

TI-Nspire™ CX CAS

Opslagsvejledning

Vigtige oplysninger

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

© 2024 Texas Instruments Incorporated

De faktiske produkter kan variere let fra de viste billeder.

Indholdsfortegnelse

Udtryksskabeloner	1
Alfabetisk oversigt	6
A	6
B	14
C	18
D	33
E	42
F	50
G	57
I	67
L	74
M	89
N	99
O	106
P	109
Q	115
R	119
S	133
T	151
U	163
V	164
W	165
X	167
Z	168
Symboler	174
TI-Nspire™ CX II - Tegn kommandoer	193
Grafikprogrammering	193
Grafikskærm	193
Standardvisning og indstillinger	194
Fejlmeddelelser på grafikskærmen	195
Ugyldige kommandoer i grafiktilstand	195
C	196
D	197
F	200
G	202
P	203
S	205
U	207

Tomme (ugyldige) elementer	208
Genveje til indtastning af matematiske udtryk	210
Hierarkiet i EOS™ ligningsoperativsystemet (Equation Operating System)	212
TI-Nspire CX II - TI-Grundlæggende programmeringsfunktioner	214
Automatisk indrykning i programmeringseditoren	214
Forbedrede fejlmeddelelser til TI-Basic	214
Konstanter og værdier	217
Fejlkoder og fejlmeddelelser	218
Fejlkoder- og meddelelser	226
Generelle oplysninger	228
Indeks	229

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 175.

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 175.

Eksempel:

$$2^3 = 8$$

Kvadratrodsskabelon

ctrl **x²** -taster



Bemærk: Se også $\sqrt{}$ (kvadratrod), side 183.

Eksempel:

Nte rod-skabelon

ctrl **^** -taster



Bemærk: Se også root(), side 130.

Eksempel:



e eksponentskabelon

e^x -taster



Eksempel:

Den naturlige eksponentialfunktion e opløftet til en potens

Bemærk: Se også $e^{\wedge}()$, side 42.

Log-skabelon

ctrl 10^x -tasten



Eksempel:

$$\log_{10} \left(\frac{2}{4} \right) = 0.5$$

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

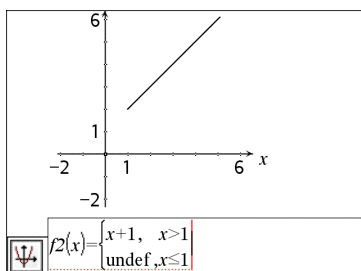
Bemærk: Se også $\log()$, side 85.

Stykkevis-skabelon (2 stykker)

Katalog >



Eksempel:



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 110.

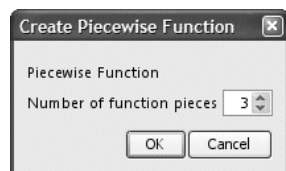
Stykkevis-skabelon (N stykker)

Katalog >

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

Eksempel:

Se eksemplet med stykkevis-skabelonen (2 stykker).



Stykkevis-skabelon (N stykker)

Katalog > 

Bemærk: Se også `piecewise()`, side 110.

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 151.

Eksempel:

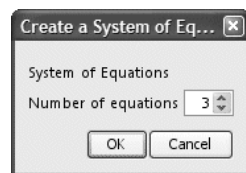
$$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=5 \end{cases}, x, y\right) \quad x=\frac{5}{2} \text{ and } y=-\frac{5}{2}$$

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

Skabelon til system med N ligninger

Katalog > 

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



Eksempel:

Se eksemplet med ligningssystemskabelonen (2-ligninger).

Bemærk: Se også `system()`, side 151.

Absolut værdi-skabelon

Katalog > 



Bemærk: Se også `abs()`, side 6.

Eksempel:

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

dd°mm'ss.ss''-skabelon

Katalog > 



Eksempel:

dd°mm'ss.ss"-skabelon

Katalog > 

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

Matrix-skabelon (2 x 2)

Katalog > 



Eksempel:

Opretter en matrix 2 x 2.

Matrix-skabelon (1 x 2)

Katalog > 



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 > 



Eksempel:

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

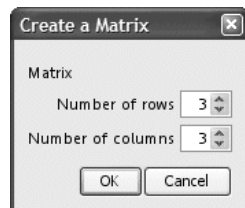
Matrix-skabelon (m x n)

Katalog > 

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

Eksempel:

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



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

Sum-skabelon (Σ)

Katalog > 

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

Eksempel:

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

Bemærk: Se også $\Sigma()$ (sumSeq), side 184.

Produkt-skabelon (Π)

Katalog > 

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

Eksempel:

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

Bemærk: Se også $\Pi()$ (prodSeq), side 183.

Skabelon til differentialkvotient af første orden

Katalog > 

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

Eksempel:

Bemærk: Se også $d()$ (differentialkvotient), side 183.

Skabelon til differentialkvotient af anden orden

Katalog > 

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

Eksempel:

Bemærk: Se også $d()$ (differentialkvotient), side 183.

Bestemt integral skabelon

Katalog > 

$$\int_a^b () dx$$

Eksempel:

Alfabetisk oversigt

Elementer, hvis navne ikke er alfabetiske (som f.eks. +, ! og >), er anført sidst i dette afsnit, startende (side 174). 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(*ListeI*)⇒*liste*

abs(*MatrixI*)⇒*matrix*

Returnerer den absolutte værdi af argumentet.

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

Hvis argumentet er et komplekst tal, returneres tallets modulus.

Bemærk: Alle udefinerede variable behandles som reelle variable.

amortTbl()

Katalog > 

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

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 161).

amortTbl(12,60,10,5000,,,12,12)				
0	0.	0.	5000.	
1	-41.67	-64.57	4935.43	
2	-41.13	-65.11	4870.32	
3	-40.59	-65.65	4804.67	
4	-40.04	-66.2	4738.47	
5	-39.49	-66.75	4671.72	
6	-38.93	-67.31	4604.41	
7	-38.37	-67.87	4536.54	
8	-37.8	-68.44	4468.1	
9	-37.23	-69.01	4399.09	
10	-36.66	-69.58	4329.51	
11	-36.08	-70.16	4259.35	
12	-35.49	-70.75	4188.6	

- Hvis du 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 $\Sigma\text{Int}()$ og $\Sigma\text{Prn}()$, side 184 og **bal()**, side 14.

and

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

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

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

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

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

Heltal1 and Heltal2 $\Rightarrow$ *heltal*

I hexadecimal tilstand:

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

Vigtigt: Tallet nul, ikke bogstavet O.

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 binær tilstand:

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

I decimal tilstand:

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

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()

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

I vinkeltilstanden Grader:

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

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

$\text{angle}(\text{Matrix1}) \Rightarrow \text{matrix}$

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 146.)

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 146.)

LevRow=0 for Blok

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

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

Output-variabel	Beskrivelse
stat.SS	Kvadratsum for kolonnefaktoren
stat.MS	Middelkvadrat for kolonnefaktoren
stat.FBlok	F statistik for faktor
stat.PValBlok	Mindste sandsynlighed, ved hvilken nul-hypotesen kan forkastes
stat.dfBlok	Frihedsgrader for faktoren
stat.SSBlok	Kvadratsum for faktoren
stat.MSBlok	Middelkvadrat for faktoren
stat.dfError	Frihedsgrader for fejl
stat.SSError	Kvadratsum for fejlene
stat.MSError	Middelkvadrat for fejlene
stat.s	Standardafvigelse for fejlen

KOLONNEFAKTOR Output

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

RÆKKEFAKTOR Output

Output-variabel	Beskrivelse
stat.Frow	F statistik for rækkefaktoren
stat.PValRow	Sandsynlighedsværdi for rækkefaktoren
stat.dfRow	Frihedsgrader for rækkefaktoren
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⇒ <i>værdi</i>	56		56
Returnerer resultatet af de sidst beregnede udtryk.	56+4		60
	60+4		64

approx()	Katalog > 
Returnerer beregningen af argumentet som et udtryk med decimale værdier, når det er muligt, uanset den aktuelle indstilling af Auto eller tilnærmet .	$\text{approx}\left(\frac{1}{3}\right)$ 0.333333
Dette svarer til at indtaste argumentet og trykke på   .	$\text{approx}\left(\left\{\frac{1}{3}, \frac{1}{9}\right\}\right)$ {0.333333,0.111111}
	$\text{approx}(\{\sin(\pi), \cos(\pi)\})$ {0,-1.}
	$\text{approx}(\left[\sqrt{2} \quad \sqrt{3}\right])$ [1.41421 1.73205]
	$\text{approx}\left(\left[\frac{1}{3} \quad \frac{1}{9}\right]\right)$ [0.333333 0.111111]
approx(List1)⇒liste	$\text{approx}(\{\sin(\pi), \cos(\pi)\})$ {0,-1.}
approx(Matrix1)⇒matrix	$\text{approx}(\left[\sqrt{2} \quad \sqrt{3}\right])$ [1.41421 1.73205]
Returnerer en liste eller <i>matrix</i> , hvor hvert element er beregnet til en decimalværdi, hvor det er muligt.	

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

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

Matrix ►**approxFraction**([*Tol*])⇒*matrix*

$$0.8333333333333333 \text{ ► } \text{approxFraction}(5.E-14)$$

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

$$\frac{5}{6}$$

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

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

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

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

Matrix[, *Tol*])⇒*matrix*

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

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

arccos()**Se $\cos^{-1}()$, side 26.****arccosh()****Se $\cosh^{-1}()$, side 27.****arccot()****Se $\cot^{-1}()$, side 27.****arccoth()****Se $\coth^{-1}()$, side 28.****arccsc()****Se $\csc^{-1}()$, side 30.**

arccsch() Se $\text{csch}^{-1}()$, side 31.

arcsec() Se $\text{sec}^{-1}()$, side 133.

arcsech() Se $\text{sech}^{-1}()$, side 134.

arcsin() Se $\sin^{-1}()$, side 141.

arcsinh() Se $\sinh^{-1}()$, side 142.

arctan() Se $\tan^{-1}()$, side 152.

arctanh() Se $\tanh^{-1}()$, side 153.

augment() [Katalog >](#) 

augment(Liste1, Liste2) ⇒ liste

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

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

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

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

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

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

Returnerer den fremadrettede differenskvotient (gennemsnitlig ændringshastighed).

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

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

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

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

B

bal()

bal(*NPmt*, *N*, *I*, *PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*afrundVærdi*])⇒*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 161).

NPmt angiver betalingsnummeret, hvorefter du vil have dataene beregnet.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter (side 161).

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

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

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

bal($NPmt, amortTabel$) beregner saldoen efter betaling nummer $NPmt$, baseret på amortiseringstabel $amortTabel$.
 $amortTabel$ -argumentet skal være en matrix i formen beskrevet under **amortTbl()**, side 6.

Bemærk: Se også **ΣInt()** og **ΣPrn()**, side 184.

►Base2

Heltal ►Base2 ⇒ *heltal*

256 ►Base2

0b100000000

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

0h1F ►Base2

0b11111

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

0b *binærtTal*

0h *hexadecimalTal*

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

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

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

–1 vises som

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

–2⁶³ vises som

0h8000000000000000 i det hexadecimale talsystem
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.

2⁶³ bliver –2⁶³ og vises som

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

2⁶⁴ til 0 og vises som

0h0 på hexadecimal form

0b0 på binær form.

–2⁶³ – 1 bliver 2⁶³ – 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 hexadecimale indtastninger skal altid have hhv. 0b eller 0h som præfiks.

0b *binærtTal*

0h *hexadecimaltTal*

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

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

Uden præfiks behandles *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 hexadecimale tal har altid henholdsvis 0b eller 0h som præfiks.

0b *binærtTal*

0h *hexadecimaltTal*

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

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

Uden præfiks behandles *Heltal1* som decimaltal (10-talssystem). Resultatet vises som hexadecimalt uanset tilstanden for talsystem.

256►Base16	0h100
------------	-------

0b111100001111►Base16	0hF0F
-----------------------	-------

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 15.

binomCdf()

binomCdf(*n*,*p*)⇒*liste*

binomCdf

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

binomCdf(*n*,*p*,*øvreGrænse*)for $P(0 \leq X \leq \textit{øvreGrænse})$ ⇒*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 $P(X \leq \textit{øvreGrænse})$, sæt *nedreGrænse*=0

binomPdf()

binomPdf(*n*,*p*)⇒*liste*

binomPdf(*n*,*p*,*XVærdi*)⇒*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()**

Returnerer det nærmeste heltal, der er ≥ argumentet.

<code>ceiling(.456)</code>	1.
----------------------------	----

Argumentet kan være et reelt eller komplekst tal.

ceiling()

Katalog > 

Bemærk: Se også `floor()`.

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

ceiling(*Matrix1*) $\Rightarrow$ *matrix*

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

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

centralDiff()

Katalog > 

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

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

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

centralDiff(*1*, *Var* [= *Værdi*][, *Trin*]) $\Rightarrow$ *liste*

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

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

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

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

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

Bemærk: Se også .

char()

Katalog > 

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

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

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

χ^2 2way obsMatrix

chi22way obsMatrix

Beregner en χ^2 test til association på tovejstabelen med tællinger i den observerede matrix *obsMatrix*. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 146.)

Oplysninger om effekten af tomme elementer i en matrix findes "Tomme (ugyldige) elementer," side 208.

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

 χ^2 Cdf()

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

$\text{chi2Cdf}(\text{nedreGrænse}, \text{øvreGrænse}, df) \Rightarrow \text{tal}$
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 208.

χ^2 GOF *obsListe,forvListe,fg*chi2GOF *obsListe,forvListe,fg*

Udfører en test for at bekræfte, at måledataene er fra en population, der er i overensstemmelse med en angivet distribution. *obsList* er en liste med antal, og skal indeholde heltal. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

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.Complst	Bidrag til chi-kvadrat elementbidrag

 χ^2 Pdf()

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

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

Beregner tæthedsfunktionen (pdf) for 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 208.

