInterval $\sigma, \bar{x}, n, [\text{CNiveau}]$

(Sammenfatning, stat input)

Beregner et z konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 181.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

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. σ	Kendt populationsstandardafvigelse for datasekvensen <i>Liste</i>

zInterval_1Prop

zInterval_1Prop $x, n, [\text{CNiveau}]$

Beregner én-proportion z konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 181.)

x er et ikke-negativt heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
stat. $\hat{p}$	Den beregnede brøkdel af succeser
stat.ME	Fejlmargen
stat.n	Antal stikprøver i datasekvens

zInterval_2Prop $x1, n1, x2, n2, [CNiveau]$

Beregner et to-proportion z konfidensinterval. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 181.)

$x1$ og $x2$ er ikke-negative heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

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 $\sigma_1, \sigma_2, Liste1, Liste2$
 $[, Hyppighed1[, Hyppighed2[, CNiveau]]]$

(Datalisteinput)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2$
 $[, CNiveau]$

(Sammenfatning, stat input)

Beregner et z konfidensinterval med to målinger. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 181.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
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. σ_1 , stat. σ_2	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 $\mu_0, \sigma, \text{Liste}, [\text{Hyppighed}], [\text{Hypot}]$

(Datalisteinput)

zTest $\mu_0, \sigma, \bar{x}, n, [\text{Hypot}]$

(Sammenfatning, stat input)

Udfører en z test med hyppigheden *Hyppighedsliste*. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 181.)

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 243.

Output-variabel	Beskrivelse
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$
stat.P Værdi	Mindste sandsynlighed, ved hvilken nul-hypotesen kan forkastes
stat. $\bar{x}$	Middelværdi af stikprøver af datasekvensen i <i>Liste</i>

Output-variabel	Beskrivelse
stat.sx	Standardmåleafvigelse for datasekvensen. Returneres kun for <i>Datainput</i> .
stat.n	Stikprøvens størrelse

zTest_1Prop

Katalog > 

zTest_1Prop $p0, x, n[, Hypot]$

Beregner en én-proportion z test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 181.)

x er et ikke-negativt heltal.

Test $H_0: p = p0$ mod en af følgende:

Til $H_a: p > p0$, sæt *Hypot*>0

til $H_a: p \neq p0$ (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 243.

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 $x1, n1, x2, n2[, Hypot]$

Beregner en toproportional z test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 181.)

$x1$ og $x2$ er ikke-negative heltal.

Test $H_0: p1 = p2$ mod en af følgende:

Til $H_a: p1 > p2$, 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 243.

Output-variabel	Beskrivelse
stat.z	Standardnormalværdi beregnet som differens af proportioner
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat. $\hat{p}1$	Første stikprøves proportionsestimat
stat. $\hat{p}2$	Anden stikprøves proportionsestimat
stat. $\hat{p}$	Puljet stikprøves proportionsestimat
stat.n1, stat.n2	Antal stikprøver taget i forsøg 1 og 2

zTest_2Samp $\sigma_1, \sigma_2, Liste1, Liste2$
 $[,Hyppighed1[,Hyppighed2[,Hypot]]]$

(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 variablen *stat.results*. (side 181.)

Test $H_0: \mu1 = \mu2$, mod af de følgende:

Til $H_a: \mu1 < \mu2$, sæt $Hypot<0$

til $H_a: \mu1 \neq \mu2$ (standard), sæt $Hypot=0$

til $H_a: \mu1 > \mu2$, $Hypot>0$

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 243.

Output-variabel	Beskrivelse
stat.z	Standardnormalværdi beregnet som differens af gennemsnit

Output-variabel	Beskrivelse
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
$Udtr1 + Udtr2 \Rightarrow udtryk$	56	56
Returnerer summen af de to argumenter.	$56+4$	60
	$60+4$	64
	$64+4$	68
	$68+4$	72
$Liste1 + Liste2 \Rightarrow liste$	$\left\{ 22, \pi, \frac{\pi}{2} \right\} \rightarrow I1$	$\left\{ 22, \pi, \frac{\pi}{2} \right\}$
$Matrix1 + Matrix2 \Rightarrow matrix$	$\left\{ 10, 5, \frac{\pi}{2} \right\} \rightarrow I2$	$\left\{ 10, 5, \frac{\pi}{2} \right\}$
Returnerer en liste (eller matrix), der indeholder summerne af tilsvarende elementer i <i>Liste1</i> og <i>Liste2</i> (eller <i>Matrix1</i> og <i>Matrix2</i>).	$I1+I2$	$\{ 32, \pi+5, \pi \}$
	$Ans+\{ \pi, 5, \pi \}$	$\{ \pi+32, \pi, 0 \}$
Argumenternes dimensioner må være ens.	$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$
$Udtr + Liste1 \Rightarrow liste$	$15+\{ 10, 15, 20 \}$	$\{ 25, 30, 35 \}$
$Liste1 + Udtr \Rightarrow liste$	$\{ 10, 15, 20 \}+15$	$\{ 25, 30, 35 \}$
Returnerer en liste med summerne af <i>Udtr</i> og hvert element i <i>Liste1</i> .		
$Udtr + Matrix1 \Rightarrow matrix$	$20+\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
Returnerer en matrix med <i>Udtr</i> føjet til hvert element på diagonalen af <i>Matrix1</i> . <i>Matrix1</i> skal være kvadratisk.		
Bemærk: Anvend .+ (punktum plus) til at addere et udtryk til hvert element.		

- (subtraher)		- -tast
$Udtr1-Udtr2 \Rightarrow udtryk$	$6-2$	4
Returnerer <i>Udtr1</i> minus <i>Udtr2</i> .	$\pi - \frac{\pi}{6}$	$\frac{5 \cdot \pi}{6}$
$Liste1-Liste2 \Rightarrow liste$	$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\}$	$\{ 12, \pi-5, 0 \}$
$Matrix1-Matrix2 \Rightarrow matrix$	$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 2 \end{bmatrix}$

-(subtraher)**-tast**

Subtraherer hvert element i *Liste2* (eller *Matrix1*) fra det tilsvarende element i *Liste1* (eller *Matrix1*), og returnerer resultaterne.

Argumenternes dimensioner må være ens.

Udtr-Liste1⇒*liste*

$$\begin{array}{r} 15 - \{10, 15, 20\} \qquad \qquad \{5, 0, -5\} \\ \{10, 15, 20\} - 15 \qquad \qquad \{ -5, 0, 5 \} \end{array}$$

Liste1-Udtr⇒*liste*

Subtraherer hvert *Liste1* element fra *Udtr* eller subtraherer *Udtr* fra hvert *Liste1* element og returnerer en liste med resultaterne.

Udtr - *Matrix1*⇒*matrix*

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad \qquad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

Matrix1-Udtr⇒*matrix*

Udtr - *Matrix1* returnerer en matrix med *Udtr* gange identitetsmatricen minus *Matrix1*. *Matrix1* skal være kvadratisk.

Matrix1 - *Udtr* returnerer en matrix med *Udtr* gange identitetsmatricen subtraheret fra *Matrix1*. *Matrix1* skal være kvadratisk.

Bemærk: Anvend .- (punktum minus) til at subtrahere et udtryk fra hvert element.

·(multiplicer)**×-tast**

Udtr1 · *Udtr2*⇒*udtryk*

$$2 \cdot 3,45 \qquad \qquad 6,9$$

Returnerer produktet af de to argumenter.

$$x \cdot y \cdot x \qquad \qquad x^2 \cdot y$$

Liste1 · *Liste2*⇒*liste*

$$\{1, 2, 3\} \cdot \{4, 5, 6\} \qquad \qquad \{4, 10, 18\}$$

Returnerer en liste med produkterne af de tilsvarende elementer i *Liste1* og *Liste2*.

$$\begin{bmatrix} 2 & 3 \\ a & 2 \end{bmatrix} \cdot \begin{bmatrix} a^2 & b \\ c & 3 \end{bmatrix} \qquad \qquad \begin{bmatrix} 2 \cdot a & b \\ 2 \cdot a & 2 \end{bmatrix}$$

Listernes dimensioner må være ens.

Matrix1 · *Matrix2*⇒*matrix*

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} \qquad \qquad \begin{bmatrix} a+2 \cdot b+3 \cdot c & d+2 \cdot e+3 \cdot f \\ 4 \cdot a+5 \cdot b+6 \cdot c & 4 \cdot d+5 \cdot e+6 \cdot f \end{bmatrix}$$

Returnerer matrixproduktet af *Matrix1* og *Matrix2*.

Antallet af kolonner i *Matrix1* skal være lig med antallet af rækker i *Matrix2*.

· (multiplicer)

 -tast

$Udtr \cdot Liste1 \Rightarrow liste$

$$\pi \cdot \{4,5,6\}$$

$$\{4 \cdot \pi, 5 \cdot \pi, 6 \cdot \pi\}$$

$Liste1 \cdot Udtr \Rightarrow liste$

Returnerer en liste med produkterne af $Udtr$ og hvert element i $Liste1$.

$Udtr \cdot Matrix1 \Rightarrow matrix$

$Matrix1 \cdot Udtr \Rightarrow matrix$

Returnerer en matrix med produkterne af $Udtr$ og hvert element i $Matrix1$.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01$$

$$\begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$\lambda \cdot \text{identity}(3)$$

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

Bemærk: Anvend $\cdot$ (punktum gangetegn) til at gange et udtryk med hvert element.

/ (divider)

 -tast

$Udtr1 / Udtr2 \Rightarrow udtryk$

Returnerer kvotienten af $Udtr1$ divideret med $Udtr2$.

Bemærk: Se også **Brøkskabelon**, side 1.

$Liste1 / Liste2 \Rightarrow liste$

Returnerer en liste med kvotienterne af $Liste1$ divideret med $Liste2$.

Listernes dimensioner må være ens.

$Udtr / Liste1 \Rightarrow liste$

$Liste1 / Udtr \Rightarrow liste$

Returnerer en liste med kvotienterne af $Udtr$ divideret med $Liste1$ eller $Liste1$ divideret med $Udtr$.

$Matrix1 / Udtr \Rightarrow matrix$

Returnerer en matrix med kvotienterne af $Matrix1 / Udtr$.

$$\frac{2}{3.45} \quad .57971$$

$$\frac{x^3}{x} \quad x^2$$

$$\frac{\{1,2,3\}}{\{4,5,6\}} \quad \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

$$\frac{a}{\{3,a,\sqrt{a}\}} \quad \left\{\frac{a}{3}, 1, \sqrt{a}\right\}$$

$$\frac{\{a,b,c\}}{a \cdot b \cdot c} \quad \left\{\frac{1}{b \cdot c}, \frac{1}{a \cdot c}, \frac{1}{a \cdot b}\right\}$$

$$\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{a \cdot b \cdot c} \quad \begin{bmatrix} \frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b} \end{bmatrix}$$

Bemærk: Anvend $/$ (punktum divisionstegn) til at dividere et udtryk med hvert element.