ClearAZ

ClearAZ

Sletter alle enkelttegnvariable i det aktuelle opgaverum.

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock**, side 163.

ClrErr

Katalog

ClrErr

Se et eksempel på **ClrErr**, i Eksempel 2 under **Try**-kommandoen, side 157.

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 110, og **Try**, side 157.

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*.

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

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

conj()

Katalog > 

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

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

Returnerer kompleks konjugerede af argumentet.

Bemærk: Alle udefinerede variable behandles som reelle variable.

constructMat()

Katalog > 

constructMat

(
Udtr, *Var1*, *Var2*, *antalRækker*, *antalKol*)
 $\Rightarrow$ *matrix*

Returnerer en matrix baseret på argumenter.

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

Udtr er et udtryk i variablerne *Var1* og *Var2*. Elementer i den resulterende matrix er dannet ved beregning af *Udtr* for hver forøget værdi af *Var1* og *Var2*.

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

CopyVar *Var1, Var2***CopyVar** *Var1., Var2.*

CopyVar *Var1, Var2* kopierer værdien af variabelen *Var1* til variabelen *Var2*, og opretter *Var2* hvis nødvendigt. Variablen *Var1* skal have en værdi

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.

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

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

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

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

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

cos()**cos**(*Liste1*)⇒*list*

I vinkeltilstanden Grader:

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

I vinkeltilstanden Nygrader:

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 Radian:

cos(kvadratMatrix1)⇒kvadratMatrix

I vinkeltilstanden Radian:

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

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

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

Beregn egenværdierne (λ_i) og egenvektorer (V_i) 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.

$\cos^{-1}()$



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

I vinkeltilstanden Grader:

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

I vinkeltilstanden Nygrader:

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 **arccos (...)**.

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

I vinkeltilstanden radian og rektangulært komplekst format:

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()**.

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

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.

$\cosh()$

Katalog >

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

I vinkeltilstanden Grader:

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

$\cosh(\text{kvadratMatrix1}) \Rightarrow \text{kvadratMatrix}$

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()**.

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

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

cosh⁻¹(Liste1) ⇒ liste

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:

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

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

cot()

 -tast

cot(Liste1) ⇒ liste

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

I vinkeltilstanden Grader:

$$\cot(45) = 1$$

I vinkeltilstanden Nygrader:

$$\cot(50) = 1$$

I vinkeltilstanden Radian:

cot⁻¹() -tastcot⁻¹(Liste1) ⇒ liste

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

I vinkeltilstanden Grader:

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

$\cot^{-1}()$

-tast

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

`coth()`

Katalog > 

`coth(Liste1)⇒liste`

$\coth^{-1}()$

Katalog > 

`coth-1(Liste1)⇒liste`

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

`count()`

Katalog > 

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

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

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

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

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

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208.

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

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

Kriterie kan være:

- En værdi, et udtryk eller en streng. For eksempel tæller **3** kun de elementer i *Liste*, der reduceres til værdien 3.
- 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 208.

Bemærk: Se også **sumIf()**, side 150, og **frequency()**, side 55.

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

Tæller antallet af elementer lig med 3.

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

Tæller antallet af elementer lig med "def."

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

Tæller 1 og 3.

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

Tæller 3, 5 og 7.

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

Tæller 1, 3, 7 og 9.

cPolyRoots()Katalog > **cPolyRoots(Poly, Var) ⇒ liste****cPolyRoots(ListeAfKoeff) ⇒ liste**

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

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

Bemærk: Se også **polyRoots()**, side 112.

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

crossP()

Katalog > 

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.

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

csc()

 -tast

csc(Liste1)⇒liste

I vinkeltilstanden Grader:

I vinkeltilstanden Nygrader:

I vinkeltilstanden Radian:

csc⁻¹()

 -tast

csc⁻¹(Liste1) ⇒ liste

I vinkeltilstanden Grader:

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

`csc-1(1)` 90.

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

I vinkeltilstanden Nygrader:

`csc-1(1)` 100.

I vinkeltilstanden Radian:

csch()

Katalog > 

csch(Liste1) ⇒ liste

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

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

CubicReg

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

Beregner polynomiell tredjegradsregression $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$ på listerne *X* og *Y* med frekvens *Frekv*. En sammenfatning af resultaterne lagres i *stat.resultat* variable. (side 146.)

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 208.

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

Output-variabel	Beskrivelse
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(ListeI)⇒liste

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

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

cumulativeSum(MatrixI)⇒matrix

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

<table> <tr><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td></tr> </table>	1	2	3	4	5	6	→ <i>mI</i>	<table> <tr><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td></tr> </table>	1	2	3	4	5	6
1	2													
3	4													
5	6													
1	2													
3	4													
5	6													
cumulativeSum(<i>mI</i>)		<table> <tr><td>1</td><td>2</td></tr> <tr><td>4</td><td>6</td></tr> <tr><td>9</td><td>12</td></tr> </table>	1	2	4	6	9	12						
1	2													
4	6													
9	12													

Et tomt (ugyldigt) element i *ListeI* eller *MatrixI* giver et ugyldigt element i den resulterende liste eller matrix. Yderligere oplysninger om tomme elementer findes på side 208.

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**).

Cycle	Katalog > 	
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

Katalog > 

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.

D

dbd()	Katalog > 	
dbd(dato1,dato2)⇒værdi	dbd(12.3103,1.0104)	1
Returnerer antallet af dage mellem <i>dato1</i> og <i>dato2</i> med tælling af faktiske dage.	dbd(1.0107,6.0107)	151
	dbd(3112.03,101.04)	1
	dbd(101.07,106.07)	151
<i>dato1</i> og <i>dato2</i> kan være tal eller lister med tal inden for området af datoer i en standardkalender. Hvis både <i>dato1</i> og <i>dato2</i> er lister, skal de have samme længde.		
<i>dato1</i> og <i>dato2</i> 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)

DDMM.ÅÅ (almindeligt format i Europa)

►DD

Katalog >

Tal ►DD⇒værdi

I vinkeltilstanden Grader:

Liste l ►DD⇒liste

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

Matrix l ►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.

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

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

►Decimal

Katalog >

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

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

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

Define

Katalog >

Define Var = Udtryk

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

Definerer variablen Var eller den brugerdefinerede funktion Funktion.

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

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 en 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 36 og **Define LibPub**, side 36.

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

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

Define LibPriv *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 34, og **Define LibPub**, side 36.

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 34 og **Define LibPriv**, side 36.

DelVar

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

DelVar *Var*.

Sletter de angivne variable, eller variabelgruppe fra hukommelse.

Hvis en eller flere af variablene er låst, viser denne kommando en fejlmeddelelse og sletter kun de ulåste variable. Se **unLock**, side 163

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.

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

delVoid()

delVoid(*Liste1*)⇒*liste*

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

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

Yderligere oplysninger om tomme elementer findes på side 208.

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:

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

diag()

diag(List) $\Rightarrow$ matrix

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

diag(rækkeMatrix) $\Rightarrow$ matrix

diag(kolonneMatrix) $\Rightarrow$ matrix

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

diag(kvadratMatrix) $\Rightarrow$ rækkeMatrix

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

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

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

kvadratMatrix skal være kvadratisk.

dim()

dim(Liste) $\Rightarrow$ heltal

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

Returnerer dimensionen af *liste*.

dim(matrix) $\Rightarrow$ liste

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

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

dim()

Katalog > 

dim(*Streng*)⇒*heltal*

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

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

Disp

Katalog > 

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

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

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

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

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

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

DispAt

Katalog > 

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.

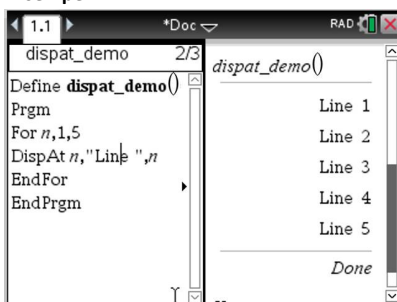
Vær opmærksom på, at linjenummeret ikke gælder hele skærmen, men for området umiddelbart efter kommando/program.

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

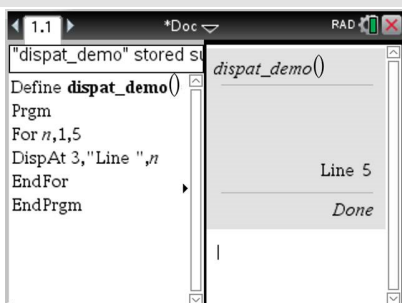
DispAtog Disp kan bruges indenfor det samme program.

DispAt

Eksempel



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



Konkrete eksempler:

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

Fejlbetingelser:

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

►DMS

Katalog >

List ►DMS

I vinkeltilstanden Grader:

Matrix ►DMS

$(45.371) \blacktriangleright \text{DMS}$	$45^{\circ}22'15.6''$
$\{\{45.371,60\}\} \blacktriangleright \text{DMS}$	$\{45^{\circ}22'15.6'',60^{\circ}\}$

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

Tolker argumentet som en vinkel og viser tilsvarende tal for grader (D), minutter (M) og sekunder (S/s) (DDDDDD°MM'SS.ss"). Se mere om DMS-formatet for grader, minutter og sekunder °, ', " på side 188.

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

dotP()

Katalog >

dotP(Liste1, Liste2)⇒udtryk

Returnerer "prik" produktet af to lister.

dotP(Vektor1, Vektor2)⇒udtryk

Returner "prik" produktet af to vektorer.

Begge skal være rækkevektorer, eller begge skal være kolonnevektorer.

E

$e^{\wedge}()$

 -tast

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

Bemærk: At trykke   for at vise $e^{\wedge}$ (er ikke det samme som at trykke på tegnet  på tastaturet.

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

$e^{\wedge}(\text{Liste1}) \Rightarrow \text{liste}$

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

$e^{\wedge}(\text{kvadratMatrix1}) \Rightarrow \text{kvadratMatrix}$

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.

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

eff()

Katalog > 

$\text{eff}(\text{nominelRente}, \text{CpY}) \Rightarrow \text{værdi}$

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

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 102.

eigVc()

`eigVc(kvadratMatrix)⇒matrix`

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

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

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.

I rektangulært komplekst format:

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

$$\text{eigVc}(m1) \begin{bmatrix} -0.800906 & 0.767947 & 0.573804+0.052258\cdot i & 0.573804-0.052258\cdot i \\ 0.484029 & 0.262687+0.096286\cdot i & 0.262687-0.096286\cdot i & 0.262687+0.096286\cdot i \end{bmatrix}$$

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

eigVl()

`eigVl(kvadratMatrix)⇒liste`

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

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

I rektangulær kompleks formattilstand:

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

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

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

Else

Se If, side 67.

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

Return 5

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

Return 3

EndIf

EndFunc

*Done***EndFor****Se For, side 52.****EndFunc****Se Func, side 56.****EndIf****Se If, side 67.****EndLoop****Se Loop, side 88.****EndPrgm****Se Prgm, side 114.****EndTry****Se Try, side 157.**

euler ()

Katalog > 

euler(Expr, Var, depVar, {Var0, VarMax}, depVar0, VarStep [, eulerStep]) $\Rightarrow$ matrix

euler(SystemOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep [, eulerStep]) $\Rightarrow$ matrix

euler(ListOfExpr, Var, ListOfDepVars, {Var0, VarMax}, ListOfDepVars0, VarStep [, eulerStep]) $\Rightarrow$ matrix

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 differentialligning (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.

Differentialligning:

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

$$\text{euler}(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1) \rightarrow \begin{bmatrix} 0. & 1. & 2. & 3. & 4. \\ 10. & 10.9 & 11.8712 & 12.9174 & 14.042 \end{bmatrix}$$

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

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 $y2(0) = 5$

$$\text{euler}\left(\begin{cases} -y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right) \rightarrow \begin{bmatrix} 0. & 1. & 2. & 3. & 4. & 5. \\ 2. & 1. & 1. & 3. & 27. & 243. \\ 5. & 10. & 30. & 90. & 90. & -2070. \end{bmatrix}$$

$\{Var0, VarMax\}$ 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 $\text{sign}(VarStep) = \text{sign}(VarMax - Var0)$ og løsninger returneres i $Var0 + i \cdot VarStep$ for alle $i=0,1,2,\dots$ således at $Var0 + i \cdot VarStep$ er i $[var0, VarMax]$ (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 trinstørrelse, som euler-metoden anvender, er $VarStep / eulerStep$.

eval ()

Hub-menu

$\text{eval}(Expr) \Rightarrow string$

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.

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

Nulstil det blå element til OFF.

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

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

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

Programmet vil langsomt indføre det røde element

```

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

```

Kør programmet.

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

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 58), **GetStr** (side 64), og **Send** (side 134).

Exit

Katalog >

Exit

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

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

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

Funktionsliste:

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

exp()

-tast

Returnerer *e* opløftet til potensen *Udtr1*.

Returnerer *e* opløftet til potensen *Værdi1*.

exp()

-tast

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(Liste1) $\Rightarrow$ *liste*

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

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

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

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

expr()

Katalog > 

expr(Streng) $\Rightarrow$ *udtryk*

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

ExpReg

Katalog > 

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

Beregner polynomielle tredjegradsregression $y = a \cdot (b)^{x \text{ på listerne } X \text{ og } Y \text{ med hyppighed } Frekv}$. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146.)

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 208.

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 <i>X</i> -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 <i>Y</i> -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>

factor()Katalog > 

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 være længere end en dag, og opløsning af et 100-cifret tal kan være længere end et århundrede.

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

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

matrixVar skal eksistere i forvejen.

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

Erstatter hvert element i variablen *ListeVar* med *Udtr*.

ListeVar skal eksistere i forvejen.

$\{1,2,3,4,5\} \rightarrow alist$	$\{1,2,3,4,5\}$
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 146.)