$Udtr1 \wedge Udtr2 \Rightarrow udtryk$

$$4^2 \qquad 16$$

$Liste1 \wedge Liste2 \Rightarrow liste$

$$\{a, 2, c\}^{\{1, b, 3\}} \qquad \{a, 2^b, c^3\}$$

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.

$Udtr \wedge Liste1 \Rightarrow liste$

$$p^{\{a, 2, -3\}} \qquad \left\{p^a, p^2, \frac{1}{p^3}\right\}$$

Returnerer *Udtr* opløftet til potensen af elementerne i *Liste1*.

$Liste1 \wedge Udtr \Rightarrow liste$

$$\{1, 2, 3, 4\}^{-2} \qquad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

Returnerer elementerne i *Liste1* opløftet til potensen i *Udtr*.

$kvadratMatrix1 \wedge heltal \Rightarrow matrix$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \qquad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

Returnerer *kvadratMatrix1* opløftet til *heltal*spotensen.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \qquad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

kvadratMatrix1 skal være en kvadratisk 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.

Hvis *heltal* < -1, beregnes den inverse matrix til en passende positiv potens.

x^2 (kvadrat) **x^2 -tast** $Udtr1^2 \Rightarrow udtryk$

4^2	16
-------	----

Returnerer kvadratet af argumentet.

$\{2,4,6\}^2$	$\{4,16,36\}$
---------------	---------------

 $Liste1^2 \Rightarrow liste$

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$	$\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$
---	--

Returnerer en liste med kvadraterne på elementerne i $Liste1$. $kvadratMatrix1^2 \Rightarrow matrix$

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^{\wedge 2}$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$
--	--

Returnerer matrix i anden potens af $kvadratMatrix1$. Dette er ikke det samme som at beregne kvadratet på hvert element. Brug $\wedge 2$ til at beregne kvadratet på hvert element.

 $+$ (punktum plustegn) **$\cdot +$ -taster** $Matrix1 \cdot + Matrix2 \Rightarrow matrix$

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$
---	--

 $Udtr \cdot + Matrix1 \Rightarrow matrix$

$x \cdot + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$
--	--

$Matrix1 \cdot + Matrix2$ returnerer en matrix, der er summen af hvert par tilsvarende elementer i $Matrix1$ og $Matrix2$.

$Udtr \cdot + Matrix1$ returnerer en matrix, der er summen af $Udtr$ og hvert element i $Matrix1$.

 $-$ (punktum minus.) **$\cdot -$ -taster** $Matrix1 \cdot - Matrix2 \Rightarrow matrix$

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$
---	--

 $Udtr \cdot - Matrix1 \Rightarrow matrix$

$x \cdot - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$
--	--

$Matrix1 \cdot - Matrix2$ returnerer en matrix, der er differensen mellem hvert par af tilsvarende elementer i $Matrix1$ og $Matrix2$.

$Udtr \cdot - Matrix1$ returnerer en matrix, der er differensen af $Udtr$ og hvert element i $Matrix1$.

· (punktum mult.)

· × -taster

$Matrix1 \cdot Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$$

$Udtr \cdot Matrix1 \Rightarrow matrix$

$$x \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$$

$Matrix1 \cdot Matrix2$ returnerer en matrix, der er produktet af hvert par tilsvarende elementer i $Matrix1$ og $Matrix2$.

$Udtr \cdot Matrix1$ returnerer en matrix med produkterne af $Udtr$ og hvert element i $Matrix1$.

/ (punktum divider)

/ ÷ -taster

$Matrix1 / Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} / \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} \frac{a}{c} & \frac{1}{2} \\ \frac{b}{5} & \frac{3}{d} \end{bmatrix}$$

$Udtr / Matrix1 \Rightarrow matrix$

$$x / \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$$

$Matrix1 / Matrix2$ returnerer en matrix, der er kvotient af hvert par af tilsvarende elementer i $Matrix1$ og $Matrix2$.

$Udtr / Matrix1$ returnerer en matrix, der er kvotienten af $Udtr$ og hvert element i $Matrix1$.

^ (punktum potens)

· ^ -taster

$Matrix1 \wedge Matrix2 \Rightarrow matrix$

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \wedge \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a^c & 16 \\ b^5 & 3^d \end{bmatrix}$$

$Udtr \wedge Matrix1 \Rightarrow matrix$

$$x \wedge \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} x^c & x^4 \\ x^5 & x^d \end{bmatrix}$$

$Tal \wedge Matrix1 \Rightarrow matrix$

$Matrix1 \wedge Matrix2$ returnerer en matrix, hvor hvert element i $Matrix2$ er eksponent til det tilsvarende element i $Matrix1$.

$Udtr \wedge Matrix1$ returnerer en matrix, hvor hvert element i $Matrix1$ er eksponenten til $Udtr$.

-(neger)

 -tast

$-Udtr1 \Rightarrow$ udtryk

-2.43 -2.43

$-Liste1 \Rightarrow$ liste

$\{-1, 0.4, 1.2 \text{E} 19\}$ $\{1, -0.4, -1.2 \text{E} 19\}$

$-Matrix1 \Rightarrow$ matrix

$-a \cdot -b$ $a \cdot b$

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

$Udtr1 \% \Rightarrow$ udtryk

$Liste1 \% \Rightarrow$ liste

$Matrix1 \% \Rightarrow$ matrix

argument

Returnerer $\frac{\text{argument}}{100}$

For en liste eller matrix returneres en liste eller matrix med hvert element divideret med 100.

Bemærk: Sådanne gennemtvinges et tilnærmelsesresultat,

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

$Udtr1 = Udtr2 \Rightarrow$ Boolsk udtryk

$Liste1 = Liste2 \Rightarrow$ Boolsk liste

$Matrix1 = Matrix2 \Rightarrow$ Boolsk matrix

Returnerer sand, hvis $Udtr1$ bestemmes til at være lig med $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være forskelligt fra $Udtr2$.

Eksempelfunktion, der anvender matematiske testsymboler: =, ≠, <, ≤, >, ≥

= (lig med)

-tast

Alt andet returnerer en reduceret form af ligningen.

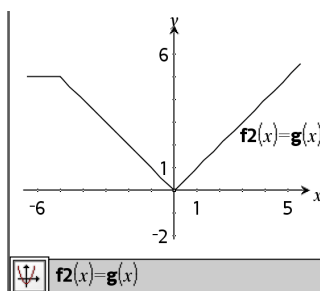
For lister og matricer returneres sammenligninger element for element.

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

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.

$\neq$ (forskellig fra)

  -taster

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive $\neq$

$<$ (mindre end)

  taster

$Udtr1 < Udtr2 \Rightarrow \text{Boolsk udtryk}$

Se " $=$ " (lig med)-eksemplet.

$Liste1 < Liste2 \Rightarrow \text{Boolsk liste}$

$Matrix1 < Matrix2 \Rightarrow \text{Boolsk matrix}$

Returnerer true, hvis $Udtr1$ bestemmes til at være mindre end $Udtr2$.

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.

$\leq$ (mindre end eller lig med)

  taster

$Udtr1 \leq Udtr2 \Rightarrow \text{Boolsk udtryk}$

Se " $=$ " (lig med)-eksemplet.

$Liste1 \leq Liste2 \Rightarrow \text{Boolsk liste}$

$Matrix1 \leq Matrix2 \Rightarrow \text{Boolsk matrix}$

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 $\leq$

> (større end)

  taster

$Udtr1 > Udtr2 \Rightarrow \text{Boolsk udtryk}$

Se “=” (lig med)-eksemplet.

$Liste1 > Liste2 \Rightarrow \text{Boolsk liste}$

$Matrix1 > Matrix2 \Rightarrow \text{Boolsk matrix}$

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.

For lister og matricer returneres sammenligninger element for element.

$\geq$ (større end eller lig med)

  taster

$Udtr1 \geq Udtr2 \Rightarrow \text{Boolsk udtryk}$

Se “=” (lig med)-eksemplet.

$Liste1 \geq Liste2 \Rightarrow \text{Boolsk liste}$

$Matrix1 \geq Matrix2 \Rightarrow \text{Boolsk matrix}$

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 $\geq$

⇒ (medfører)

  -taster

BoolskUdtryk1 ⇒ *BoolskUdtryk2*
returnerer *Boolsk udtryk*

$5 > 3$ or $3 > 5$	true
--------------------	------

BoolskListe1 ⇒ *BoolskListe2* returnerer
Boolsk liste

$5 > 3 \Rightarrow 3 > 5$	false
---------------------------	-------

3 or 4	7
------------	---

$3 \Rightarrow 4$	-4
-------------------	----

BoolskMatrix1 ⇒ *BoolskMatrix2*
returnerer *Boolsk matrix*

$\{1, 2, 3\}$ or $\{3, 2, 1\}$	$\{3, 2, 3\}$
--------------------------------	---------------

$\{1, 2, 3\} \Rightarrow \{3, 2, 1\}$	$\{-1, -1, -3\}$
---------------------------------------	------------------

Heltal1 ⇒ *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 =>

⇔ (ensbetydende med)

  -taster

BoolskUdtryk1 ⇔ *BoolskUdtryk2*
returnerer *Boolsk udtryk*

$5 > 3$ xor $3 > 5$	true
---------------------	------

$5 > 3 \Leftrightarrow 3 > 5$	false
-------------------------------	-------

BoolskListe1 ⇔ *BoolskListe2* returnerer
Boolsk liste

3 xor 4	7
-------------	---

$3 \Leftrightarrow 4$	-8
-----------------------	----

BoolskMatrix1 ⇔ *BoolskMatrix2*
returnerer *Boolsk matrix*

$\{1, 2, 3\}$ xor $\{3, 2, 1\}$	$\{2, 0, 2\}$
---------------------------------	---------------

$\{1, 2, 3\} \Leftrightarrow \{3, 2, 1\}$	$\{-3, -1, -3\}$
---	------------------

Heltal1 ⇔ *Heltal2* returnerer *Heltal*

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***Udtr1!* $\Rightarrow$ *udtryk*

5! 120

Liste1! $\Rightarrow$ *liste* $\{\{5,4,3\}\}!$ $\{120,24,6\}$ *Matrix1!* $\Rightarrow$ *matrix* $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

Returnerer faktultetværdien af argumentet.

For en liste eller matrix returneres en liste eller matrix af elementerne.

& (tilføj)  **-taster***Streng1 & Streng2* $\Rightarrow$ *streng*

"Hello "&"Nick" "Hello Nick"

Returnerer en tekststreng, der er *Streng2* adderet til *Streng1*.**d() (differentialkvotient)****Katalog** > **Udtr1**, *Var*[*Orden*]] $\Rightarrow$ *udtryk* $\frac{d}{dx}(f(x) \cdot g(x))$ $\frac{d}{dx}(f(x)) \cdot g(x) + \frac{d}{dx}(g(x)) \cdot f(x)$ **d**(*l*, *Var*[*Orden*]] $\Rightarrow$ *liste* $\frac{d}{dy}\left(\frac{d}{dx}(x^2 \cdot y^3)\right)$ $6 \cdot y^2 \cdot x$ **d**(*Matrix1*, *Var*[*Orden*]] $\Rightarrow$ *matrix* $\frac{d}{dx}\left(\left\{x^2, x^3, x^4\right\}\right)$ $\left\{2 \cdot x, 3 \cdot x^2, 4 \cdot x^3\right\}$ Returnerer differentialkvotienten af første orden med hensyn til variabelen *Var*.*Orden* skal være et heltal, hvis det medtages. Hvis ordenen er mindre end nul, vil resultatet være en stamfunktion.**Bemærk:** Du kan indsætte denne funktion fra computerens tastatur ved at skrive **derivative (...)**.**d()** følger ikke den normale beregningsmekanisme, der reducerer argumenterne fuldt ud, og derefter anvender funktionsdefinitionen på disse fuldt reducerede argumenter. I stedet udfører **d()** følgende trin:

1. Reducerer kun andet argument i den udstrækning, det ikke fører til en ikke-variabel.
2. Reducerer kun første argument i den udstrækning, det ikke henter en lagret

værdi for variabelen, der er bestemt af trin 1.

- Bestemmer den symbolske differentialkvotient af resultatet af trin 2 med hensyn til variabelen fra trin 1.

Hvis variabelen fra trin 1 har en lagret værdi, der er angivet med ("|")-betingelsesoperatoren, substitueres denne værdi i resultatet fra trin 3.

Bemærk: Se også **Differentialkvotient af første orden** side 5 **Differentialkvotient af anden orden**, side 6 eller **differentialkvotient af Nte orden**, side 6.

∫() (integrer)

∫(Udtr1, Var[, Nedre, Øvre]) ⇒ udtryk

∫(Udtr1, Var[, Konstant]) ⇒ udtryk

Returnerer integralet af *Udtr1* med hensyn til variabelen *var* fra *Nedre* til *Øvre*.

Bemærk: Se også **Bestemt** eller **ubestemt integral-skabelonen**, side 6.

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

Returnerer en stamfunktion, hvis *Nedre* og *Øvre* udelades. En symbolsk integrationskonstant udelades, medmindre du angiver argumentet *Konstant*.

$$\int_a^b x^2 dx = \frac{b^3}{3} - \frac{a^3}{3}$$

$$\int x^2 dx = \frac{x^3}{3}$$

$$\int (a \cdot x^2, x, c) = \frac{a \cdot x^3}{3} + c$$

Stamfunktioner kan adskille sig fra hinanden med en numerisk konstant. En sådan konstant kan være skjult—især når en stamfunktion indeholder logaritmer eller inverse trigonometriske funktioner. Desuden tilføjes stykkevise konstantudtryk nogen gange for at gøre en stamfunktion gyldig over et større interval end den sædvanlige formel.

$\int()$ (integrer)

Katalog > 

$\int()$ returnerer sig selv for dele af *Udtr1*, som ikke kan bestemmes som en eksplicit endelig kombination af dens indbyggede funktioner og operatører.

Når du angiver *Nedre* og *Øvre*, forsøges det at finde eventuelle diskontinuiteter eller diskontinuære differentialkvotienter i intervallet *Nedre* < *Var* < *Øvre* og underdele intervallet på disse steder.

For Auto-indstillingen af

Auto eller tilnærmet-tilstanden anvendes numerisk integration, hvor den er relevant, hvis en stamfunktion eller en grænse ikke kan bestemmes

For Approks.-indstillingen prøves først numerisk integration, hvis det er relevant. Stamfunktioner bestemmes kun, hvor en sådan numerisk integration ikke er relevant eller ikke fungerer.

$\int()$ kan indlejres og dermed klare multiple integraler. Integrationsgrænser kan afhænge af integrationsvariable uden for dem.

Bemærk: Se også *nInt()*, side 126.

$$\int b \cdot e^{-x^2} + \frac{a}{x^2 + a^2} dx \quad b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$$

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på  .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på  + **Enter**.

iPad®: Hold **ENTER** nede, og vælg .

$$\int_{-1}^1 e^{-x^2} dx \quad 1.49365$$

$$\int_0^a \int_0^x \ln(x+y) \, dy \, dx \quad \frac{a^2 \cdot \ln(a)}{2} + \frac{a^2 \cdot (4 \cdot \ln(2) - 3)}{4}$$

$\sqrt{}$ (kvadratrod)

  -taster

$\sqrt{(\text{Udtr1})} \Rightarrow \text{udtryk}$

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

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

$$\sqrt{4} \quad 2$$
$$\sqrt{\{9, a, 4\}} \quad \{3, \sqrt{a}, 2\}$$

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 (...)**.

Beregner *Udtryk1* for hver værdi af *Var* fra *Lav* til *Høj* og returnerer produktet af resultaterne.

Bemærk: Se også **Produktskabelon (Π)**, 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 $Høj < Lav-1$

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.

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

$$\prod_{k=1}^n (k^2) = (n!)^2$$

$$\prod_{n=1}^5 \left(\left\{ \frac{1}{n}, n, 2 \right\} \right) = \left\{ \frac{1}{120}, 120, 32 \right\}$$

$$\prod_{k=4}^3 (k) = 1$$

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) = 6$$

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) \cdot \prod_{k=2}^4 \left(\frac{1}{k} \right) = \frac{1}{4}$$

Σ() (sumSeq)

Katalog > 

$\Sigma(Udtr1, Var, Lav, Høj) \Rightarrow \text{udtryk}$

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

Beregner *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 $Høj < Lav-1$

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.

$\sum_{n=1}^5 \left(\frac{1}{n} \right)$	$\frac{137}{60}$
$\sum_{k=1}^n (k^2)$	$\frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$
$\sum_{n=1}^{\infty} \left(\frac{1}{n^2} \right)$	$\frac{\pi^2}{6}$
$\sum_{k=-4}^3 (k)$	0

$\sum_{k=4}^1 (k)$	-5
$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k)$	4

ΣInt()

Katalog > 

$\Sigma\text{Int}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [afrundVærdi]) \Rightarrow \text{værdi}$

$\Sigma\text{Int}(NPmt1, NPmt2, amortTabel) \Rightarrow \text{værdi}$

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 200.

- Hvis du udelader *Pmt*, sættes den som

$\Sigma\text{Int}(1, 3, 12, 4.75, 20000, 12, 12)$	-213.48
---	---------

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.

$\Sigma\text{Int}(NPmt1, NPmt2, \text{amortTabel})$ beregner summen af renten baseret på amortiseringstabellen amortTabel . amortTabel -argumentet skal være en matrix i formen beskrevet under $\text{amortTbl}()$, side 8.

Bemærk: Se også $\Sigma\text{Prn}()$ nedenfor og $\text{Bal}()$, side 17.

$\text{tbl} := \text{amortTbl}(12, 12, 4.75, 20000., 12, 12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\text{Int}(1, 3, \text{tbl})$ -213.48

$\Sigma\text{Prn}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [\text{afrundVærdi}]) \Rightarrow \text{værdi}$

$\Sigma\text{Prn}(1, 3, 12, 4.75, 20000., 12, 12)$ -4916.28

$\Sigma\text{Prn}(NPmt1, NPmt2, \text{amortTabel}) \Rightarrow \text{værdi}$

Amortiseringsfunktion, der beregner summen af afdraget 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 200.

- 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.

$\text{tbl} := \text{amortTbl}(12, 12, 4.75, 20000., 12, 12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\text{Prn}(1, 3, \text{tbl})$ -4916.28

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

ΣPrn(*NPmt1*,*NPmt2*,*amortTabel*) beregner summen af afdrag på hovedstolen baseret på amortiseringstabellen *amortTabel*.

amortTabel-argumentet skal være en matrix i formen beskrevet under **amortTbl()**, side 8.

Bemærk: Se også ΣInt() ovenfor og Bal(), side 17.

(henvisning)

  -taster

varNavnestreng

#("x"&"y"&"z") xyz

Kalder variablen, hvis navn er *varNavnestreng*. Dermed kan du anvende strenge til at oprette variabelnavne fra en funktion.

Opretter eller kalder variablen xyz.

10 → *r* 10

"r" → *s1* "r"

#*s1* 10

Returnerer værdien af variablen (*r*), hvis navn er lagret i variablen *s1*.

E (videnskabelig notation)

 -tast

mantisseEksponent

23000. 23000.

Indtaster et tal i videnskabelig notation. Tallet fortolkes som *mantisse* × 10^{*eksponent*}.

2300000000.+4.1E15 4.1E15

3·10⁴ 30000

Tip: Hvis du vil indtaste en 10. potens uden at give et decimalværdiresultat, skal du bruge 10[^]*heltal*.

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

$Udtr1^g \Rightarrow udtryk$

$Udtr1^g \Rightarrow udtryk$

$Liste1^g \Rightarrow liste$

$Matrix1^g \Rightarrow matrix$

I vinkeltilstanden Grader, Nygrader eller Radianer:

$$\frac{\cos(50^g)}{\frac{\sqrt{2}}{2}} = \cos\left(\left\{0, 100^g, 200^g\right\}\right) \left\{1, 0, -1\right\}$$

Denne funktion giver mulighed for at angive en vinkel i nygrader, mens programmet er i vinkeltilstanden Grader eller Radianer.

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)

-tast

$Udtr1^r \Rightarrow udtryk$

$Liste1^r \Rightarrow liste$

$Matrix1^r \Rightarrow matrix$

I vinkeltilstanden Grader, Nygrader eller Radianer:

$$\frac{\cos\left(\frac{\pi}{4^r}\right)}{\frac{\sqrt{2}}{2}} = \cos\left(\left\{0^r, \frac{\pi}{12}, \left(\pi\right)^r\right\}\right) \left\{1, \frac{(\sqrt{3}+1) \cdot \sqrt{2}}{4}, -1\right\}$$

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.