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 208

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()

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

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

Argumentet kan være et reelt eller komplekst tal.

floor(Liste1) $\Rightarrow$ liste

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

floor(Matrix1) $\Rightarrow$ matrix

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

Returnerer en liste eller matrix med nedrunding af hvert element.

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

For

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.

Define $g()$ = Func	Done
Local <i>tempsum</i> , <i>step</i> , <i>i</i>	
$0 \rightarrow \text{tempsum}$	
$1 \rightarrow \text{step}$	
For <i>i</i> , 1, 100, <i>step</i>	
$\text{tempsum} + i \rightarrow \text{tempsum}$	
EndFor	
EndFunc	

$g()$	5050
-------	------

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.

format()

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

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

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

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

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

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

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

fPart()

fPart(*Udtr1*)⇒*udtryk*

<code>fPart(-1.234)</code>	-0.234
----------------------------	--------

fPart(*Liste1*)⇒*liste*

<code>fPart({1, -2.3, 7.003})</code>	{0, -0.3, 0.003}
--------------------------------------	------------------

fPart(*Matrix1*)⇒*matrix*

Returnerer decimaldelen af argumentet.

For en liste eller matrix returneres decimaldelen af elementerne.

Argumentet kan være et reelt eller komplekst tal.

$\text{FPdf}(XVærdi, dfTæller, dfNævner)$

$\text{FPdf}(XVærdi, dfTæller, dfNævner)$

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

freqTable►liste

$(Liste1, frekvHeltalListe) \Rightarrow liste$

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

Liste1 kan være enhver gyldig liste.

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

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

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208.

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

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 { $b(1) \leq b(2) < b(3) \leq b(4) < \dots < b(n-1) < b(n)$ }. 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*, $b(1) \leq b(2)$), countIf(*liste*, $b(1) < b(2)$), ..., countIf(*liste*, $b(n-1) < b(n)$), countIf(*liste*, $b(n) > ?$)}.

Elementer i *Liste1*, der ikke kan "placeres i en størrelse" ignoreres. Tomme (ugyldige) elementer ignoreres også. Yderligere oplysninger om tomme elementer findes på side 208.

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

Bemærk: Se også **countIf()**, side 29.

<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.

FTest_2Samp

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

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

(*Datalisteinput*)

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

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

(Sammenfatning, *stat input*)

Udfører en F test med to målinger. En sammenfatning af resultaterne lagres i variabelen *stat.results*. (side 146.)

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 208.

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

Func*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.

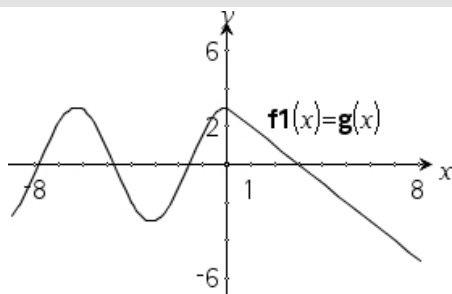
Definition af en stykvis funktion:

Define $g(x)$ =Func	Done
If $x < 0$ Then	
Return $3 \cdot \cos(x)$	
Else	
Return $3 - x$	
EndIf	
EndFunc	

Tegnet resultat af grafen $g(x)$

Bemærk indtastning af

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

**G****gcd()**

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 **tilnærmet**-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)⇒matrix

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

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

geomCdf()**geomCdf**

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

geomCdf(p,øvreGrænse)for $P(1 \leq X \leq \text{øvreGrænse}) \Rightarrow \text{tal}$ 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

Hub-menu

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

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

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

Der skal anmodes om værdien:

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

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

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

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

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

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

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 64 og **Send**, side 134.

getDenom()

Katalog >

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

getKey()

Katalog >

getKey([0|1]) ⇒ returnString

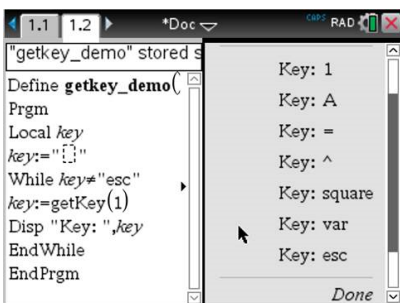
Beskrivelse: **getKey()** - tillader et TI-Basic-program at få tastaturinput - håndholdt, stationær og emuleret på stationær.

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

getKey()

Eksempel:



tast er trykket ned.

Håndtering af tasttryk:

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

Håndholdte enheder/emulatortast * (gangetast)	Skrivebord *	Returværdi **
e^x	n/a	"eksp"
10^x	n/a	"10potens"
+	+	"+"
-	-	"_"
(	(	"("
)	)	")"
.	.	". "
(-)	n/a	"-" (negativ fortegn)
Indtast	Indtast	"indtast"
ee	n/a	"E" (eksponentiel notation E)
a - z	a - z	alpha = bogstav trykket ned (lille bogstav) ("a" - "z")
skift a - z	skift a - z	alpha = bogstav trykket ned "A" - "Z"
		Bemærk: ctrl-shift fungerer som lock-caps
?!	n/a	"?!"
pi	n/a	"pi"
Flag	n/a	ingen retur
,	,	". "
Return	n/a	"retur"
mellemrum	mellemrum	" " (mellemrum)
Utilgængelig	Specielle tegntaster som @, !^, osv.	Tegnet returneres
n/a	Funktionstaster	Ingen returnerede tegn
n/a	Specielle skrivebordskontroltaster	Ingen returnerede tegn

Håndholdte enheder/emulatortast	Skrivebord	Returværði
Utilgængelig	Andre skrivebordstaster, som ikke er tilgængelige på beregneren, når getKey () venter på et tastetryk. {[,],,;, ...)	Samme tegn du får i noter (ikke i et matematik-felt)

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

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

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

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

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

getLangInfo()⇒*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*

a:=65

65

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

Lock *a*

Done

getLockInfo(*a*)

1

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

a:=75

"Error: Variable is locked."

DelVar *a*

"Error: Variable is locked."

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

Unlock *a*

Done

a:=75

75

DelVar *a*

Done

Se **Lock**, side 85, og **unLock**, side 163.

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.

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 136.

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=Gadian
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
Vektorformat	6	1=Rektangulær, 2=Cylindrisk, 3=Sfærisk
Talsystem	7	1=Decimal, 2=Hex, 3=Binær

getNum()

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

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 58 og **Send**, side 134.

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

(*Biblioteksnavnstreng*)returnerer en matrix med oplysninger for alle biblioteksobjekter, der er defineret i biblioteket *BibNavnStreng*.

BibNavnStreng skal være en streng (tekst omsluttet af citationstegn) eller en strengvariabel.

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

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

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

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

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

$a:=1$	1												
$b:=[1\ 2]$	$[1\ 2]$												
$c:=[1\ 3\ 7]$	$[1\ 3\ 7]$												
$vs:=\text{getVarInfo}()$	<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										
$vs[1]$	$[1\ \text{"NUM"}\ \text{"[]"}\ 0]$												
$vs[1,1]$	1												
$vs[2]$	"Error: Invalid list or matrix"												
$vs[2,1]$	$[1\ 2]$												

Goto

Goto etiketnavn

Overfører kontrol til etiketten *etiketnavn*.

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

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

Define $g()$ = 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

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:

$\{1.5\} \blacktriangleright \text{Grad}$	$\{1.66667\}^g$
---	-----------------

I vinkeltilstanden Radian:

$\{1.5\}$ ►Grad

$\{95.493\}^9$

I

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

If

Katalog > 

If *BooleanExpr*
Statement

If *BooleanExpr* **Then**
Block

EndIf

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

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

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

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

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

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

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

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

Block1 og *Block2* kan være en enkelt sætning.

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

:

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

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

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

Done

If $x < 0$ ThenReturn $\neg x$

Else

Return x

EndIf

EndFunc

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

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

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

Return 5

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

Return 3

EndIf

EndFunc

Done

$g(-4)$	4
---------	---

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

ifFn()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*.

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

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

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

Value_If_True-element på 6 kopieres til resultatlisten.

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

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

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

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

Value_If_False er ikke specificeret. Undef anvendes.

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

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

imag()

Returnerer imaginærdelen af argumentet.

imag(ListI) ⇒ *liste*

Returnerer en liste med imaginærdelen af elementerne.

imag(MatrixI) ⇒ *matrix*

Returnerer en matrix med imaginærdelene af elementerne.

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

Henvisning

Se #(), side 186.

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(*List1*) ⇒ *liste*

int(*Matrix1*) ⇒ *matrix*

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

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

Argumentet kan være et reelt eller komplekst tal.

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

intDiv()

Katalog > 

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

intDiv(*List1*, *List2*) ⇒ *liste*

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

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(*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 en 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.

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 ▲ og derefter bruge ◀ og ▶ til at bevæge markøren.

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

```
xvalueList:=seq(i,i,0,10,0.5)
{0,0.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,7.,7.5,8.,8.5,9.,9.5,10.}
xlist:=mat▶list(rk[1])
{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.}
ylist:=mat▶list(rk[2])
{5.,3.19499,5.00394,6.99957,9.00593,10.99706,12.99957,15.00394,17.00593,19.00706,21.00957}
yprimeList:=-3*y+6*t+5|y=ylist and t=xlist
{-10.,1.41503,1.98819,2.00129,1.98221,2.00679,2.01129,2.00129,1.98221,2.00679,2.01129}
interpolate(xvalueList,xlist,ylist,yprimeList)
{5.,2.67062,3.19499,4.02782,5.00394,6.00011,7.00011,8.00011,9.00011,10.00011,11.00011}
```

invχ²(*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(*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

(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()

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()

invNorm(Areal[, μ , σ])

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

invt(*Areal*,*df*)

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

iPart()

iPart(*Number*) $\Rightarrow$ *heltal*

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

iPart(*Matrix1*) $\Rightarrow$ *matrix*

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

Returnerer heltalsdelen af argumentet.

Returnerer heltalsdelen af hvert element for lister og matrixer.

Argumentet kan være et reelt eller komplekst tal.

irr()

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

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.

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

Bemærk: Se også **mirr()**, side 94.

isPrime()

Katalog > 

isPrime(*Number*) $\Rightarrow$ *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.

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

isPrime(5)	true
isPrime(6)	false

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

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

isVoid()

Katalog > 

isVoid(*Var*) $\Rightarrow$ *boolsk konstantudtryk*
isVoid(*Expr*) $\Rightarrow$ *boolsk konstantudtryk*
isVoid(*List*) $\Rightarrow$ *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 208.

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

L

Lbl

Katalog > 

Lbl *etiketNavn*

Definerer en etiket med navnet *etiketNavn* i en funktion.

Du kan anvende en **Goto** *etiketNavn*-kommando til at videregive kontrollen til kommandoen lige efter etiketten.

EtiketNavn skal opfylde de samme navngivningskrav som et variabelnavn.

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

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

lcm()

lcm(*Værdi1*, *Værdi2*) $\Rightarrow$ *udtryk*

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

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

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

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

Returnerer det mindste fælles multiplum af to argumenter. **lcm** af to brøker er **lcm** af deres tællere divideret med **gcd** af deres nævnere. **lcm** af brøker med flydende komma er deres produkt.

For to lister eller matricer returneres det mindste fælles multiplum af deres tilsvarende elementer.

left()

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

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

Returnerer *Antal*-tegn fra venstre i tegnstrengen *kildeStreng*.

Hvis du udelader *Antal*, returneres alle *kildeStreng*-variable.

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

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

Returnerer *Antal*-elementer til venstre i *Liste1*.

Hvis du udelader *Antal*, returneres hele *Liste1*.

left(*Sammenligning*) $\Rightarrow$ *udtryk*

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

Returnerer venstre af en ligning eller ulighed.

libShortcut(*BibNavneStreng*,
GenvejNavneStreng

[, *BibPrivFlag*]) ⇒ *liste med variable*

Opretter en variabelgruppe i den aktuelle opgave, som indeholder referencer til alle objekter i det specificerede biblioteksdokument *bibNavneStreng*. Tilføjer også 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 24.

For at slette en variabelgruppe, se **DelVar** på side 37.

Dette eksempel forudsætter et korrekt gemt og opdateret biblioteksdokument med navnet, **linalg2**, som indeholder objekter defineret som *clearmat*, *gauss1*, og *gauss2*.

```
getVarInfo("linalg2")
    [clearmat  "FUNC"  "LibPub "]
    [gauss1    "PRGM"  "LibPriv "]
    [gauss2    "FUNC"  "LibPub "]
```

```
libShortcut("linalg2", "la")
    {la.clearmat, la.gauss2}
```

```
libShortcut("linalg2", "la", 1)
    {la.clearmat, la.gauss1, la.gauss2}
```

LinRegBx

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

Beregner den lineære regression $y = a + b \cdot x$ på listerne *X* og *Y* med hyppigheder *Frekv*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146.)

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 208.

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 146.)

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 .

Kategory 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 208.

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-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>

LinRegtIntervaller

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

Til hældning. Beregner et niveau C konfidensinterval for hældningen.

LinRegtIntervals $X, Y[, F[, 1, Xværdi[, CNiveau]]]$

Åbent svar Beregner en forudset y -værdi, et niveau C forudsigelsesinterval for enkle observationer, og et niveau C konfidensinterval til gennemsnits-responsen.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146)

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 208.

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
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

Output-variabel	Beskrivelse
stat.MEPred	Prædiktionsintervalsmargin for fejl
stat.SEPred	standardfejl for prædiktion
Statistik.ŷ	$a + b \cdot X$ Værdi

LinRegtTest

katalog > 

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

Beregner en lineær regression ud fra X og Y listerne og en t test på værdien af hældningen 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 $Hypot = 0$

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

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

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a + b \cdot x$
stat.t	t -Statistik for signifikanstest

Output-variabel	Beskrivelse
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	Standarfejl for hældning
stat.r ²	Forklaringsgraden
stat.r	Korrelationskoefficient
stat.Resid	Residualer fra regressionen

linSolve()

Katalog > 

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

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

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

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

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

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

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

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

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

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

Returnerer en løsning for variablene Var1, Var2, ...

Det første argument skal kunne beregnes til et system af lineære ligninger eller en enkelt lineær ligning. Eller opstår der en argumentfejl.

For eksempel giver beregningen af **linSolve(x=1 og x=2,x)** resultatet "Argumentfejl".

$\Delta\text{list}()$

Katalog >  $\Delta\text{list}(\text{Liste1}) \Rightarrow \text{liste}$ $\Delta\text{List}(\{\{20,30,45,70\}\}) \quad \{10,15,25\}$

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()`

Katalog > `list►mat(Liste [, elementerPrRække]) ⇒ matrix`

<code>list►mat(\{1,2,3\})</code>	$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$
<code>list►mat(\{1,2,3,4,5\},2)</code>	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$

Returnerer en matrix fyldt rækkevis med elementerne fra *Liste*.

ElementerPrRække angiver antallet af elementer pr. række, hvis den er medtaget. Standard er antallet af elementer i *Liste* (en række).

Hvis *Liste* ikke udfylder den resulterende matrix, tilføjes nuller.

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

`ln()`

  -taster`ln(Liste1) ⇒ liste` $\ln(2.) \quad 0.693147$

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(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 $\blacktriangle$ og derefter bruge $\blacktriangleleft$ og $\blacktriangleright$ til at bevæge markøren.

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 146.)

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 208.

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 <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

Local

Katalog > 

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

Erklærer de angivne *var* som lokale variable. Disse variable eksisterer kun under beregning af en funktion og slettes, når eksekveringen af funktionen afsluttes.

Bemærk: Lokale variable sparer hukommelse, fordi kun eksisterer midlertidigt. De forstyrrer heller ikke de eksisterende globale variabelværdier. Lokale variable skal anvendes til **For**-løkker og midlertidig lagring af variabelværdier i en flerlinjefunktion da modifikationer af globale ikke er tilladt i en funktion.

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

```

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

```

	Done
rollcount()	16
rollcount()	3

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 systemvariabeln *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 163 og **getLockInfo()**, side 63.

<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

log()

  -taster

Returner *-Udtr2*-talslogaritmen til argumentet.

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

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

Hvis *Udtr2* udelades, anvendes 10-talslogaritmen.

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

Hvis tal-argumentet udelades, anvendes 10-talslogaritmen.

Hvis kompleks formattilstand er reel:

Hvis kompleks formattilstand er rektangulær:

I vinkeltilstanden radian og rektangulært komplekst format:

$$\log_{10} \left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix} \right) = \begin{bmatrix} 0.795387+0.753438 \cdot i & 0.003993-0.6474 \cdot i \\ 0.194895-0.315095 \cdot i & 0.462485+0.27079 \cdot i \\ -0.115909-0.904706 \cdot i & 0.488304+0.7774 \cdot i \end{bmatrix}$$

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

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 146.)

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 208.

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 <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>

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 146.)

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 208.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $c / (1 + a \cdot e^{-bx}) + d$
stat.a, stat.b, stat.c, stat.d	Regressionskoefficienter
stat.Resid	Residualer fra regressionen

Output-variabel	Beskrivelse
stat.XReg	Liste af datapunkter i den modificerede <i>X-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>Kategori liste</i> og <i>Medtag Kategorier</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y-liste</i> der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>Kategori liste</i> og <i>Medtag Kategorier</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

Loop

Katalog > 

Loop

Blok

EndLoop

Eksekverer gentagne gange sætningerne i *Blok*. Bemærk, at løkken eksekveres uendeligt, medmindre en **Goto** eller **Exit**-kommando eksekveres i *Blok*.

Blok er en sekvens af sætninger adskilt med kolon.

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

```

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

```

	Done
rollcount()	16
rollcount()	3

LU *Matrix*, *lMatrix*, *uMatrix*, *pMatrix* [,*Tol*]

Beregner Doolittle LU (nedre-øvre) opløsningen af en reel eller kompleks matrix. Den nedre triangulære matrix lagres i *lMatrix*, den øvre triangulære matrix i *uMatrix*, og permutationsmatricen (der beskriver de foretagne rækkeombytninger under beregningen) i *pMatrix*.

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

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

- Hvis du anvender   eller indstiller **Auto eller tilnærmet** - tilstanden til Approximate, foretages beregningerne med aritmetik med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:
- $5E-14 \cdot \max(\dim(Matrix)) \cdot \text{rækkeNorm}(Matrix)$

LU-algoritmen til faktoropløsning anvender partiel pivotering med rækkeombytninger.

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

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

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

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

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

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1 \quad \begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

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

$$\text{lower} \quad \begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \end{bmatrix}$$

$$\text{upper} \quad \begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$$

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

M

mat▶list()

mat▶list(*Matrix*) ⇒ *liste*

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

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

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

$$\text{mat}\blacktriangleright\text{list}(m1) \quad \{1, 2, 3, 4, 5, 6\}$$

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

max()

Katalog >

max(Liste1, Liste2)⇒liste

$\max\{2.3, 1.4\}$	2.3
--------------------	-----

max(Matrix1, Matrix2)⇒matrix

$\max\left\{\begin{Bmatrix} 1, 2 \end{Bmatrix}, \begin{Bmatrix} -4, 3 \end{Bmatrix}\right\}$	$\begin{Bmatrix} 1, 3 \end{Bmatrix}$
--	--------------------------------------

Returnerer maksimum af de to argumenter. Hvis argumenterne er to lister eller matricer, returneres en liste eller matrix med maksimumsværdier for hvert sammenhørende elementpar.

max(List)⇒udtryk

$\max\{0, 1, -7, 1.3, 0.5\}$	1.3
------------------------------	-----

Returnerer det største element i *liste*.

max(Matrix1)⇒matrix

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

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

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208

Bemærk: Se også `min()`.

mean()

Katalog >

mean(Liste[, hyppighedsliste])⇒udtryk

$\text{mean}(\{0.2, 0.1, -0.3, 0.4\})$	0.26
--	------

Returnerer middelværdien for elementerne i *Liste*.

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

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

mean(Matrix1[, Hyppighedsmatrix])⇒matrix

I rektangulært vektorformat:

Returnerer en rækkevektor af middelværdierne af alle kolonner i *Matrix1*.

mean()Katalog > 

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208.

mean	$\begin{pmatrix} 0.2 & 0 \\ -1 & 3 \\ 0.4 & -0.5 \end{pmatrix}$	$\begin{bmatrix} -0.133333 & 0.833333 \end{bmatrix}$
mean	$\begin{pmatrix} \frac{1}{5} & 0 \\ -1 & 3 \\ \frac{2}{5} & \frac{-1}{2} \end{pmatrix}$	$\begin{bmatrix} \frac{-2}{15} & \frac{5}{6} \end{bmatrix}$
mean	$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}, \begin{pmatrix} 5 & 3 \\ 4 & 1 \\ 6 & 2 \end{pmatrix}$	$\begin{bmatrix} \frac{47}{15} & \frac{11}{3} \end{bmatrix}$

median()Katalog > 

median(*Liste*[,
Hyppighedsliste]) $\Rightarrow$ udtryk

Returnerer medianen af elementerne i *Liste*.

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

median(*Matrix1*[,
Hyppighedsmatrix]) $\Rightarrow$ matrix

Returnerer en rækkevektor med medianerne af kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af forekomster i en ubrudt rækkefølge af det tilsvarende element i *Matrix1*.

median	$\{ \{ 0.2, 0.1, -0.3, 0.4 \} \}$	0.2
median	$\begin{pmatrix} 0.2 & 0 \\ 1 & -0.3 \\ 0.4 & -0.5 \end{pmatrix}$	$\begin{bmatrix} 0.4 & -0.3 \end{bmatrix}$

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 208.

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 146.)

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 208.

Output-variabel	Beskrivelse
stat.RegEqn	Median-median-linjeligning: $m \cdot x + b$
stat.m, stat.b	Modelkoefficienter
stat.Resid	Residualer fra median-median-linjen
stat.XReg	Liste af datapunkter i den modificerede X -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag 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>

mid()

Katalog > **mid(kildeStreng, Start[, Antal])**⇒streng

Returnerer *Antal* tegn fra tegnstrengen *kildeStreng*, startende med tegn nummer *Start*.

Hvis *Antal* udelades eller er større end dimension på *kildeStreng*, returneres alle tegn fra *kildeStreng*, begyndende med tegn nummer *Start*.

Antal skal være ≥ 0 . Hvis *Antal* = 0, returneres en tom streng.

mid(kildeListe, Start [, Antal])⇒liste

Returnerer *Antal* elementer fra *kildeListe*, begyndende med element nummer *Start*.

Hvis *Antal* udelades eller er større end dimensionen på *kildeListe*, returneres alle elementer fra *kildeListe*, begyndende med element nummer *Start*.

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

mid(kildeStrengListe, Start[, Antal])⇒liste

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

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

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

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

min()

Katalog > **min(Liste1, Liste2)**⇒liste**min(Matrix1, Matrix2)**⇒matrix

Returnerer minimum af de to argumenter. Hvis argumenterne er to lister eller matricer, returneres en liste eller matrix med minimumværdi af hvert sammenhørende elementpar.

min(Liste)⇒udtryk

Returnerer det mindste element af *Liste*.

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

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

min()Katalog > **min**(*Matrix1*) $\Rightarrow$ *matrix*

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

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

Bemærk: Se også **max()**.

mirr()Katalog > **mirr**

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

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

Finansfunktion, der returnerer den modificerede interne rente af en investering.

finansRente er rentesatsen, du betaler for pengestrømsbeløbene.

geninvestRente er rentesatsen, som pengestrømmen geninvesteres til.

CF0 er startpengestrømmen på tidspunkt 0. Den skal være et reelt tal.

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

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

Bemærk: Se også **irr()**, side 73.

mod()Katalog > **mod**(*Liste1*, *Liste2*) $\Rightarrow$ *liste* $\text{mod}(7,0)$ 7**mod**(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix* $\text{mod}(7,3)$ 1

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

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

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 125

mRow()Katalog > 

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

Returnerer en kopi af *Matrix1* med hvert element i rækken *Indeks2* af *Matrix1* erstattet med:

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 146.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

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.yList	$\hat{y}_{Liste} = b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualer fra regressionen

MultRegIntervals

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-responen.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146.)

Alle lister skal have samme dimension.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
Statistik. $\hat{y}$	Et punktestimat: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ for <i>XValListe</i>
stat.dfFejl (stat.dfeError)	Frihedsgrader for fejl
stat.CLower, stat.CUpper	Konfidensinterval for en middelværdi
stat.ME	Konfidensinterval, fejlmargen
stat.SE	Standardfejl for middelværdi
stat.LowerPred,	Prædiktionsinterval for en enkelt observation

Output-variabel	Beskrivelse
stat.UpperPred	
stat.MEPred	Prædiktionsintervalsmargin for fejl
stat.SEPred	Standardfejl for prædiktion
stat.bList	Liste med regressionskoefficienter, {b0,b1,b2,...}
stat.Resid	Residualer fra regressionen

MultRegTests

Katalog > 

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

Multipel lineær regressionstest beregner en multipel lineær regression fra de givne data, og danner den globale F teststatistik og t teststatistikker for koefficienterne.

En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

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

Output-variabel	Beskrivelse
stat.SSError	fejl, kvadratsum
stat.MSError	fejl, middeltkvadrat
stat.bList	{b0,b1,...} Liste med koefficienter
stat.tList	Liste med t statistikker for hver koefficient i bListen
stat.PList	Liste P-værdier for hver t-statistik
stat.SEList	Liste med standardfejl for koefficienter i bListe
stat.yList	$\hat{y}$ Liste = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residualer fra regressionen
stat.sResid	Standardiserede residualer; Værdi opnået ved at dividere en residual med dens standardafvigelse
stat.CookDist	Cooks distance; Mål for påvirkningen af en observation baseret på residual og udnyttelse
stat.Leverage	Mål for, hvor langt værdierne for de uafhængige variable er fra deres middelværdier

nand

  -taster

BoolskUdtryk1 **nand** *BoolskUdtryk2*
returnerer *boolsk udtryk*

$x \geq 3$ and $x \geq 4$	$x \geq 4$
---------------------------	------------

$x \geq 3$ 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.

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

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

N

nCr()

Katalog > 

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

Returnerer en liste med kombinationer baseret på de sammenhørende elementpar i de to lister. Argumenterne skal være lister af samme størrelse.

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

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

Returnerer en matrix af kombinationer baseret på de sammenhørende elementpar i de to matricer. Argumenterne skal være matricer af samme størrelse.

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

nDerivative()

Katalog > 

nDerivative(*Udtr1*, *Var*=*Værdi* [*Orden*]) $\Rightarrow$ *værdi*

nDerivative(*Udtr1*, *Var* [*Orden*]) | *Var*=*Værdi* $\Rightarrow$ *værdi*

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

nDerivative()

Katalog > 

Returnerer den numeriske differentialkvotient udregnet med en automatisk differentiationsmetode.

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)⇒liste

<code>newList(4)</code>	$\{0,0,0,0\}$
-------------------------	---------------

Returnerer en liste med en dimension af *antalElementer*. Hvert element er nul.

newMat()

Katalog > 

newMat(antalRækker, antalKolonner)⇒matrix

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

Returnerer en matrix med nulpunkter af dimensionen *antalRækker* gange *antalKolonner*.

nfMax()

Katalog > 

nfMax(Udtr, Var)⇒værdi

<code>nfMax($-x^2-2 \cdot x-1, x$)</code>	-1.
--	-----

nfMax(Udtr, Var, nedreGrænse)⇒værdi

<code>nfMax($0.5 \cdot x^3-x-2, x, -5, 5$)</code>	5.
--	----

nfMax(Udtr, Var, nedreGrænse, øvreGrænse)⇒værdi

nfMax(Udtr, Var) | nedreGrænse≤Var ≤øvreGrænse⇒værdi

Returnerer mulig numerisk værdi for variabelen *Var*, hvor det lokale maksimum for *Udtr* optræder.