° (grader)



$Udr 1^\circ \Rightarrow udtryk$

$Liste 1^\circ \Rightarrow liste$

$Matrix 1^\circ \Rightarrow matrix$

Denne funktion giver mulighed for at angive en vinkel i grader, mens programmet er i vinkeltilstanden Nygrader eller 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.

I vinkeltilstanden Grader, Nygrader eller Radianer:

$$\cos(45^\circ) = \frac{\sqrt{2}}{2}$$

I vinkeltilstanden Radianer:

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) = \{1, 0.707107, 0, 0.864976\}$$

°, ', " (grader/minutter/sekunder)



$gg^\circ mm'ss.ss'' \Rightarrow udtryk$

gg Et positivt eller negativt tal

mm Et ikke-negativt tal

$ss.ss$ Et 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 ("").

I vinkeltilstanden Grader:

$$\begin{array}{r} 25^\circ 13' 17.5'' \\ 25^\circ 30' \end{array} \quad \begin{array}{r} 25.2215 \\ 51 \\ 2 \end{array}$$

∠ (vinkel)



$[Radius, \angle \theta \text{ Vinkel}] \Rightarrow vektor$
(polært input)

I vinkeltilstanden Radian og vektorformatet indstillet til:

∠ (vinkel)

  -taster

[Radius, ∠θ_Vinkel, Z_Koordinat]

⇒vektor

(cylindrisk input)

[Radius, ∠θ_Vinkel, ∠θ_Vinkel]⇒vektor

(sfærisk input)

Returnerer koordinater som en vektor afhængigt af tilstandsindstillingen for vektorformat: rektangulær, cylindrisk eller sfærisk.

Bemærk: Du kan indsætte dette symbol fra computerens tastatur ved at skrive @<.

rektangulær

$$\left[5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \quad \left[\frac{5 \cdot \sqrt{2}}{4} \quad \frac{5 \cdot \sqrt{6}}{4} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

cylindrisk

$$\left[5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \quad \left[\frac{5 \cdot \sqrt{2}}{2} \quad \angle \frac{\pi}{3} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$$

sfærisk

$$\left[5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \quad \left[5 \quad \angle \frac{\pi}{3} \quad \angle \frac{\pi}{4} \right]$$

(Størrelse ∠ Vinkel)⇒kompleksVærdi
(polært input)

Indtaster en kompleks værdi i den polære form ($r\angle\theta$). Vinkel fortolkes efter den aktuelle indstilling af vinkeltilstand.

I vinkeltilstanden radian og rektangulært
komplekst format:

$$5+3 \cdot i - \left(10 \angle \frac{\pi}{4} \right) \quad 5-5 \cdot \sqrt{2} + (3-5 \cdot \sqrt{2}) \cdot i$$

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på  .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

$$5+3 \cdot i - \left(10 \angle \frac{\pi}{4} \right) \quad -2.07107-4.07107 \cdot i$$

' (mærke)

 -tast

variabel '

variabel ''

Indsætter et mærketegn i en differentialligning. Et enkelt mærketegn markerer en differentialligning af første orden, to mærketegn markerer anden orden osv.

$$\text{deSolve} \left(y'' = \frac{-1}{2} \text{ and } y(0)=0 \text{ and } y'(0)=0, t, y \right) \quad \frac{\frac{3}{2} \cdot y^4}{3} = t$$

_ (understregning som en enhedsbenævnelse)

  -taster

Udtr_Enhed

3·_m►_ft 9.84252·_ft

Angiver enhederne i et *Udtr*. Alle enhedsnavne skal begyndes med en understregning.

Bemærk: Du kan finde konverteringstegnet,

►, i Katalog. Klik på , og klik derefter på **Matematiske operatører**.

Du kan anvende prædefinerede enheder eller oprette dine egne enheder. Du får en liste med prædefinerede enheder ved at åbne Katalog og vise fanen med enhedsomregninger. Du kan vælge enhederne fra Katalog eller skrive enhederne direkte.

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.

$\text{real}(z)$	z
$\text{real}(z_)$	$\text{real}(z_)$
$\text{imag}(z)$	0
$\text{imag}(z_)$	$\text{imag}(z_)$

Hvis *Variabel* har en værdi, ignoreres _, og *Variabel* bevarer sin oprindelige datatype.

Bemærk: Du kan lagre et komplekst tal i en variabel uden

at anvende _. Men for det bedste resultat i beregninger som **cSolve()** og **cZeros()** anbefales_.

► (konverter)

  -taster

Udtr_Enhed1 ►_Enhed2⇒Udtr_Enhed2

3·_m►_ft 9.84252·_ft

Konverterer et udtryk mellem enheder.

Understregningstegnet _ benævner enhederne. Enhederne skal være i samme kategori, for eksempel Længde eller Areal.

► (konverter)

  -taster

Du får en liste med prædefinerede enheder ved at åbne Catalog og vise fanen med enhedsomregninger:

- Du kan vælge et enhed fra listen.
- Du kan vælge konverteringsoperatoren ► øverst på listen.

Du kan også skrive enhederne manuelt. For at skrive “_”, når du indtaster enheder på den håndholdte, trykker du  .

Bemærk: Temperaturenhederne konverteres med **tmpCnv()** og **ΔtmpCnv()**. Konverteringsoperatoren ► håndterer ikke temperaturenheder.

10^()

Katalog > 

10^ (Udtr1)⇒udtryk

$10^{1.5}$ 31.6228

10^ (Liste1)⇒liste

$10^{\{0, -2, 2, a\}}$ $\left\{1, \frac{1}{100}, 100, 10^a\right\}$

Returnerer 10 opløftet til potensen af argumentet.

I en liste returneres 10 opløftet til potensen af elementerne i *Liste1*.

10^(kvadratMatrix1)⇒kvadratmatrix

$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}}$

1.14336E7	8.17155E6	6.67589E6
9.95651E6	7.11587E6	5.81342E6
7.65298E6	5.46952E6	4.46845E6

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()**.

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

^-1(reciprok)

Katalog > 

Udtr1 ^-1⇒udtryk

$(3.1)^{-1}$ 0.322581

Liste1 ^-1⇒liste

$\{a, 4, -0.1, x, -2\}^{-1}$ $\left\{\frac{1}{a}, \frac{1}{4}, -10, \frac{1}{x}, -\frac{1}{2}\right\}$

Returnerer den reciproke værdi af argumentet.

For en liste returneres de reciprokke værdier af elementerne i *Liste1*.

kvadratMatrix1 $\wedge^{-1} \Rightarrow$ *kvadratMatrix*

Returnerer den inverse værdi af *kvadratMatrix1*.

kvadratMatrix1 skal være en ikke-singulær kvadratisk matrix.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{-2}{a-2} & \frac{1}{a-2} \\ \frac{a}{2 \cdot (a-2)} & \frac{-1}{2 \cdot (a-2)} \end{bmatrix}$$

| (betingelses-operator)

ctrl -taster

Udtryk | *BoolskUdtryk1*
[*andBoolskUdtryk2*]...

$$x+1|x=3 \quad 4$$

$$x+y|x=\sin(y) \quad \sin(y)+y$$

Udtryk | *BoolskUdtryk1*
[*orBoolskUdtryk2*]...

$$x+y|\sin(y)=x \quad x+y$$

("|")-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
- Udelukkelser

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.

$$x^3-2 \cdot x+7 \rightarrow f(x) \quad Done$$

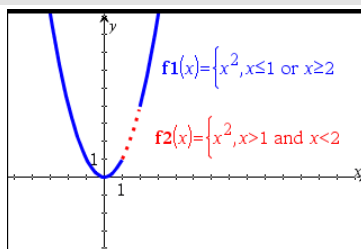
$$f(x)|x=\sqrt{3} \quad \sqrt{3}+7$$

$$(\sin(x))^2+2 \cdot \sin(x)-6|\sin(x)=d \quad d^2+2 \cdot d-6$$

$$\text{solve}(x^2-1=0,x)|x>0 \text{ and } x<2 \quad x=1$$

$$\sqrt{x} \cdot \sqrt{\frac{1}{x}}|x>0 \quad 1$$

$$\sqrt{x} \cdot \sqrt{\frac{1}{x}} \quad \sqrt{\frac{1}{x}} \cdot \sqrt{x}$$



Udelukkelser benytter "forskellig fra"-kommandoen ($\neq$ eller $\neq$) til at udelukke en specifik værdi fra reduktionen. De anvendes primært til at udelukke en eksakt løsning ved anvendelse af **cSolve()**, **cZeros()**, **fMax()**, **fMin()**, **solve()**, **zeros()** osv.

$$\text{solve}(x^2 - 1 = 0, x) | x \neq 1 \quad x = -1$$

→ (lagring)

ctrl  -tast

Udtr → *Var*

Liste → *Var*

Matrix → *Var*

Udtr → *Funktion*(*Param1*,...)

List → *Funktion*(*Param1*,...)

Matrix → *Funktion*(*Param1*,...)

Hvis variabelen *Var* ikke findes, oprettes *Var* og initialiseres til *Udtr*, *Liste* eller *Matrix*.

Hvis *Var* findes i forvejen og ikke er låst eller beskyttet, erstattes dens indhold med *Udtr*, *Liste* eller *Matrix*.

Tip: Hvis du vil udføre symbolske beregninger med udefinerede variable, skal du undgå at lagre noget i almindeligt brugte et-bogstavsvariable som a, b, c, x, y, z osv.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive **=:** som en genvej. Skriv for eksempel **pi/4 =:** **myvar**.

$\frac{\pi}{4} \rightarrow \text{myvar}$	$\frac{\pi}{4}$
$2 \cdot \cos(x) \rightarrow y1(x)$	Done
$\{1, 2, 3, 4\} \rightarrow \text{lst5}$	$\{1, 2, 3, 4\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \text{matg}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
"Hello" → str1	"Hello"

:= (tildel)

  -taster

Var := *Udtr*

Var := *Liste*

Var := *Matrix*

Funktion(*Param1*,...) := *Udtr*

Funktion(*Param1*,...) := *Liste*

Funktion(*Param1*,...) := *Matrix*

Hvis variablen *Var* ikke findes, oprettes *Var* og initialiseres til *Udtr*, *Liste* eller *Matrix*.

Hvis *Var* findes i forvejen og ikke er låst eller beskyttet, erstattes dens indhold med *Udtr*, *Liste* eller *Matrix*.

Tip: Hvis du vil udføre symbolske beregninger med udefinerede variable, skal du undgå at lagre noget i almindeligt brugte et-bogstavsvariable som a, b, c, x, y, z osv.