Hvis du opgiver *nedreGrænse* og *øvreGrænse*, søger funktionen i det lukkede interval $[nedreGrænse, øvreGrænse]$ efter det lokale maksimum.

nfMin(*Udtr*, *Var*) $\Rightarrow$ værdi

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.

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

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.

$\text{nInt}(e^{-x^2}, x, -1, 1)$	1.49365
-----------------------------------	---------

Der vises en advarsel ("Tvivl om nøjagtighed") når målet ikke ser ud til at være nået.

Indskyd flere **nInt()**, for at foretage numerisk integration i flere variable. Integrationsgrænser kan afhænge af integrationsvariable uden for dem.

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

nom(*effektivRente*, *CpY*) ⇒ *værdi*

$$\text{nom}(5.90398, 12) \quad 5.75$$

Finansfunktion, der omregner den effektive årlige rente *effektivRente* til en nominel rente, hvor *CpY* er antallet af rentetilskrivninger per år.

effektivRente skal være et reelt tal, og *CpY* skal være et reelt tal > 0.

Bemærk: Se også **eff()**, side 42.

BoolskUdtryk1 **nor** *BoolskUdtryk2*
returnerer *boolsk udtryk*

$$x \geq 3 \text{ or } x \geq 4 \quad x \geq 3$$

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

$$x \geq 3 \text{ nor } x \geq 4 \quad x < 3$$

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 hexadecimal indtastninger skal du som præfiks benytte henholdsvis 0b eller 0h. Uden præfiks behandles heltallene som decimaltal (10 talssystem)

norm()

Katalog > 

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

norm(*Vektor*) $\Rightarrow$ *udtryk*

Returnerer Frobenius-normen.

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).

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

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 hexadecimale 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 15.

I hexadecimal tilstand:

Vigtigt: Tallet nul, ikke bogstavet O.

not 0h7AC36	0hFFFFFFFFFFFF853C9
-------------	---------------------

I binær tilstand:

0b100101 ►Base10	37
not 0b100101	
0b11111111111111111111111111111111 ►	
not 0b100101 ►Base10	-38

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

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

nPr()

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

Returnerer en liste med permutationer baseret på de sammenhørende elementpar i de to lister. Argumenterne skal være lister af samme størrelse.

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

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

nPr($\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$)	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$
---	---

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*])

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.

$list1 := \{6000, -8000, 2000, -3000\}$	
$\{6000, -8000, 2000, -3000\}$	
$list2 := \{2, 2, 2, 1\}$	$\{2, 2, 2, 1\}$
$npv(10, 5000, list1, list2)$	4769.91

nSolve()

Katalog >

nSolve(*Ligning*,*Var*[=*Gæt*]) $\Rightarrow$ *tal* eller fejlstreng

nSolve(*Ligning*,*Var*
[=*Gæt*],*nedreGrænse*) $\Rightarrow$ *tal* eller fejlstreng

nSolve(*Ligning*,*Var*
[=*Guess*],*nedreGrænse*,*øvreGrænse*)

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

Bemærk: Hvis der er flere løsninger, kan du anvende et gæt til at finde en partikulær løsning.

$\Rightarrow$ tal eller fejlstreng

nSolve(*Ligning*, *Var*[=*Guess*]) |
nedreGrænse $\leq$ *Var* $\leq$ *øvreGrænse* $\Rightarrow$ tal
 eller fejlstreng

Søger iterativt efter en approksimeret
 reel numerisk løsning af *Ligning* for
 dens ene variabel. Angiv variabelen som:

variabel

– eller –

variabel = *reelt tal*

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

nSolve() forsøger at bestemme enten et
 punkt, hvor residualen er nul, eller to
 forholdsvis tætte punkter, hvor
 residualerne har modsatte fortegn, og
 residualen ikke er for stor. Hvis dette
 ikke kan opnås med et beskedent antal
 datapunkter, returneres strengen "Ingen
 løsning blev fundet."

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

O

OneVar

OneVar [*1*], *X*[, [*Hyp*], [*pyghed*]]
 [, *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 146.)

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 208

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*

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

BoolskMatrix1 **or** *BoolskMatrix2*
returnerer *Boolsk matrix*

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

Returnerer true, hvis enten et eller begge udtryk kan reduceres til true. Returnerer kun false, hvis begge udtryk evalueres til false.

Bemærk: Se *xor*.

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

Heltal1 **or** *Heltal2* $\Rightarrow$ *heltal*

Sammenligner to reelle heltal bit for bit med en or-operation. Internt konverteres begge heltal til 64-bit binære tal med fortegn. Når de tilsvarede bits sammenlignes, er resultatet 1, hvis en af bittene er 1. Resultatet er kun 0, hvis begge bits er 0. Den returnerede værdi repræsenterer bit-resultaterne og vises i overensstemmelse med den valgte talsystemstilstand.

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

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

I hexadecimal tilstand:

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

Vigtigt: Tallet nul, ikke bogstavet O.

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 15.

Bemærk: Se xor.

ord()

ord(*Streng*)⇒*heltal*

ord("hello")	104
--------------	-----

ord(*Liste1*)⇒*liste*

char(104)	"h"
-----------	-----

ord(char(24))	24
---------------	----

Returnerer den numeriske kode til første tegn i tegnstrengen *Streng*, eller en liste med de første tegn i hvert listeelement.

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

P**►R_x()**

►R_x(*rUdtr*, *θUdtr*)⇒*udtryk*

I vinkeltilstanden Radian:

►R_x(*rListe*, *θListe*)⇒*liste*

►R_x(*rMatrix*, *θMatrix*)⇒*matrix*

Returnerer den ækvivalente x-koordinat til parret (*r*, *θ*).

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@>R_x** (...).

►R_y()

►R_y(*rListe*, *θListe*)⇒*liste*

I vinkeltilstanden Radian:

►R_y(*rMatrix*, *θMatrix*)⇒*matrix*

Returnerer den ækvivalente y-koordinat til parret (r, θ).

Bemærk: Argumentet θ tolkes i grader, nygrader eller radianer afhængigt af den aktuelle vinkeltilstand.

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

PassErr

PassErr

Se et eksempel på **PassErr** i Eksempel2 under **Try**-kommandoen, side 157.

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 22, og **Try**, side 157.

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 stykkevis definitioner ved hjælp af en skabelon.

Bemærk: Se også **Stykkevis skabelon**, side 2.

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

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 .

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 122.

kompleksVærdi ►Polar

I vinkeltilstanden Radian:

Viser *kompleksVærdi* i polær form.

- Vinkeltilstanden Grader returnerer

I vinkeltilstanden Nygrader:

$(r \angle \theta)$.

- Vinkeltilstanden Radian returnerer $re^{i\theta}$.

 $(4 \cdot i)$ ►Polar $(4 \angle 100.)$

kompleks Værdi kan have enhver kompleks form. Men en $re^{i\theta}$ -indtastning udløser en fejl i vinkeltilstanden Grader.

I vinkeltilstanden Grader:

Bemærk: Du skal anvende parenteser til en $(r \angle \theta)$ polær indtastning.

polyEval()**polyEval**(Liste1, Udtr1)⇒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.

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

Den første syntaks, **polyRoots**(Poly, Var), returnerer en liste af reelle rødder af polynomiet *Poly* med hensyn til variablen *Var*. Hvis der ikke findes en reel rod, returneres en tom liste: {}.

Den anden syntaks, **polyRoots**(ListeAfKoeff), returnerer en liste med reelle rødder for koefficienterne i *ListeAfKoeff*.

Bemærk: Se også **cPolyRoots()**, side 29.

PowerReg

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 146.)

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 208.

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*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

prodSeq()

Se $\Pi()$, side 183.

Product (Pi)

Se $\Pi()$, side 183.

product()

Katalog > 

product(*Liste*[, *Start*[, *slut*]])⇒*udtryk*

Returnerer produktet af elementerne indeholdt i *Liste*. *Start* og *Slut* er valgfri. De angiver en serie af elementer.

product(*Matrix1*[, *Start*[, *slut*]])⇒*matrix*

Returnerer en rækkevektor med produkterne af elementerne i kolonnerne i *Matrix1*. *Start* og *slut* er valgfri. De angiver en serie af rækker.

product $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ [28 80 162]

product $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, 1, 2$ [4 10 18]

Tomme (ugyldige) elementer ignoreres.
Yderligere oplysninger om tomme elementer findes på side 208.

propFrac()

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.

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.

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

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

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

$$\text{propFrac}(\text{Ans}) \quad \frac{1}{x+1} + x + \frac{1}{y+1} + y$$

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.

Tallet med flydende komma (9). i m1 får resultaterne beregnet med flydende komma.

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

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.	

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

QR faktoropløsningen beregnes numerisk med Householder-transformationer. Den symbolske løsning beregnes med Gram-Schmidt. Kolonnerne i *qmatNavn* er ortonormale vektorer, der udspejler rummet defineret ved *matrix*.

$\begin{bmatrix} m & n \\ o & p \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$
QR m1,qm,rm Done	
qm	$\begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m \cdot p - n \cdot o) \cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m \cdot \text{sign}(m \cdot p - n \cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$
rm	$\begin{bmatrix} \sqrt{m^2+o^2} & \frac{m \cdot n + o \cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{ m \cdot p - n \cdot o }{\sqrt{m^2+o^2}} \end{bmatrix}$

QuadReg

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 146.)

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 208.

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 X -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategori</i>
stat.YReg	Liste af datapunkter i den modificerede Y -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategori</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

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 146.)

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 208.

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</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategori</i>
stat.YReg	Liste af datapunkter i den modificerede <i>Y</i> -liste der faktisk bruges i regressionen ud fra begrænsninger i <i>Frekv</i> , <i>kategoriliste</i> , og <i>Medtag Kategori</i>
stat.FreqReg	Liste med hyppigheder, der svarer til <i>stat.XReg</i> og <i>stat.YReg</i>

R ► $P\theta()$ Katalog > R ► $P\theta(xList, yList) \Rightarrow$ listeR ► $P\theta(xMatrix, yMatrix) \Rightarrow$ matrix

Returnerer den ækvivalente θ -koordinat for (x, y) argumentparret.

I vinkeltilstanden Grader:

I vinkeltilstanden Nygrader:

I vinkeltilstanden Radian:

Bemærk: Resultatet returneres som en vinkel i grader, nygrader eller radian afhængig af den aktuelle vinkeltilstand.

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive

`R@>Ptheta (...)`.

R ► $Pr()$ Katalog > R ► $Pr(xList, yList) \Rightarrow$ listeR ► $Pr(xMatrix, yMatrix) \Rightarrow$ matrix

Returnerer den ækvivalente r -koordinat for (x, y) argumentparret.

I vinkeltilstanden Radian:

Bemærk: Du kan indsætte denne funktion fra computerens tastatur ved at skrive `R@>Pr`

`(...)`.

► Rad

Katalog > 

Konverterer argumentet til vinkelmåling i radian.

I vinkeltilstanden grader:

$(1.5) \blacktriangleright \text{Rad}$	$(0.02618)^r$
--	---------------

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

I vinkeltilstanden Nygrader:

$(1.5) \blacktriangleright \text{Rad}$	$(0.023562)^r$
--	----------------

rand()

Katalog > rand() $\Rightarrow$ udtrykrand(#Trials) $\Rightarrow$ liste

Angiv seed-værdien for et vilkårligt tal.

rand()

Katalog > 

rand() returnerer en vilkårlig værdi mellem 0 og 1.

RandSeed 1147

Done

rand(2)

{ 0.158206, 0.717917 }

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.

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

(
lowBound, *upBound*)

returnerer et vilkårligt heltal i det område, der angives af heltalsgrænserne *lowBound* og *upBound*.

randInt

(*lowBound*, *upBound*
, #Trials)

returnerer en liste med #Trials vilkårlige heltal i det angivne område.

randMat()Katalog > **randMat**(*numRows*, *numColumns*) ⇒ *matrice*

Returnerer en matrix med heltal mellem -9 og 9 af den angivne dimension.

Begge argumenter skal kunne reduceres til heltal.

RandSeed 1147	Done
randMat (3,3)	$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$

Bemærk: Værdierne i denne matrix ændres, hver gang du trykker .

randNorm()Katalog > **randNorm**(μ , σ) ⇒ *udtryk***randNorm**(μ , σ , #*Trials*) ⇒ *liste*

randNorm(μ , σ) returnerer et decimaltal 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**(*Var*, *Order*) ⇒ *udtryk*

Returnerer et polynomium i *Var* af den angivne *Order*. Koefficienterne er vilkårlige heltal i området -9 til 9. Koefficienten af højeste grad vil ikke være nul.

Order skal være 0–99.

RandSeed 1147	Done
randPoly (x,5)	$-2\cdot x^5+3\cdot x^4-6\cdot x^3+4\cdot x-6$

randSamp()Katalog > **randSamp**(*List*, #*Trials*[, *noRepl*]) ⇒ *liste*

Returnerer en liste med en vilkårlig stikprøve af #*Trials* målinger fra *List* med mulighed for tilbagelægning (*noRepl*=0) eller ingen tilbagelægning (*noRepl*=1). Standardindstillingen er med tilbagelægning.

RandSeed *Number*

Hvis *Number* = 0, indstilles startværdien til fabriksindstillingerne for vilkårlig-talgeneratoren. Hvis *Number* $\neq$ 0, anvendes det til at generere to startværdier, der lagres i systemvariablerne seed1 og seed2.

RandSeed 1147

Done

rand()

0.158206

real()

Returnerer den reelle del af argumentet.

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

Returnerer realdelen af alle elementer.

real(*Matrix1*) $\Rightarrow$ *matrix*

Returnerer realdelen af alle elementer.

► Rect

Vektor ► **Rect**

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

Viser *Vektor* i rektangulær form [x, y, z]. Vektoren skal være af dimensionen 2 eller 3 og kan være en række eller kolonne.

Bemærk: ► **Rect** er en displayformatkommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje, og den opdaterer ikke *ans*.

Bemærk: Se også ► **Polær ligning**, side 111.

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.

I vinkeltilstanden Radian:

I vinkeltilstanden Nygrader:

$((1 \angle 100)) \blacktriangleright \text{Rect}$	<i>i</i>
--	----------

Bemærk: Du skal anvende parenteser til en $(r \angle \theta)$ polær indtastning.

I vinkeltilstanden grader:

Bemærk: Du kan skrive $\angle$ ved at vælge det i symbollisten i kataloget.

ref()

ref(MatrixI[, Tol]) $\Rightarrow$ matrix

Returnerer række-echelonformen af *MatrixI*.

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

- Hvis du anvender   eller indstiller **Auto-** eller **Approximate** - tilstanden til Approximate, foretages beregningerne med flydende komma.
- Hvis *Tol* udelades eller ikke anvendes, beregnes standardtolerancen som:
 $5E-14 \cdot \max(\dim(\text{MatrixI})) \cdot \text{rowNorm}(\text{MatrixI})$

Undgå udefinerede elementer i *MatrixI*. De kan føre til uventede resultater.

Hvis for eksempel *a* er udefineret i følgende udtryk, vil en advarsel blive vist, og resultatet vises som:

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

Advarslen vises, fordi det generaliserede element $1/a$ ikke ville være gyldig for $a=0$.

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

Du kan undgå dette ved i forvejen at gemme en værdi til *a* eller ved at bruge betingelsesoperatoren ("|") til at substituere en værdi som vist i følgende eksempel.

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

Bemærk: Se også `rref()`, page 132.

RefreshProbeVars

RefreshProbeVars

Giver dig adgang til sensordata fra alle tilkoblede sensorprober i dit TI-Basic-program.

StatusVar-værdi	Status
<i>statusVar</i> =0	Normal (fortsæt med programmet) Vernier DataQuest™-applikation er i dataindsamlingstilstand.
<i>statusVar</i> =1	Bemærk: Vernier DataQuest™-applikation skal være i målerstilstand for at denne kommando virker. 
<i>statusVar</i> =2	Vernier DataQuest™-applikation blev ikke startet.
<i>statusVar</i> =3	Vernier DataQuest™-applikationen blev startet, men du har ikke tilkoblet nogen prober.

Eksempel

```

Define temp()=
Prgm
© Tjek, om systemet er klar
RefreshProbeVars-status
If status=0 Then
Disp "klar"
For n,1,50
RefreshProbeVars-status
temperatur:=meter.temperatur
Disp "Temperatur: ",temperatur
If temperature>30 Then
Disp "For varmt"
EndIf
© Vent i 1 sekund mellem
stikprøver
Wait 1
EndFor
Else
Disp "Ikke klar. Prøv igen senere"

```

EndIf

EndPrgm

Bemærk: Dette kan også bruge med TI-Innovator™-hubben.

remain()

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

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 95.

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

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

Request

Request *promptString*, *var*[, *DispFlag* [, *statusVar*]]

Request *promptString*, *func*(*arg1*, ...*argn*) [, *DispFlag* [, *statusVar*]]

Programmeringskommando: Standser programmet midlertidigt og viser en dialogboks med meddelelsen *promptString* og et indtastningsfelt til brugerens svar.

Når brugeren skriver et svar og klikker på **OK**, tildeles indtastningsfeltets indhold til variabelen *var*.

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**:

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 historikken.

Det valgfrie 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.

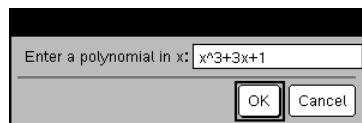
Radius: 6/2
Areal = 28,2743

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}

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 127.

RequestStr

RequestStr *promptString*, *var*[, *DispFlag*]

Programmeringskommando: Fungerer identisk med den første syntaks i kommandoen **Request**, bortset fra at brugerens svar altid fortolkes som en streng. I modsætning hertil fortolker kommandoen **Request** svarene som et udtryk, medmindre brugeren omslutter det med citationstegn ("").

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

Sådan standses et program, der indeholder en **RequestStr** kommando i en uendelig løkke:

- **Håndholdt:** Hold tasten  nede, mens du gentagne gange trykker på .
- **Windows®:** Hold tasten **F12** nede, mens du gentagne gange trykker på **Enter**.
- **Macintosh®:** Hold tasten **F5** nede, mens du gentagne gange trykker på **Enter**.
- **iPad®:** App'en viser en meddelelse. Du kan fortsat vente eller annullere.

Definer et program:

```
Define requestStr_demo()=Prgm
    RequestStr "Dit navn:",name,0
    Disp "Svaret har ",dim(name),"
    tegn."
EndPrgm
```

Start programmet, og indtast et svar:

requestStr_demo()



Resultat efter valg af **OK** (Bemærk, at argumentet *DispFlag* fra **0** udelader klarmeddelelsen og svaret fra historikken):

requestStr_demo()

Svaret har fem tegn.

Bemærk: Se også Request, page 125.

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()

right(*List1* [, *Num*]) ⇒ *liste*

Returnerer *Num*-elementer længst til højre i *List1*.

Hvis du udelader *Num*, returneres hele *List1*.

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

Returnerer *Num*-tegn længst til højre i tegnstrengen *sourceString*.

Hvis du udelader *Num*, returneres hele *sourceString*.

right(*Comparison*) ⇒ *udtryk*

Returnerer højre side af en ligning eller ulighed.

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

{ 3, 2, 4 }

right("Hello", 2)

"lo"

right($x < 3$)

3

rk23 ()

rk23(*Expr*, *Var*, *depVar*, {*Var0*, *VarMax*}, *depVar0*, *VarStep* [, *difTol*])
⇒ *matrix*

Differentialligning:

rk23(*SystemOfExpr*, *Var*,
ListOfDepVars, {*Var0*, *VarMax*},
ListOfDepVars0, *VarStep*[, *diftol*]) $\Rightarrow$
matrix

rk23(*ListOfExpr*, *Var*, *ListOfDepVars*,
{*Var0*, *VarMax*}, *ListOfDepVars0*,
VarStep[, *diftol*]) $\Rightarrow$ *matrix*

Anvender Runge-Kutta-metoden til at
løse systemet

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

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

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

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

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

Var er den uafhængige variabel.

ListOfDepVars er en liste med
afhængige variabler.

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

ListOfDepVars0 er en liste af
oprindelige værdier for afhængige
variabler.

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

$$\text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9493	13.042	14.2

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

Samme ligning med *diftol* sat til $1.E-6$

$$\text{rk23}\left(0.001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1, 1.E-6\right)$$

0.	1.	2.	3.	4.
10.	10.9367	11.9495	13.0423	14.2189

System af ligninger:

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

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

$$\text{rk23}\left(\begin{cases} y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0, 5\}, \{2, 5\}, 1\right)$$

0.	1.	2.	3.	4.
2.	1.94103	4.78694	3.25253	1.82848
5.	16.8311	12.3133	3.51112	6.27245

Hvis $VarStep$ evalueres til et tal forskelligt fra nul: $sign(VarStep) = sign(VarMax - Var0)$ og løsninger returneres ved $Var0 + i * VarStep$ for alle $i=0,1,2,...$ således at $Var0 + i * VarStep$ er i $[var0, VarMax]$ (der vil muligvis ikke være en løsningsværdi ved $VarMax$).

Hvis *VarStep* evalueres til nul, returneres løsningerne i "Runge-Kutta" *Var*-værdier.

*dif*tol er fejltolerancen (som standard 0.001).

root()

Bemærk: Se også N-te rod-skabelon, side 1.

rotate()

$$\text{rotate}(\text{Integer } l[, \text{\#ofRotations}]) \Rightarrow \text{heltal}$$

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 15.