$myvar := \frac{\pi}{4}$	$\frac{\pi}{4}$
$yI(x) := 2 \cdot \cos(x)$	Done
$lst5 := \{1, 2, 3, 4\}$	$\{1, 2, 3, 4\}$
$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
$str1 := \text{"Hello"}$	"Hello"

© (kommentar)

  -taster

© [*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 $g(n) = \text{Func}$	
© Declare variables	
Local <i>i</i> , <i>result</i>	
<i>result</i> := 0	
For <i>i</i> , 1, <i>n</i> , 1 © Loop <i>n</i> times	
<i>result</i> := <i>result</i> + i^2	
EndFor	
Return <i>result</i>	
EndFunc	
$g(3)$	14

0b, 0h

  -taster,   -taster

0b *binærtTal*

I decimal tilstand:

0h *hexadecimaltTal*

0b10+0hF+10 27

Betegner henholdsvis binært eller hexadecimalt tal. For at indtaste et binært eller hexadecimalt tal skal du indtaste præfikset 0b eller 0h uanset talsystemet. Uden præfiks behandles tallet som decimaltal (10 talssystem).

Resultatet vises i den valgte tilstand for talsystem.

I binær tilstand:

0b10+0hF+10	0b11011
-------------	---------

I hexadecimal tilstand:

0b10+0hF+10	0h1B
-------------	------

Tomme (ugyldige) elementer

Ved analyse af data fra virkeligheden har du muligvis ikke et komplet datasæt. TI-Nspire™ CAS 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 50, og **isVoid()**, side 97.

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 list1,list2	Done
list1	$\{1,3,4,5,_{-}\}$
list2	$\{1,3,4,5,2\}$

$\{1,2,3,_,5\} \rightarrow list1$	$\{1,2,3,_,5\}$
$\{1,2,3,4,5\} \rightarrow list2$	$\{1,2,3,4,5\}$
SortD list1,list2	Done
list1	$\{5,3,2,1,_{-}\}$
list2	$\{5,3,2,1,4\}$

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 l1,l2	Done
stat.Resid	$\{0.434286,_, -0.862857, -0.011429, 0.44\}$
stat.XReg	$\{1,_,3,4,5\}$
stat.YReg	$\{2,_,3,5,6,6\}$
stat.FreqReg	$\{1,_,1,1,1,1\}$

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 l1,l2,1,cat,incl	Done
stat.Resid	$\{_,_,0,0,0\}$
stat.XReg	$\{_,_,4,5\}$
stat.YReg	$\{_,_,5,6,6\}$
stat.FreqReg	$\{_,_,1,1,1\}$

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 l1,l2,{1,0,1,1}	Done
stat.Resid	$\{0.069231,_, -0.276923, 0.207692\}$
stat.XReg	$\{1,_,4,5\}$
stat.YReg	$\{2,_,5,6,6\}$
stat.FreqReg	$\{1,_,1,1,1\}$

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:
π	<code>pi</code>
θ	<code>theta</code>
∞	<code>uendeligt</code>
$\leq$	<code><=</code>
$\geq$	<code>>=</code>
$\neq$	<code>/=</code>
$\Rightarrow$ (medfører)	<code>=></code>
$\Leftrightarrow$ (ensbetydende med)	<code><=></code>
$\rightarrow$ (gem operator)	<code>=:</code>
$ $ (absolut værdi)	<code>abs (...)</code>
$\sqrt{()}$	<code>sqrt (...)</code>
$d()$	<code>derivative (...)</code>
$\int()$	<code>integral (...)</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>
ΔtmpCnv	<code>deltaTmpCnv (...)</code>

På computerens tastatur

Indsæt dette:	- ved at skrive denne genvej:
$c1, c2, \dots$ (konstanter)	<code>@c1, @c2, ...</code>
$n1, n2, \dots$ (heltalskonstanter)	<code>@n1, @n2, ...</code>
i (imaginær konstant)	<code>@i</code>

Indsæt dette:	- ved at skrive denne genvej:
e (naturlig logbase e)	@ e
E (videnskabelig notation)	@ E
T (transponere)	@ t
r (radianer)	@ r
° (grader)	@ d
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™ CAS 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 [], krølparenteser { }
2	Henviſning (nr.)
3	Funktionskald
4	Postoperatorer: grader-minutter-sekunder (°, ', "), fakultet (!), procentsats (%), radian (°), lavtstillet ([]), skifte side for lighedstegnet (°)
5	Eksponentiering, potensoperator (^)
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 not
12	logisk and
13	Logisk or
14	xor, nor, nand
15	medfører, ⇒
16	ensbetydende med (⇔)
17	Betingelses-operator (" ")
18	Gem (→)

Parenteser, kantparenteser og krølparenteser

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™ CAS 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 $4^3!$ beregnes $3!$ først. Resultatet, 6, bliver derefter eksponenten af 4, så det bliver 4096.

Ekspontiering

Ekspontiering (^) og elementvis ekspontiering (.^) beregnes fra højre mod venstre. Udtrykket 2^3^2 beregnes for eksempel på samme måde som 2^3^2 og giver 512. Dette er forskelligt fra $(2^3)^2$, der giver 64.

Negation

Negative tal indtastes ved at trykke på $\boxed{-}$ efterfulgt af tallet. Postoperationer og ekspontiering 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.

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	8987551787.3682 _m/_F
_Fc	Faradays konstant	96485.33289 _coul/_mol
_g	Tyngdeaccelerationen	9.80665 _m/_s ²
_Gc	Tyngdekonstant	6.67408E-11 _m ³ /_kg/_s ²
_h	Plancks konstant	6.626070040E-34 _J_s
_k	Boltzmanns konstant	1.38064852E-23 _J/_K
_μ0	Vakuumpermeabiliteten	1.2566370614359E-6 _N/_A ²
_μb	Bohr-magneton	9.274009994E-24 _J_m ² /_Wb
_Me	Elektronens hvilemasse	9.10938356E-31 _kg
_Mμ	Myonens masse	1.883531594E-28 _kg
_Mn	Neutronens hvilemasse	1.674927471E-27 _kg
_Mp	Protonens hvilemasse	1.672621898E-27 _kg
_Na	Avogadros konstant	6.022140857E23 /_mol
_q	Elementarladningen	1.6021766208E-19 _coul
_Rb	Bohr-radius	5.2917721067E-11 _m
_Rc	Gaskonstanten	8.3144598 _J/_mol/_°K
_Rdb	Rydbergs konstant	10973731.568508/_m
_Re	Elektronradius	2.8179403227E-15 _m
_u	Atommasse	1.660539040E-27 _kg
_Vm	Det molare volumen	2.2413962E-2 _m ³ /_mol
_ε0	Permittivitet for et vakuum (konstant)	8.8541878176204E-12 _F/_m
_σ	Stefan-Boltzmanns konstant	5.670367E-8 _W/_m ² /_°K ⁴
_φ0	Magnetisk fluxkvantum	2.067833831E-15 _Wb

Fejlkode og fejlmeddelelser

Hvis der opstår en fejl, tildeles dens kode variabelen `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 196.

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<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: • ikke begynder med et tal • ikke indeholder mellemrum eller specialtegn • ikke anvender understregning eller punktum forkert • ikke overskrider begrænsningerne for længden 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 Nedre grænse skal være mindre end øvre grænse for at definere søgeintervallet.

Fejlkode	Beskrivelse
180	Afbrudt Der er blevet trykket på  eller  -tasten under en længere beregning eller under en programeksekvering.
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 udefineret variabel give denne fejl.
200	Ugyldigt begrænsningsudtryk For eksempel vil $\text{solve}(3x^2-4=0, x) \mid x < 0 \text{ or } x > 5$ 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 $\text{rand}(0)$ ikke gyldig.
270	Variabelnavn findes allerede
280	Else og Elseif er ikke gyldige uden for blokken <code>if..endif</code>
290	EndTry mangler den tilsvarende Else-erklæring
295	For stor iteration
300	Forventede en liste med 2 eller 3 elementer eller en matrix
310	Første argument af <code>nSolve</code> 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 <code>solve</code> eller <code>cSolve</code> skal være en ligning eller en ulighed For eksempel er $\text{solve}(3x^2-4, x)$ ugyldig, fordi det første argument ikke er en ligning.

Fejlkode	Beskrivelse
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.
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

Fejlkode	Beskrivelse
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
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

Fejlkode	Beskrivelse
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: • <code>sto →</code> • <code>:=</code> • Define for at tildele værdier til variable.
965	OS uden licens
970	Variabel i brug, så referencer eller ændringer er ikke tilladt
980	Variablen er beskyttet
990	Ugyldigt variabelnavn Kontroller, at navnet ikke overskrider begrænsningerne for længden
1000	Talområde for Vindue variabel
1010	Zoom

Fejlkode	Beskrivelse
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: • Define • := • sto → 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 Kompleks 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 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 xxx\yyy, hvor: • xxx-delen kan have 1 til 16 tegn. • yyy-delen kan have 1 til 15 tegn.

Fejlkode	Beskrivelse
	Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1170	Ugyldig brug af biblioteksstinavn • En værdi kan ikke tildeles et stinavn med Define, := eller sto →. • Et stinavn kan ikke erklæres som en lokal variabel eller anvendes som en parameter i en funktion eller en programdefinition.
1180	Ugyldigt biblioteksvariabelnavn Kontroller, at navnet: • Ikke indeholder et punktum • Ikke begynder med en understregning • Ikke er længere end 15 tegn Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1190	Biblioteksdokumentet blev ikke fundet. • Kontroller, at biblioteket er i MyLib folder. • Opdater biblioteker. Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1200	Biblioteksvariablen blev ikke fundet. • Kontroller, at biblioteksvariablen findes i den første opgave i biblioteket. • Kontroller, at biblioteksvariablen er defineret som LibPub eller LibPriv. • Opdater biblioteker. Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1210	Ugyldigt navn på biblioteksgenvej Kontroller, at navnet: • Ikke indeholder et punktum • Ikke begynder med en understregning • Ikke er længere end 16 tegn • Ikke er et reserveret navn Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1220	Domænefejl: 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 .

Fejlkode	Beskrivelse
	Eksempel på et system med to lineære ligninger med variablene x og y: $3x+7y=5$ $2y-5x=-1$
1260	Argumentfejl: Det første argument til <code>nfMin</code> eller <code>nfMax</code> 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 <code>domain()</code>-funktionen.

Fejlkode og -meddelelser

Du kan bruge funktionen **warnCodes()** til at gemme fejlkode genereret ved beregning af et udtryk. Denne tabel viser de numeriske fejlkode og de tilhørende meddelelser.

Se **warnCodes()**, side 205 for at få et eksempel på lagring af fejlkode.

Fejlkode	Meddelelse
10000	Operationen kan måske indføre falske løsninger.
10001	Differentiation af en ligning kan give en falsk ligning.
10002	Tvivlsom løsning
10003	Tvivlsom præcision
10004	Operationen kan miste løsninger.
10005	cSolve kan give flere nulpunkter.
10006	Solve kan give flere nulpunkter.
10007	Der kan være flere løsninger. Forsøg at specificere relevante nedre og øvre grænser og/eller et gæt. Eksempler med brug af solve(): • <code>solve(Ligning, Var=Gæt) nedreGrænse<Var<øvreGrænse</code> • <code>solve(Ligning, Var) nedreGrænse<Var<øvreGrænse</code> • <code>solve(Ligning,Var=Gæt)</code>
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	∞^0 eller undef^0 erstattet med 1
10014	undef^0 erstattet med 1
10015	1^∞ eller 1^undef erstattet med 1
10016	1^undef erstattet med 1
10017	Overflow erstattet med ∞ 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 parameterverdier.

Fejlkode	Meddelelse
10022	Angivelse af korrekte nedre og øvre grænser kan give en løsning.
10023	Skalaren er multipliceret med identitetsmatricen.
10024	Resultatet er opnået vha. tilnærmet aritmetik.
10025	Ækvivalensen kan ikke verificeres i tilstanden EXACT.
10026	Begrænsningen ignoreres muligvis. Angiv begrænsningen på formen "\" 'Variabel MatTestSymbol Konstant' eller en konjunktion af disse former, f.eks. 'x<3 and x>-12'

Generelle oplysninger

Online hjælp

education.ti.com/eguide

Vælg dit land for at få flere produktoplysninger.

Kontakt TI-Support

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

Vælg dit land vedrørende oplysninger om garantiens varighed og betingelser eller om produktservice.

Begrænset reklamationsret. Denne garanti påvirker ikke dine lovbestemte rettigheder

Indeks

-	216	$\wedge$, potens	219
-, subtrahere	216	—	
!	227	_, enhedsbenævnelse	237
!, fakultet	227		
"	235	, betingelses-operator	239
", sekundnotation	235	,	
#	233	', minutnotation	235
#, henvisning	233	', mærke	236
#, henvisningsoperator	248	+	
%	222	+, adder	216
%, procent	222	≠	
&	227	≠, forskellig fra	223
&, tilføj	227	=	
*	217	=, lig med	222
*, multiplicere	217	>	
.	220	>, større end	225
., punktum subtraktion	220	Π	
., punktum multiplikation	221	Π , produkt	230
./, punkt division	221	Σ	
., punktum potens	221	$\Sigma()$, sum	231
., punktum addition	220	$\Sigma\text{Int}()$	231
/	218	$\Sigma\text{Prn}()$	232
/, dividere	218	$\sqrt{}$	
:	241	$\sqrt{}$, kvadratrods	229
:=, tilde	241	$\int$	
$\wedge$	238	$\int$, integrer	228
$\wedge^{-1}$, reciprok	238		

$\leq$		$\circ$	
$\leq$, mindre end eller lig med	224	$^\circ$, grader/minutter/sekunder	235
$\geq$		$^\circ$, gradnotation	235
$\geq$, større end eller lig med	225	0	
$\blacktriangleright$		Ob, binær indikator	241
$\blacktriangleright$, konvertere enheder	237	Oh, hexadecimal indikator	241
$\blacktriangleright$ approxFraction()	14	1	
$\blacktriangleright$ Base10, vise som decimalt heltal	19	$10^{\wedge}()$, tiende potens	238
$\blacktriangleright$ Base16, vise som hexadecimalt	20	A	
$\blacktriangleright$ Base2, vise som binær	18	abs(), absolut værdi	8
$\blacktriangleright$ cos, vises i termer af cosinus	31	Absolut værdi	
$\blacktriangleright$ Cylind, vise som cylindrisk vektor	44	skabelon til	3-4
$\blacktriangleright$ DD, vise som decimal vinkel	47	adder, +	216
$\blacktriangleright$ Decimal, vise resultat som decimal	47	afrund, round()	159
$\blacktriangleright$ DMS, vise som		afslut løkke, EndLoop	113
grader/minutter/sekunder	56	afslut, Exit	65
$\blacktriangleright$ exp, viser i termer af e	65	afslutte funktion, EndFunc	79
$\blacktriangleright$ Grad, konvertere til vinkelmål i		amortiseringstabel, amortTbl()	8, 17
nygrader	89	amortTbl(), amortiseringstabel	8, 17
$\blacktriangleright$ Polar, vise som polær vektor	136	and, Boolsk operator	9
$\blacktriangleright$ Rad, konverter til vinkelmål i		anden afledede	
radianer	147	skabelon til	6
$\blacktriangleright$ Rect, vis som rektangulær vektor	150	andengradsregression, QuadReg	144
$\blacktriangleright$ sin, vis udtryk i termer af sinus	171	angle(), vinkel	10
$\blacktriangleright$ Sphere, vise som en kuglevektor	179	Anmodning	153
$\blacktriangleright$ tmpCnv()	194	ANOVA, envejs variansanalyse	10
$\rightarrow$		ANOVA2way, to-vejs variansanalyse	11
$\rightarrow$, lagring	240	ans, sidste resultat	13
$\Rightarrow$		approx(), tilnærmet	14
$\Rightarrow$, medfører[*]	226, 245	approxRational()	14
$\Leftrightarrow$		arccos()	15
$\Leftrightarrow$, ensbetydende med[*]	226	arccosh()	15
$\copyright$		arccot()	15
$\copyright$, kommentar	241	arcoth()	15
		arccsc()	15
		arccsch()	15
		arcLen(), buemål	15
		arcsec()	15
		arcsech()	16

arcsin()	16
arcsinh()	16
arctan()	16
arctanh()	16
arcuscosinus, $\cos^{-1}()$	32
arcussinus, $\sin^{-1}()$	172
arcustangens, $\tan^{-1}()$	187
argumenter i TVM-funktioner	200
augment(), udvid/sammenkæd	16
avgRC(), gennemsnitlig ændringshastighed	17

B

beregne polynomium, polyEval()	138
beregning, rækkefølge af	247
Bestemt integral skabelon til	6
betingelses-operator " "	239
betingelses-operator, rækkefølge for en beregning	247
Bibliotek (Library) danner genveje til objekter	99
binomCdf()	21, 95
binomPdf()	21
binær indikator, 0b	241
vise, ▶Base2	18
Boolske operatører ⇒	226, 245
⇐	226
and	9
nand	123
nor	127
not	129
or	133
xor	206
brugerdefinerede funktioner	47
brugerdefinerede funktioner og programmer	49
brøker skabelon til	1
ægteBrøk	142
buemål, arcLen()	15

C

Cdf()	71
ceiling(), rund op	21
centralDiff()	22
cFactor(), kompleks faktor	22
char(), tegnstreng	23
charPoly()	24
χ^2 2way	24
ClearAZ	26
ClrErr, slet fejl	26
colAugment	27
colDim(), matrix kolonnedimension	27
colNorm(), matrix kolonnenorm	27
comDenom(), fællesnævner	27
completeSquare()	28
conj(), komplekst konjugat	29
constructMat(), konstruerer matrix	29
corrMat(), korrelationsmatrix	30
$\cos^{-1}$, arcuscosinus	32
cos(), cosinus	31
cosh $^{-1}()$, hyperbolsk arcuscosinus	34
cosh(), hyperbolsk cosinus	33
cosinus viser udtryk ud fra	31
cosinus, cos()	31
cot $^{-1}()$, arcuscotangens	35
cot(), cotangens	34
cotangens, cot()	34
coth $^{-1}()$, hyperbolsk arcuscotangens	36
coth(), hyperbolsk cotangens	35
count(), tælle elementer i en liste	36
countif(), tælle elementer i en liste betinget	36
cPolyRoots()	37
crossP(), vektorprodukt	38
csc $^{-1}()$, invers cosecans	38
csc(), cosecans	38
csch $^{-1}()$, invers hyperbolsk cosecans	39
csch(), hyperbolsk cosecans	39
cSolve(), kompleks løse	39
CubicReg, tredjegradsregression	42
cumulative sum, cumulativeSum()	43
cumulativeSum(), cumulative sum	43

Cycle, cyklus	43	DispAt	55
cyklus, Cycle	43	distributionsfunktioner	
cylindrisk vektorvisning, ►Cylind	44	binomCdf()	21
cZeros(), komplekse nulpunkter	44	binomPdf()	21
		normCdf()	128
D		normPdf()	128
d(), første differentialkvotient	227	poissCdf()	136
dage mellem datoer, dbd()	46	poissPdf()	136
dbd(), dage mellem datoer	46	tCdf()	190
decimal		tPdf()	195
heltal vise, ►Base10	19	χ^2 2way()	24
vinkelvisning, ►DD	47	χ^2 Cdf()	25
Define	47	χ^2 GOF()	25
Define LibPriv	49	χ^2 Pdf()	26
Define LibPub	49	dividere, P	218
Define, definere	47	division af heltal, intDiv()	93
definere		domain(), definitions­mængde-	
offentlig funktion eller program	49	kommando	57
privat funktion eller program ..	49	dominantTerm(), dominerende led	58
definere, Define	47	dominerende led, dominantTerm()	58
definitions­mængde-kommando,		dot	
domain()	57	produkt, dotP()	59
delmatrix, subMat()	184, 186	dotP(), prikprodukt	59
deltalist()	50		
deltaTmpCnv()	50	E	
DelVar, slette variabel	50	e eksponent	
delVoid(), fjerner ugyldige elementer	50	skabelon til	2
derivative()	51	e til en potens, e [^] ()	59, 66
deSolve(), løsning	51	E, eksponent	233
det(), matrixdeterminant	53	e, viser udtryk i termer af	65
diag(), matrixdiagonal	53	e [^] (), e til en potens	59
differentialkvotient eller		eff(), omregn nominel til effektiv	
differentialkvotienten af nte		rente	60
orden		effektiv rente, eff()	60
skabelon til	6	egenvektor, eigVc()	60
differentialkvotienter		egenværdi, eigVl()	61
første differentialkvotient, d() ..	227	eigVc(), egenvektor	60
numerisk differentialkvotient,		eigVl(), egenværdi	61
nDeriv()	125-126	eksakt, exact()	65
numerisk differentialkvotient,		eksponent, E	233
nDerivative()	124	eksponenter	
dim(), dimension	54	skabelon til	1
dimension, dim()	54	eksponentiel regression, ExpReg	68
Disp, vis data	54, 162	ellers hvis, Elseif	61