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
```

```
rotate(0b11111111111111111111111111111111)
0b1000000000000000000000000000000001
rotate(256,1)                                0b1000000000
```

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 hexadecimaltastninger skal du som præfiks altid benytte henholdsvis 0b eller 0h (nul, ikke bogstavet O).

rotate()

Katalog > 

Resultatet vises i den valgte tilstand for talsystem.

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

I decimal tilstand:

Returnerer en kopi af *ListI* roteret til højre eller venstre med *#ofRotations*-elementer. Ændrer ikke *ListI*.

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

Hvis *#ofRotations* er positivt, kører rotationen mod venstre. Hvis *#ofRotations* er negativt, kører rotationen mod højre. Standardindstillingen er -1 (roter en bit til højre).

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

Returnerer en kopi af *StringI* roteret til højre eller venstre med *#ofRotations*-tegn. Ændrer ikke *StringI*.

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

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 > 

Returnerer argumentet afrundet til det angivne antal cifre efter decimalpunktet.

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

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

Returnerer en liste med elementerne afrundet til det angivne antal cifre.

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

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

Returnerer en matrix med elementerne afrundet til det angivne antal cifre.

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

rowAdd()Katalog > 

rowAdd(*Matrix1*, *rIndex1*, *rIndex2*) $\Rightarrow$
matrix

Returnerer en kopi af *Matrix1* med rækken *rIndex2* udskiftet med summen af rækker *rIndex1* og *rIndex2*.

rowDim()Katalog > 

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

Returnerer antallet af rækker i *Matrix*.

Bemærk: Se også **colDim()**, side 22.

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

rowNorm()Katalog > 

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

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

Bemærk: Alle matricens elementer skal kunne reduceres til tal. Se også **colNorm()**, side 23.

rowNorm	25
$\begin{pmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{pmatrix}$	

rowSwap()Katalog > 

rowSwap(*Matrix1*, *rIndex1*, *rIndex2*)
 $\Rightarrow$ *matrix*

Returnerer *Matrix1* med rækkerne *rIndex1* og *rIndex2* byttet.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \rightarrow mat$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
rowSwap (<i>mat</i> , 1, 3)	$\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$

rref()Katalog > 

rref(*Matrix1* [, *Tol*]) $\Rightarrow$ *matrix*

Returnerer den reducerede række-echelonform af *Matrix1*.

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

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

Bemærk: Se også **ref()**, page 123.

S

sec()

 -tast

sec(ListeI) $\Rightarrow$ *list*

I vinkeltilstanden Grader:

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

sec⁻¹()

 -tast

sec⁻¹(ListeI) $\Rightarrow$ *liste*

I vinkeltilstanden Grader:

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

I vinkeltilstanden Nygrader:

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

I vinkeltilstanden Radian:

sech()

Katalog > 

sech(ListeI) $\Rightarrow$ *list*

sech⁻¹ (*Liste1*) ⇒ *liste*

I vinkeltilstanden radian og tilstanden rektangulært kompleks:

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

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 58), **GetStr** (side 64), og **eval()** (side 46).

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 *lightval*

Done

lightval

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.

n:=50

50

m:=4

4

Send "SET SOUND eval(*m*·*n*)"

Done

iostr.SendAns

"SET SOUND 200"

seq()

Katalog > seq(*Udtr*, *Var*, *Lav*, *Høj* [, *Trin*]) ⇒ *liste*

Øger *Var* fra *Low* til *High* i trin på *Step*, beregner *Expr* og returnerer resultaterne som en liste. Det oprindelige indhold af *Var* er der stadigvæk, når **seq()** er gennemført.

Standardværdien for *trin* = 1.

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

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 *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.

Det oprindelige indhold af *Var* er uændret efter **seqGen()** er gennemført .

Standardværdien for *VarStep* = 1.

Genererer de første fem led i sekvensen *u*(*n*) = *u*(*n*-1)²/2 med *u*(1)=2 og *VarStep*=1.

$$\text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right) \\ \left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$$

Eksempel, hvor *Var0*=2:

$$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right) \\ \left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$$

System med to sekvenser:

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

Bemærk: Void () i den oprindelige ledmatrix ovenfor bruges for at angive, at det oprindelige led for *u*1(*n*) er beregnet vha. den eksplicitte sekvensformel *u*1(*n*)=1/*n*.

seqn(*Expr*(*u*, *n* [, *ListOfInitTerms* [, *nMax* [, *CeilingValue*]]]) ⇒ *liste*

Genererer en liste med led for sekvensen $u(n) = \text{Expr}(u, n)$ som følger: Øger n fra 1 til $nMax$ med 1, beregner $u(n)$ for de tilsvarende værdier af n vha. udtrykket $\text{Expr}(u, n)$ og *ListOfInitTerms*, og returnerer resultaterne som en liste.

seqn(*Expr*(*n* [, *nMax* [, *CeilingValue*]]]) ⇒ *liste*

Genererer en liste med led for sekvens $u(n) = \text{Expr}(n)$ som følger: Øger n fra 1 til $nMax$ med 1, beregner $u(n)$ for de tilsvarende værdier af n vha. udtrykket $\text{Expr}(n)$ og returnerer resultaterne som en liste.

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 $u(1)=2$.

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

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

setMode()

setMode(*tilstandNavnHeltal*, *indstilHeltal*) ⇒ *heltal* **setMode**(*liste*) ⇒ *heltalsliste*

Kun gyldig i en funktion eller et program.

setMode(*tilstandNavnHeltal*, *indstilHeltal*) indstiller midlertidigt tilstanden *tilstandNavnHeltal* til den nye indstilling *indstilHeltal* og returnerer et heltal, der svarer til denne tilstands oprindelige indstilling. Ændringen er begrænset til varigheden af eksekveringen af programmet/funktionen.

tilstandNavnHeltal angiver hvilken tilstand, du vil indstille. Det skal være et af tilstandsheltallene fra nedenstående tabel.

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.

indstilHeltal angiver den nye indstilling for tilstanden. Det skal være et af indstillingsheltallene for den tilstand, du indstiller.

setMode(*liste*) lader dig ændre flere indstillinger. *liste* indeholder et par af tilstandsheltal og indstillingsheltal. **setMode(*liste*)** returnerer en tilsvarende liste, hvis heltalspar repræsenterer de oprindelige tilstande og indstillinger.

Hvis du har gemt alle tilstandsindstillinger med **getMode(0)** → *var*, kan du anvende **setMode(*var*)** til at gendanne disse indstillinger, indtil funktionen eller programmet afsluttes. Se **getMode()**, side 63.

Bemærk: De aktuelle tilstandsindstillinger videresendes til kaldte under rutiner. Hvis en under rutine ændrer en tilstandsindstilling, mistes tilstandsændringen, når kontrollen returnerer til den kaldende rutine.

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

Tilstands- navn	Tilstands- heltal	Indstillingsheltal
Vektorformat	6	1=Rektangulær, 2=Cylindrisk, 3=Sfærisk
Talsystem	7	1=Decimal, 2=Hex, 3=Binær

shift()

Katalog > 

shift(Heltal1[,antalFlyt])⇒heltal

Flytter bittene i et binært heltal. Du kan indtaste *Heltal1* i ethvert talsystem. Det konverteres automatisk til en 64-bit binær form med fortegn. Hvis *Heltal1* er for stort til denne form, bringer en symmetrisk modulooperation værdien ind i det korrekte område. Yderligere oplysninger findes under ►Base2, side 15.

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 bitten længst mod højre, og 0 eller 1 indsættes for at matche bitten længst til venstre. I en flytning til venstre droppes bitten længst mod venstre, og 0 indsættes som bitten længst til højre.

For eksempel i en højreflytning:

Hver bit flytter til højre.

0b0000000000000111101011000011010

Indsætter 0, hvis bitten længst til venstre er 0,

eller 1, hvis bitten længst til venstre er 1.

giver:

00b0000000000000011110101100001101

Resultatet vises i den valgte tilstand for talsystem. Der vises ikke foranstillede nuller.

shift(Liste1[,antalFlytninger])⇒liste

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()Katalog > 

Returnerer en kopi af *Liste1* flyttet til højre eller venstre med *antalflytninger* elementer. Ændrer ikke *Liste1*.

Hvis *antalFlytninger* er positivt, kører flytningen mod venstre. Hvis *antalFlytninger* er negativt, kører flytningen mod højre. Standardindstillingen er -1 (flyt en bit til højre).

Elementer indført i starten eller slutningen af *liste* af flytningen, indstilles til symbol "undef".

shift(*Streng1* [,*antalFlytninger*])⇒*streng*

Returnerer en kopi af *Streng1* flyttet til højre eller venstre med *antalflytninger* tegn. Ændrer ikke *Streng1*.

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({ 1,2,3,4 })	{ undef,1,2,3 }
shift({ 1,2,3,4 },-2)	{ undef,undef,1,2 }
shift({ 1,2,3,4 },2)	{ 3,4,undef,undef }

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

sign()Katalog > 

sign(*Liste1*)⇒*liste*

sign(*Matrix1*)⇒*matrix*

Hvis kompleks formattilstand er real:

sign(0) returnerer ±1, hvis det komplekse format er reel. Ellers returnerer det sig selv.

sign(0) repræsenterer enhedscirklen i det komplekse domæne.

For en liste eller matrix returneres fortegnene af elementerne.

simult()Katalog > 

simult(*koeffMatrix*, *konstVektor* [,

Løs for x og y:

$Tol \Rightarrow matrix$

Returnerer en kolonnevektor, der indeholder løsningerne til et system af lineære ligninger.

Bemærk: Se også **linSolve()**, side 81.

coeffMatrix skal være en kvadratisk matrix, der indeholder koefficienterne til ligningerne.

konstVektor skal have samme antal rækker (samme dimension) som *coeffMatrix* 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:
 $5E-14 \cdot \max(\dim(coeffMatrix)) \cdot r\ddot{a}kkeNorm(coeffMatrix)$

simult(coeffMatrix, 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.

$$x + 2y = 1$$

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

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

Løsningen er $x=-3$ og $y=2$.

Løs:

$$ax + by = 1$$

$$cx + dy = 2$$

Løs:

$$x + 2y = 1$$

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

$$x + 2y = 2$$

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

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

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

sin() -tast

sin(Liste1)⇒liste

I vinkeltilstanden Grader:

sin(Liste1) returnerer en liste med sinus til alle elementer i *Liste1*.

I vinkeltilstanden Nygrader:

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 Radian:

sin(kvadratMatrix1)⇒kvadratMatrix

I vinkeltilstanden Radian:

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

$$\sin \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

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

sin⁻¹() -tast

sin⁻¹(Liste1)⇒liste

I vinkeltilstanden Grader:

sin⁻¹(Liste1) returnerer en liste med de inverse sinusværdier for hvert element af *Liste1*.

I vinkeltilstanden Nygrader:

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 **arcsin(...)**.

sin⁻¹(kvadratMatrix1)⇒kvadratMatrix

I vinkeltilstanden radian og tilstanden rektangulært komplekst format:

$\sin^{-1}()$



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

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

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

$\sinh()$

Katalog >

sinh(Liste1)⇒*liste*

sinh (Liste1) returnerer en liste af de hyperbolske sinuser af hvert element af *Liste1*.

sinh(kvadratMatrix1)⇒*kvadratMatrix*

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()**.

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

$$\begin{array}{ll} \sinh(1.2) & 1.50946 \\ \sinh(\{0,1.2,3\}) & \{0,1.50946,10.0179\} \end{array}$$

I vinkeltilstanden Radian:

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

$\sinh^{-1}()$

Katalog >

sinh⁻¹(Liste1)⇒*liste*

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)⇒*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()**.

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

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

SinReg

SinReg *X*, *Y* [, [*Iterationer*],[*Periode*] [, [*Kategori*, *Medtag*]]

Beregner sinusregressionen på listerne *X* og *Y*. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146.)

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 208.

Output-variabel	Beskrivelse
stat.RegEqn	Regressionsligning: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Regressionskoefficienter
stat.Resid	Residualer fra regressionen
stat.XReg	Liste af datapunkter i den modificerede <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>

SortA

Katalog > 

SortA *Liste1* [, *Liste2*] [, *Liste3*] ...

$\{2,1,4,3\} \rightarrow list1$	$\{2,1,4,3\}$
---------------------------------	---------------

SortA *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

SortA <i>list1</i>	Done
--------------------	------

Sorterer elementerne i første argument i stigende rækkefølge.

<i>list1</i>	$\{1,2,3,4\}$
--------------	---------------

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.

$\{4,3,2,1\} \rightarrow list2$	$\{4,3,2,1\}$
---------------------------------	---------------

SortA <i>list2</i> , <i>list1</i>	Done
-----------------------------------	------

<i>list2</i>	$\{1,2,3,4\}$
--------------	---------------

<i>list1</i>	$\{4,3,2,1\}$
--------------	---------------

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 208.

SortD

Katalog >

SortD *Liste1* [, *Liste2*] [, *Liste3*] ...

$\{2,1,4,3\} \rightarrow list1$ $\{2,1,4,3\}$

SortD *Vektor1* [, *Vektor2*] [, *Vektor3*] ...

$\{1,2,3,4\} \rightarrow list2$ $\{1,2,3,4\}$

Identisk med **SortA**, 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 208.

►Sphere

Katalog >

Vektor ►Sphere

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

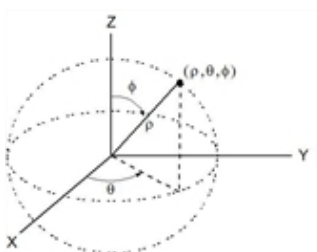
$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \text{►Sphere}$
 $\begin{bmatrix} 3.74166 & \angle 1.10715 & \angle 0.640522 \end{bmatrix}$

Viser række- eller kolonnevektoren i sfærisk form $[r \angle \theta \angle \phi]$.

Vektor skal have dimensionen 3 og kan være enten en række- eller kolonnevektor.

$\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix} \text{►Sphere}$
 $\begin{bmatrix} 3.60555 & \angle 0.785398 & \angle 0.588003 \end{bmatrix}$

Bemærk: ►Sphere er en displayformat-kommando, ikke en konverteringsfunktion. Du kan kun anvende den i slutningen af en indtastningslinje.



sqrt()

Katalog >

sqrt(*Liste1*)⇒*liste*

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

Bemærk: Se også Kvadratrodsskabelon, side 1.

stat.results

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.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.ox	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.oy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.ox1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.ox2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.Σx̄
stat.b9	stat.FBlock	stat.ṗ	stat.Σx ²	stat.Σx1
stat.b10	stat.Fcol	stat.ṗ1	stat.Σxy	stat.Σx2

stat.bList	stat.FInteract	stat. $\hat{p}^2$	stat. Σy	stat. $\bar{x}$ Diff
stat. χ^2	stat.FreqReg	stat. $\hat{p}$ Diff	stat. Σy^2	stat. $\bar{x}$ List
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg
stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat. $\bar{y}$
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat. $\hat{y}$
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat. $\hat{y}$ List
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

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])` ⇒ *udtryk*

I vinkeltilstanden Radian og tilstanden Auto:

Returnerer population standardafvigelsen af elementerne i *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

Bemærk: *Liste* skal have mindst to elementer. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208

stDevPop(*Matrix1*[,
Hyppighedsmatrix])⇒*matrix*

Returnerer en rækkevektor af populationsstandardafvigelser for kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

Bemærk: *Matrix1* skal have mindst to rækker. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208.

stDevSamp(*Liste*[,
Hyppighedsliste])⇒*udtryk*

Returnerer stikprøvestandardafvigelsen af elementerne i *liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

Bemærk: *Liste* skal have mindst to elementer. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208

stDevSamp(*Matrix1*[,
Hyppighedsmatrix])⇒*matrix*

Returnerer en rækkevektor af standardafvigelser for målingerne i kolonnerne i *Matrix1*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *Matrix1*.

Bemærk: *Matrix1* skal have mindst to rækker. Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208.

Stop

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.

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

<i>prog1</i> ()	Done
-----------------	------

<i>i</i>	5
----------	---

Store

Se → (store), side 191.

string()

string(*Udtr*)⇒*streng*

Reducerer *Udtr* og returnerer resultatet som en tegnstreng.

subMat()

subMat(*Matrix1* [, *Startrække*] [, *Startkolonne*] [, *Slutrække*] [, *Slutkolonne*]) ⇒*matrix*

Returnerer den angivne delmatrix af *Matrix1*.

Standardindstillinger: *Startrække*=1, *Startkolonne*=1, *Slutrække*=sidste række, *Slutkolonne*=sidste kolonne.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
subMat(<i>m1</i> ,2,1,3,2)	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
subMat(<i>m1</i> ,2,2)	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

sum()Katalog > **sum(Liste[, Start[, Slut]])**⇒udtrykReturnerer summen af elementer i *Liste*.*Start* og *Slut* er valgfri. De angiver en serie af elementer.

Alle ugyldige argumenter giver et ugyldigt resultat. Tomme (ugyldige) elementer i *List* ignoreres. Yderligere oplysninger om tomme elementer findes på side 208.

sum(MatrixI[, Start[, Slut]])⇒matrixReturnerer en rækkevektor med summerne af elementerne i kolonnerne i *MatrixI*.*Start* og *Slut* er valgfri. De angiver en serie af rækker.

Alle ugyldige argumenter giver et ugyldigt resultat. Tomme (ugyldige) elementer i *MatrixI* ignoreres. Yderligere oplysninger om tomme elementer findes på side 208.

sum	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
sum	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
sum	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, 2, 3$	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

sumlf()Katalog > **sumlf(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

sumIf()

Katalog > 

elementer i *Liste*, der reduceres til værdien 34.

- Et Boolsk udtryk, der indeholder symbolet ? som pladsholder for hvert element. For eksempel akkumulerer $\text{?} < 10$ kun de elementer i *Liste*, der er mindre end 10.

Når et element i *Liste* opfylder *Kriterie*, føjes elementet til den akkumulerede sum. Hvis du inkluderer *sumListe*, tilføjes det tilsvarende element fra *sumListe* til summen i stedet.

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

Tomme (ugyldige) elementer ignoreres. Yderligere oplysninger om tomme elementer findes på side 208.

Bemærk: Se også `countIf()`, side 29.

sumSeq()

Se $\Sigma()$, side 184.

system()

Katalog > 

Returnerer et system af ligninger formateret som en liste. Du kan også oprette et system ved hjælp af en skabelon.

$\text{solve}\left(\begin{cases} x+y=0 \\ x-y=8 \end{cases}, x, y\right)$	$x=4 \text{ and } y=-4$
---	-------------------------

T

T (transponere)

Katalog > 

$Matrix1^T \Rightarrow matrix$

Returnerer den kompleks konjugerede transponerede *Matrix1*.

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

tan()



tan(Liste1)⇒liste

I vinkeltilstanden Grader:

tan(Liste1) returnerer en liste med tangens til alle elementer i *Liste1*.

I vinkeltilstanden Nygrader:

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 Radian:

tan(kvadratMatrix1)⇒kvadratMatrix

I vinkeltilstanden Radian:

Returnerer matrixtangens til *kvadratMatrix1*. Dette er ikke det samme som at beregne tangens for hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$$\tan \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$$

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

tan⁻¹()



tan⁻¹(Liste1)⇒list

I vinkeltilstanden Grader:

tan⁻¹(Liste1) returnerer en liste med den inverse tangens til hvert element af *Liste1*.

$$\tan^{-1}(1) \quad 45$$

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

I vinkeltilstanden Nygrader:

$$\tan^{-1}(1) \quad 50$$

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

I vinkeltilstanden Radian:

$$\tan^{-1}(\{0,0,2,0,5\}) \quad \{0,0.197396,0.463648\}$$

tan⁻¹(kvadratMatrix1)⇒kvadratMatrix

I vinkeltilstanden Radian:

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

$$\tan^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

$\tan^{-1}()$



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

$\tanh()$

Katalog >

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

$\tanh(\text{Liste1})$ returnerer en liste med den hyperbolske tangens til hvert enkelt element i *Liste1*.

$\tanh(\text{kvadratMatrix1}) \Rightarrow \text{kvadratMatrix}$

Returnerer den matrixhyperbolske tangens af *kvadratMatrix1*. Dette er ikke det samme som at beregne den hyperbolske tangens for hvert element. Oplysninger om beregningsmetoden findes i $\cos()$.

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

I vinkeltilstanden Radian:

$$\tanh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

$\tanh^{-1}()$

Katalog >

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

$\tanh^{-1}(\text{Liste1})$ 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 (...)**.

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

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:

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

I vinkeltilstanden Radian og rektangulært komplekst format:

$$\tanh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.099353+0.164058 \cdot i & 0.267834-1.4908 \\ -0.087596-0.725533 \cdot i & 0.479679-0.94730 \\ 0.511463-2.08316 \cdot i & -0.878563+1.7901 \end{bmatrix}$$

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

tCdf()

tCdf(*nedreGrænse*,*øvreGrænse*,*fg*)⇒*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*.

Text

Text*promptStreng*[, *DispFlag*]

Programmeringskommando: Standser
programmet midlertidigt og viser
tegnstrengen *promptStreng* i en
dialogboks.

Når en bruger vælger **OK**, fortsætter
programudførelsen.

Det valgfrie argument *flag* kan være
et hvilket som helst udtryk.

- Hvis *DispFlag* udelades eller
beregnes til **1**, føjes
tekstmeddelelsen til Regner-
historikken.
- Hvis *DispFlag* udelades eller
beregnes til **0**, føjes
tekstmeddelelsen ikke til
historikken.

Hvis programmet skal bruge et
skriftligt svar fra brugeren, henvises til
Request, side 127 eller **RequestStr**, side
127.

Bemærk: Du kan bruge denne
kommando i et brugerdefineret
program men ikke i en funktion.

Definer et program, der standser midlertidigt for
at vise hvert af fem tilfældige tal i en dialogboks.

I Prgm...EndPrgm-skabelonen udfyldes hver linje
ved at trykke på  i stedet for på . På
computerens tastatur skal du holde **Alt** nede og
trykke på **Enter**.