ellers, Else	90	fMin(), funktionsminimum	74
Elseif, ellers hvis	61	For	75
end		for, For	75
for, EndFor	75	For, for	75
løkke...EndLoop	113	fordelingsfunktioner	
EndWhile, slut while-kompleks	206	binomCdf()	95
enheder		invNorm()	96
konvertere	237	invt()	96
enhedsvektor, unitV()	202	Inv χ^2 ()	94
ensbetydende med, $\Leftrightarrow$	226	format(), formatere streng	75
EOS ligningsoperativsystem		formatere streng, format()	75
(Equation Operating		forskellig fra, $\neq$	223
System)	247	fortegn, sign()	169
Equation Operating System (EOS)		fpart(), funktionsdel	76
ligningsoperativsystem	247	freqTable()	76
etiket, Lbl	98	frequency()	77
euler(), Euler function	62	Frobenius-norm, norm()	128
exact(), eksakt	65	Func, funktion	79
Exit, afslut	65	Func, programfunktion	79
exp(), e til en potens	66	funktioner	
exp►list(), udtryk til liste	66	brugerdefinerede	47
expand(), ledform	67	del, fpart()	76
expr(), streng til udtryk	68, 110	maksimum, fMax()	73
ExpReg, eksponentiel regression	68	minimum, fMin()	74
		programfunktion, Func	79
F		funktioner og variable	
F-test- med 2 målinger	78	kopiere	30
factor(), opløse i faktorer	69	fællesnævner, comDenom()	27
fakultet, !	227	første orden	
fejlkoder og -meddelelser	258	skabelon til	5
fejlmeddelelser og fejlfinding			
slette fejl, ClrErr	26	G	
videresende fejl, PassErr	135	g, nygrader	234
Fill, matrixudfyldning	72	gcd(), største fælles divisor	79
finansfunktioner, tvnFV()	198	gennemsnitlig ændringshastighed,	
finansfunktioner, tvnI()	199	avgRC()	17
finansfunktioner, tvnN()	199	genveje, tastatur	245
finansfunktioner, tvnPmt()	199	geomCdf()	80
finansfunktioner, tvnPv()	200	geomPdf()	80
FiveNumSammendrag	72	Get	80
fjerdegradsregression, QuartReg	145	getDenom(), hente/returnere	
floor(), runde ned	73	nævner	82
flyt, shift()	168	getKey()	82
fMax(), funktionsmaksimum	73	getLangInfo(), hent/returner	85

sproginformation	
getLockInfo(), tester låsestatus af en variabel eller variabelgruppe	86
getMode(), hente tilstandsindstillinger	86
getNum(), hente/returnere tal	87
GetStr	87
getType(), get type of variable	88
getVarInfo(), hent/returner variabeloplysninger	88
Goto, gå til	89
grader/minutter/sekunder notation	235
gradnotation, g	235
grupper, låse og låse op	109, 203
grupper, teste låsestatus	86
grænseværdi skabelon til	7
gå til, Goto	89

H

heltal, int()	93
heltalsdel, iPart()	96
hente/returnere nævner, getDenom()	82
tal, getNum()	87
variabeloplysninger, getVarInfo()	85, 88
henvisning, #	233
henvisningsoperator (#)	248
hexadecimal indikator, Oh	241
vise, ►Base16	20
hvis, If	90
hvisFn()	91
hyperbolsk arctangens, $\tanh^{-1}()$	189
arcuscosinus, $\cosh^{-1}()$	34
arcussinus, $\sinh^{-1}()$	173
cosinus, cosh()	33
sinus, sinh()	173
tangens, tanh()	188
højre, right()	94, 156

I

i streng, inString()	93
identitetsmatrix, identity()	90
identity(), identitetsmatrix	90
If, hvis	90
imag(), imaginær del	92
imaginær del, imag()	92
ImpDif(), implicit differentialkvotient	92
implicit differentialkvotient, Impdif()	92
indstille tilstand, setMode()	166
indstillinger, hente aktuelle	86
input, Input	92
Input, input	92
inString(), i streng	93
int(), heltal	93
intDiv(), division af heltal	93
integrere, f	228
interpolate(), interpolation	94
invers kumuleret normalfordeling (invNorm()	96
invers, $\wedge^{-1}$	238
invF()	94
invNorm(), invers kumuleret normalfordeling)	96
invt()	96
Inv χ^2 ()	94
iPart(), heltalsdel	96
irr(), intern rente intern rente, irr()	96
isPrime(), primaltest	97
isVoid(), test for ugyldighed	97

K

kombinationer, nCr()	124
kommentar, ©	241
kompleks faktor, cFactor()	22
løse, cSolve()	39
nulpunkter, cZeros()	44
komplekst konjugat, conj()	29

Loop, løkke	113	udvid/sammenkæd, augment()	16
LU, matrix opløsning i øvre-nedre ..	113	vilkårlige, randMat()	148
længde af streng	54	matrix (1 × 2)	
løkke, Loop	113	skabelon til	4
løs, solve()	175	matrix (2 × 1)	
løsning, deSolve()	51	skabelon til	4
låse variable og variabelgrupper	109	matrix (2 × 2)	
		skabelon til	4
		matrix (m × n)	
		skabelon til	4
		matrix til liste, mat►list()	114
		matrixer	
matricer		række-echelonform, ref()	150
cumulative sum,		rækkeaddition, rowAdd()	159
cumulativeSum()	43	rækkedimension, rowDim()	159
delmatrix, subMat()	184, 186	rækkenorm, rowNorm()	160
determinant, det()	53	rækkeskift, rowSwap()	160
diagonal, diag()	53	max(), maksimum	114
dimension, dim()	54	mean(), middel	115
egenvektor, eigVc()	60	med, 	239
egenværdi, eigVl()	61	medfører, ⇒	226, 245
identitet, identity()	90	median(), median	115
kolonnedimension, colDim()	27	median, median()	115
kolonnenorm, colNorm()	27	medium-medium linjeregression,	
liste til matrix, list►mat()	106	MedMed	116
maksimum, max()	114	MedMed, medium-medium	
matrix til liste, mat►list()	114	linjeregression	116
minimum, min()	118	mens, While	206
multiplikation og addition af		mid(), midt-streng	117
rækker, mRowAdd()	120	middel, mean()	115
ny, newMat()	125	midt-streng, mid()	117
opløsning i øvre-nedre, LU	113	min(), minimum	118
produkt, product()	142	mindre end eller lig med, {	224
punktum addition, .+	220	mindste fælles multiplum, lcm	98
punktum division, ./	221	minimum, min()	118
punktum multiplikation, .*	221	minutnotation,	235
punktum potens, .^	221	mirr(), modificeret intern	
punktum subtraktion, .-	220	forrentning	118
QR opløse i faktorer, QR	143	mod(), modulo	119
reduceret række-echelonform,		modificeret intern forrentning, mirr(	
rref()	160	)	118
rækkeoperation, mRow()	119	modulo, mod()	119
summation, sum()	185	mRow(), matrixrækkeoperation ...	119
transponere, T	186	mRowAdd(), multiplikation og	
udfylde, Fill	72	addition af matrixrækker ..	120

Multipel lineær regression ttest	121	løsning, nSolve()	130
multiplacere, *	217	ny	
MultReg	120	liste, newList()	125
MultRegIntervals()	120	matrix, newMat()	125
MultRegTests()	121	nygrader notation, g	234
mærke, '	236	nævner	27
		når, when()	205
N		O	
nand, Boolsk operator	123	objekter	
naturlig logaritme, ln()	107	danner genveje til bibliotek	99
nCr(), kombinationer	124	OneVar, statistik med en variabel	131
nDerivative(), numerisk		operatorer	
differentialkvotient	124	rækkefølge af beregning	247
negation, indtaste negative tal	248	opløse i faktorer, factor()	69
nettonutidsværdi, npv()	130	oplåsning af variable og	
newList(), ny liste	125	variabelgrupper	203
newMat(), ny matrix	125	or (Boolsk), or	133
nfMax(), numerisk		or, Boolske operatorer	133
funktionsmaksimum	125	ord(), numerisk tegnkode	134
nfMin(), numerisk			
funktionsminimum	126	P	
nInt(), numerisk integral	126	P►Rx(), rektangulært x-koordinat	134
nom), omregn effektiv til nominal		P►Ry(), rektangulært y-koordinat	134
rente	127	PassErr, videregående fejl	135
nominel rente, nom()	127	Pdf()	76
nor, Boolsk operator	127	permutationer, nPr()	129
norm(), Frobenius-norm	128	piecewise()	135
normal linje, normalLine()	128	poissCdf()	136
normalLine()	128	poissPdf()	136
normalsandsynlighedsfordeling,		polyCoeff()	137
normCdf()	128	polyDegree()	138
normCdf()	128	polyEval(), beregne polynomium	138
normPdf()	128	polyGcd()	138-139
not, Boolsk operator	129	polynomier	
nPr(), permutationer	129	beregne, polyEval()	138
npv(), nettonutidsværdi	130	vilkårlige, randPoly()	149
nSolve(), numerisk løsning	130	PolyRoots()	140
n-te rod		polær	
skabelon til	1	koordinat, R►Pr()	147
nulpunkter, zeroes()	207	koordinat, R►Pθ()	146
numerisk		vektorvisning, ►Polar	136
differentialkvotient, nDerivative()	124	potens, ^	219
differentialkvotient, nDeriv()	125-126		
integral, nInt()	126		

PotensReg, potensregression	140	randNorm(), vilkårlig norm	148
potensregression, PowerReg	140, 153-154, 191	randPoly(), vilkårligt polynomium ..	149
Prgm, definere program	141	randSamp()	149
primaltest, isPrime()	97	RandSeed, vilkårligt tal-seed	149
procent, %	222	real(), reel	149
prodSeq()	142	real, reel()	149
product(), produkt	142	reciprok, $\wedge^{-1}$	238
produkt $\prod()$ skabelon til	5	reduceret række-echelonform, rref()	160
produkt, $\prod()$	230	ref(), række-echelonform	150
produkt, product()	142	RefreshProbeVars	152
programmer definere privat bibliotek	49	regressioner andengrads, QuadReg	144
programmer og programmering slette fejl, ClrErr	26	eksponentiel, ExpReg	68
vis I/O-skærm, Disp	54	fjerdegrads, QuartReg	145
vis I/O-skærm, disp.	162	lineær regression, LinRegAx	102
programmering definere program, Prgm	141	lineær regression, LinRegBx	101
videresende fejl, PassErr	135	logistiske, Logistik	112
vis data, Disp	54, 162	potensregression, PowerReg	140, 153-154, 191
propFrac, ægte brøk	142	sinus, SinReg	174
punktum addition, .+	220	tredjegrads, CubicReg, CubicReg	42
division, ./	221	Regressioner lineær regression, LinRegBx	103
multiplikation, .*	221	logaritmisk, LnReg	107
potens, . $\wedge$	221	Logistisk	111
subtraktion, .-	220	medium-medium linje, MedMed	116
		MultReg	120
		rektangulær vektorvisning, $\blacktriangleright$ Rect ..	150
		rektangulært x-koordinat, $\blacktriangleright$ Rx() ..	134
		rektangulært y-koordinat, $\blacktriangleright$ Ry() ..	134
		remain(), rest	153
		RequestStr	154
		rest, remain()	153
		resultat vis i termer af cosinus	31
		viser i termer af e	65
		viser ud fra sinus	171
		resultat (sidste), ans	13
		resultater, statistiske	181
		resultatværdier, statistik	182
		Return, returner	155
		returner, Return	155
		right, right()	28, 62, 205
		rk23(), Runge Kutta-funktion	156

Q

QR opløse i faktorer, QR	143
QuadReg, andengradsregression ...	144
QuartReg, fjerdegradsregression ...	145

R

R, radian	234
R $\blacktriangleright$ Pr(), polær koordinat	147
R $\blacktriangleright$ P θ (), polær koordinat	146
radian, R	234
rand(), vilkårligt tal	147
randBin, vilkårligt tal	147
randInt(), vilkårligt heltal	148
randMat(), vilkårlig matrix	148

270 Indeks

rotate(), roter	157-158
roter, rotate()	157-158
round(), afrund	159
rowAdd(), addition af matrixrækker	159
rowDim(), matrixrækkedimension	159
rowNorm(), matrixrækkenorm	160
rowSwap(), matrixrækkeskift	160
rref(), reduceret række-echelonform	160
rund op, ceiling()	21
runde ned, floor()	73
runde op, ceiling()	22, 37
række-echelonform, ref()	150
række, series()	165

S

sammenhørende ligninger, simult()	170
sandsynlighedsdensitivitet, normPdf ()	128
sec ⁻¹ (), invers sekans	161
sec(), sekans	161
sech ⁻¹ (), invers hyperbolsk sekans	162
sech(), hyperbolsk sekans	161
sekundnotation, "	235
sekvens, seq()	163
seq(), sekvens	163
seqGen()	163
seqn()	164
sequence, seq()	163-164
series(), række	165
setMode(), indstil tilstand	166
shift(), flyt	168
sign(), fortegn	169
simult(), sammenhørende ligninger	170
sin ⁻¹ (), arcussinus	172
sin(), sinus	171
sinh ⁻¹ (), hyperbolsk arcussinus	173
sinh(), hyperbolsk sinus	173
SinReg, sinusregression	174
sinus	
viser udtryk ud fra	171
sinus, sin()	171
sinusregression, SinReg	174

skabeloner	
Absolut værdi	3-4
anden afledede	6
bestemt integral	6
brøk	1
differentialkvotient eller differentialkvotienten af nte orden	6
eksponent	2
eksponent	1
første orden	5
grænseværdi	7
kvadratrods	1
ligningssystem (2-ligninger)	3
ligningssystem (N-ligning)	3
Log	2
matrix (1 × 2)	4
matrix (2 × 1)	4
matrix (2 × 2)	4
matrix (m × n)	4
nte rod	1
produkt $\prod$ ()	5
stykkevis-skabelon (N stykker) ..	3
stykkevis funktion (2 stykker) ..	2
sum Σ ()	5
ubestemt integral	6
slet	
fej, ClrErr	26
slette	
ugyldige elementer fra liste	50
variabel, DelVar	50
slut	
funktion, EndFunc	79
hvis, EndIf	90
mens, EndWhile	206
slut på hvis, EndIf	90
slut while-kompleks, EndWhile	206
solve(), løs	175
som af rentebetaling	231
SortA, sorter stigende	178
SortD, sortere faldende	179
sortere	
faldende, SortD	179
stigende, SortA	178

sprog		
hent sproginformation	85	
sqrt(), kvadratrod	180	
standardafvigelse, stdDev() ...	182-183, 203	
stat.results	181	
stat.values	182	
statistik		
fakultet, !	227	
kombinationer, nCr()	124	
median, median()	115	
middel, mean()	115	
permutationer, nPr()	129	
standardafvigelse, stdDev() ..	182-183, 203	
statistik med en variabel, OneVar	131	
to-variabel-resultater, TwoVar .	200	
varians, variance()	203	
vilkaarlig norm, randNorm()	148	
vilkaarligt tal-seed, RandSeed ...	149	
statistik med en variabel, OneVar ...	131	
stdDevPop(), population		
standardafvigelse	182	
stdDevSamp(), standardafvigelse for		
måling	183	
Stop-kommando	184	
streng		
dimension, dim()	54	
længde	54	
streng		
anvende til at skabe		
variabelnavne	248	
flyt, shift()	168	
formatere	75	
formatere, format()	75	
henvisning, #	233	
højre, right()	94, 156	
i, inString	93	
midt-streng, mid()	117	
roter, rotate()	157-158	
streng til udtryk, expr()	68, 110	
tegnkode, ord()	134	
tegnstreng, char()	23	
tilføj, &	227	
udtryk til streng, string()	184	
venstre, left()	98	
string(), udtryk til streng	184	
strings		
right, right()	28, 62, 205	
student-t sandsynlighedsfordeling,		
tCdf()	190	
student-t sandsynlighedstæthed,		
tPdf()	195	
stykkevis-skabelon (N stykker)		
skabelon til	3	
stykkevis funktion (2 stykker)		
skabelon til	2	
større end eller lig med, 	225	
større end, >	225	
største fælles divisor, gcd()	79	
subMat(), delmatrix	184, 186	
substitution med " " -operator	239	
subtrahere, -	216	
sum $\Sigma()$		
skabelon til	5	
sum af hovedstolsbetalinger	232	
sum(), summation	185	
sum, S()	231	
sumIf()	185	
summation, sum()	185	
sumSeq()	186	
T		
T (transponere)	186	
t test, tTest	197	
tan ⁻¹ (), arcustangens	187	
tan(), tangens	187	
tangens, tan()	187	
tangent linje, tangentLine()	188	
tangentLine()	188	
tanh ⁻¹ (), hyperbolsk arcustangens .	189	
tanh(), hyperbolsk tangens	188	
tastaturgenveje	245	
Taylor-polynomium, taylor()	190	
taylor(), Taylor-polynomium	190	
tCdf(), student t		
sandsynlighedsfordeling ...	190	
tCollect(), trigonometrisk opsamling	190	
tegn		
numerisk kode, ord()	134	
streng, char()	23	

tegnstreng, char()	23	tælle dage mellem datoer, dbd()	46
test for ugyldighed, isVoid()	97	tælle elementer i en liste betinget, countif()	36
Test_2S, F test med 2 målinger	78	tælle elementer i en liste, count()	36
tExpand(), trigonometrisk ekspansion	191		
Text-kommando	191	U	
tidsdiskonterede pengeværdier, aktuel værdi	200	ubestemt integral skabelon til	6
tidsdiskonterede pengeværdier, antal betalinger	199	udelukkelse med " " -operator	239
tidsdiskonterede pengeværdier, betalingsbeløb	199	udtryk	
tidsdiskonterede pengeværdier, Fremtidig værdi	198	streng til udtryk, expr()	68, 110
tidsdiskonterede pengeværdier, Rente	199	udtryk til liste, exp►list()	66
tiende potens, 10^()	238	udvid/sammenkæd, augment()	16
tilføj, &	227	ugyldige elementer	243
tilnærmet, approx()	14	ugyldige elementer, fjerne	50
tilstande		ugyldighed, test for	97
indstille, setMode()	166	understregning, _	237
tilstandsindstillinger, getMode()	86	unitV(), enhedsvektor	202
tInterval, t konfidensinterval	192	unLock, oplåse variabel eller variabel gruppe	203
tInterval_2Samp, tomåligners t konfidensinterval	193		
tmpCnv()	194	V	
to-variabel-resultater, TwoVar	200	variabel	
tomme (ugyldige) elementer	243	oprette navn fra en tegnstreng	248
tPdf(), studentt sandsynlighedstæthed	195	slette alle enkeltbogstaver	26
transponere, T	186	variable	
tredjegradsregression, CubicReg	42	lokal, Local	109
trigonometrisk ekspansion, tExpand()	191	slette, DelVar	50
trigonometrisk opsamling, tCollect()	190	variable og funktioner	
Try, fejlhåndteringskommando	196	kopiere	30
tTest, t test	197	variable, låse og låse op	86, 109, 203
tTest_2Samp, to-måligners t test	197	varians, variance()	203
TVM-argumenter	200	varPop()	203
tvmFV()	198	varSamp(), målingsvarians	203
tvmI()	199	vektorer	
tvmN()	199	cylindrisk vektorvisning, ►Cylind	44
tvmPmt()	199	enhed, unitV()	202
tvmPV()	200	prikprodukt, dotP()	59
TwoVar, to-variabel-resultater	200	vektorprodukt, crossP()	38
		vektorprodukt, crossP()	38
		venstre, left()	98
		Vent på kommando	204
		videresende fejl, PassErr	135

vilkårlig		zTest_2Prop, toproportional z test .	213
matrix, randMat()	148	zTest_2Samp, to-prøvers z test	214
vilkårlig norm, randNorm()	148		
vilkårlig stikprøve	149	Æ	
vilkårligt		ægte brøk, propFrac	142
tal-seed, RandSeed	149		
vinkel, angle()	10	Δ	
vis data, Disp	54, 162	Δlist(), listedifferens	106
vis som			
rektangulær vektor, ►Rect	150	X	
vise som		χ ² Cdf()	25
binær, ►Base2	18	χ ² GOF	25
cylindrisk vektor, ►Cylind	44	χ ² Pdf()	26
decimal vinkel, ►DD	47		
decimalt heltal, ►Base10	19		
grader/minutter/sekunder,			
►DMS	56		
hexadecimal, ►Base16	20		
kuglevektor, ►Sphere	179		
polær vektor, ►Polar	136		
Viser grader/minutter/sekunder,			
►DMS	56		
W			
warnCodes(), Warning codes	205		
when(), når	205		
While, mens	206		
X			
x ² , kvadrat	220		
XNOR	226		
xor, Boolsk exclusive or	206		
Z			
zeroes(), nulpunkter	207		
zInterval, z konfidensinterval	210		
zInterval_1Prop, en-proportionalt z			
konfidensinterval	210		
zInterval_2Prop, to-proportionalt z			
konfidensinterval	211		
zInterval_2Samp, z konfidensinterval			
med to-målinger	211		
zTest	212		
zTest_1Prop, enproportional z test	213		