```
Define text_demo()=Prgm
```

```
  For i,1,5
```

```
    strinfo:="Random number " & string  
    (rand(i))
```

```
    Text strinfo
```

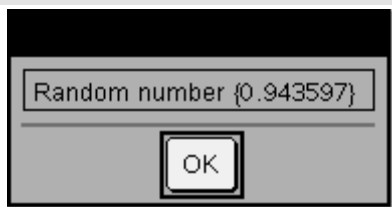
```
  EndFor
```

```
EndPrgm
```

Kør programmet:

```
text_demo()
```

Eksempel på en dialogboks:

**tInterval**

tInterval *Liste[,Frekv[,CNiveau]]*

(Datalisteinput)

tInterval $\bar{x}, s_x, n[, CNiveau]$

(Sammenfatning, stat input)

Beregner et *t* konfidensinterval En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

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

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 variablen *stat.results*. (side 146.)

Puljet=1 puljer varianser. *Puljet=0* puljer ikke varianser.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

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. $sx1$, stat. $sx2$	Stikprøve standardafvigelse for <i>Liste 1</i> og <i>Liste 2</i>
stat. $n1$, stat. $n2$	Antal målinger i datasekvenserne
stat.sp	Den puljede standardafvigelse. Beregnes, når <i>Puljet</i> = 1.

tPdf(*XVærdi*,*fg*) $\Rightarrow$ *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*.

Returner sporet (sum af alle elementer på hoveddiagonalen) af *kvadratMatrix*.

Try

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 fejltrening. En liste med fejkoder findes i "*Fejkoder og fejlmeddelelser*," side 218.

blok1 og *blok2* kan enten være en enkelt sætning eller en række sætninger adskilt med kolon.

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

Eksempel 2

Du kan se kommandoerne **Try**, **ClrErr** og **PassErr** i funktion ved at indtaste programmet *eigenvals()* vist til højre. Kør programmet ved at eksekvere hvert af de følgende udtryk.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, [-1 \quad 2 \quad -3.1]\right)$$

Bemærk: Se også **ClrErr**, side 22, og **PassErr**, side 110.

```
Define prog1()  
  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
 egenværdier 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 variabelen *stat.results*. (side 146.)

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 208.

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

(Datalisteinput)

tTest_2Samp $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2$ [, *Hypot*
[, *Puljet*]]

(Sammenfatning, stat input)

Beregner en to-prøvers *t* test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 146.)

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 208.

Output-variabel	Beskrivelse
stat.t	Standardnormalværdi beregnet som differens af gennemsnit
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat.df	Frihedsgrader for t-statistik
stat. $\bar{x}1$, stat. $\bar{x}2$	Middelværdi af stikprøver for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Størrelse på stikprøverne
stat.sp	Den puljede standardafvigelse. Beregnes, når <i>Puljet</i> =1.

tvmFV()

tvmFV(*N, I, PV, Pmt, [PpY], [CpY],
[PmtAt]*) $\Rightarrow$ værdi

tvmFV(120,5,0,-500,12,12)

77641.1

tvmFV()Katalog > 

Finansfunktion, der beregner penges fremtidige værdi.

Bemærk: Argumenter, der bruges i TVM-funktionerne er beskrevet i tabellen over TVM-argumenter, side 161. Se også **amortTbl()**, side 6.

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 161. Se også **amortTbl()**, side 6.

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 161. Se også **amortTbl()**, side 6.

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 161. Se også **amortTbl()**, side 6.

tvmPV(*N, I, Pmt, FV, [PpY], [CpY], [PmtAt]*) ⇒ *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 161. Se også **amortTbl()**, side 6.

TVM-argument*	Beskrivelse	Datatype
N	Antal betalingsperioder	reelt tal
I	Årlig rentesats	reelt tal
PV	Nutidsværdi	reelt tal
Pmt	Betalingsbeløb	reelt tal
FV	Fremtidsværdi	reelt tal
PpY	Betalinger pr år, standardværdi=1	heltal > 0
CpY	Rentetilskrivninger per år, standardværdi=1	heltal > 0
PmtAt	Betaling, der forfalder ved starten af hver periode, standardværdi=slut	heltal (0=slut, 1=start)

* Disse argumentnavne for tidsdiskonterede pengeværdier svarer til til TVM-variabelnavnene (som f.eks. **tvm.pv** og **tvm.pmt**), der anvendes af *Calculator* applikationens FinansRegner. Finansfunktioner gemmer dog ikke deres argumentværdier eller resultater i TVM-variablene.

TwoVar *X, Y*, [*Frekv*] [, *Kategori, Medtag*]

Beregner statistik med to variable. En sammenfatning af resultaterne lagres i *stat.results* variable. (side 146.)

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 208.

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.

unLock

Catalog > 

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

unLock*Var*.

Oplåser de angivne variable eller variabelgrupper. Låste variable kan ikke redigeres eller slettes.

Se **Lock**, side 85 og **getLockInfo()**, side 63.

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

varPop()Katalog > **varPop(Liste[, hyppighedsliste])**⇒udtrykReturnerer populationsvariansen af *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

Bemærk: *Liste* skal indeholde mindst to elementer.

Hvis et element i en af listerne er tom (ugyldig), ignoreres dette element, og det tilsvarende element i den anden liste ignoreres også. Yderligere oplysninger om tomme elementer findes på side 208.

varSamp()Katalog > **varSamp(Liste[, Hyppighedsliste])**⇒udtrykReturnerer stikprøvevariansen for *Liste*.

Hvert *hyppighedsliste*-element tæller antallet af konsekutive forekomster de tilsvarende elementer i *Liste*.

Bemærk: *Liste* skal indeholde mindst to elementer.

Hvis et element i en af listerne er tom (ugyldig), ignoreres dette element, og det tilsvarende element i den anden liste ignoreres også. Yderligere oplysninger om tomme elementer findes på side 208.

varSamp(MatrixI[, Hyppighedsmatrix])⇒matrix

Returnerer en rækkevektor med stikprøvevariansen for hver kolonne i *MatrixI*.

Hvert *Hyppighedsmatrix*-element tæller antallet af konsekutive forekomster af det tilsvarende element i *MatrixI*.

varSamp	$\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{pmatrix}$	$[4.75 \ 1.03 \ 4]$
varSamp	$\begin{pmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{pmatrix} \cdot \begin{pmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{pmatrix}$	$[3.91731 \ 2.08411]$

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 208.

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 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.

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"

warnCodes ()

warnCodes(*Expr1*, *StatusVar*)⇒*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 226 for at få en oversigt over fejlkoder og tilhørende meddelelser.

when()

when(*Betingelse*, *sandtResultat* [, *falskResultat*][, *ubekendtResultat*])
⇒ *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.

Anvend en **undef** *falskResultat* til at definere et udtryk, der kun tegner grafen til et interval.

when() er nyttig til definition af rekursive funktioner.

$\text{when}(x < 0, x + 3) x = 5$	undef
-------------------------------------	-------

$\text{when}(n > 0, n \cdot \text{factorial}(n-1), 1) \rightarrow \text{factorial}(n)$	Done
$\text{factorial}(3)$	6
3!	6

While *Betingelse*
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 $sum_of_recip(n)$ =Func	
Local $i,tempsum$	
$1 \rightarrow i$	
$0 \rightarrow tempsum$	
While $i \leq n$	
$tempsum + \frac{1}{i} \rightarrow tempsum$	
$i + 1 \rightarrow i$	
EndWhile	
Return $tempsum$	
EndFunc	
Done	
$sum_of_recip(3)$	$\frac{11}{6}$

X

*BoolskUdtryk1***xor***BoolskUdtryk2*
returnerer *Boolsk udtryk*

*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 15.

Bemærk: Se **or**, side 108.

Heltal1 **xor** *Heltal2* $\Rightarrow$ *heltal*

true xor true	false
5>3 xor 3>5	true

I hexadecimal tilstand:

Vigtigt: Tallet nul, ikke bogstavet O.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

I binær tilstand:

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.

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.

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

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

Bemærk: Se **or**, side 108.

Z

zInterval

zInterval $\sigma, \text{Liste[, Hyppighed[, CNiveau]]}$

(Datalisteinput)

zInterval $\sigma, \bar{x}, n [, CNiveau]$

(Sammenfatning, stat input)

Beregner et *z* konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 146.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval for en ukendt populationsmiddelværdi

Output-variabel	Beskrivelse
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

Katalog > 

zInterval_1Prop x, n [,CNiveau]

Beregner én-proportion z konfidensinterval.
En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 146.)

x er et ikke-negativt heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelingssandsynlighed
stat. $\hat{p}$	Den beregnede brøkdelt af succeser
stat.ME	Fejlmargen
stat.n	Antal stikprøver i datasekvens

zInterval_2Prop

Katalog > 

zInterval_2Prop $x1, n1, x2, n2$ [,CNiveau]

Beregner et to-proportion z konfidensinterval. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 146.)

$x1$ og $x2$ er ikke-negative heltal.

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelings sandsynlighed
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

Katalog > 

zInterval_2Samp σ_1, σ_2 , *Liste1*, *Liste2*
[,*Hyppighed1*], [*Hyppighed2*, [*CNiveau*]]

(DataListeininput)

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 variablen *stat.results*. (side 146.)

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

Output-variabel	Beskrivelse
stat.CLower, stat.CUpper	Konfidensinterval med et konfidensniveau for fordelings sandsynlighed
stat. $\bar{x}1 - \bar{x}2$	Middelværdi stikprøver i i datasekvenserne fra den vilkårlige normalfordeling
stat.ME	Fejlmargen
stat. $\bar{x}1$, stat. $\bar{x}2$	Middelværdi stikprøver i datasekvenserne fra den vilkårlige normalfordeling
stat. $\sigma x1$, stat. $\sigma x2$	Stikprøve standardafvigelse for <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Antal stikprøver i datasekvenserne

Output-variabel	Beskrivelse
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 variablen *stat.results*. (side 146.)

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 208.

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>
stat.sx	Standardmåleafvigelse for datasekvensen. Returneres kun for <i>Data</i> input.
stat.n	Stikprøvens størrelse

zTest_1Prop

Katalog > 

zTest_1Prop $p_0, x, n, [\text{Hypot}]$

Beregner en én-proportion z test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 146.)

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 208.

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 $x1, n1, x2, n2[, Hypot]$

Beregner en toproportional z test. En sammenfatning af resultaterne lagres i variablen *stat.results*. (side 146.)

$x1$ 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 208.

Output-variabel	Beskrivelse
stat.z	Standardnormalværdi beregnet som differens af proportioner
stat.PVal	Mindste signifikansniveau, ved hvilket 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

Katalog > 

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 146.)

Test $H_0: \mu_1 = \mu_2$, mod af de følgende:

Til $H_a: \mu_1 < \mu_2$, sæt *Hypot*<0

til $H_a: \mu_1 \neq \mu_2$ (standard), sæt *Hypot*=0

til $H_a: \mu_1 > \mu_2$, *Hypot*>0

Oplysninger om effekten af tomme elementer i en liste findes "Tomme (ugyldige) elementer," side 208.

Output-variabel	Beskrivelse
stat.z	Standardnormalværdi beregnet som differens af gennemsnit
stat.PVal	Mindste signifikansniveau, ved hvilket nul-hypotesen kan forkastes
stat. $\bar{x}_1$, stat. $\bar{x}_2$	Middelværdi af stikprøver for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.sx1, stat.sx2	Stikprøve standardafvigelse for datasekvenserne i <i>Liste 1</i> og <i>Liste 2</i>
stat.n1, stat.n2	Størrelse på stikprøverne

Symboler

+ (adder)		+ -tast
Returnerer summen af de to argumenter.	56	56
	56+4	60
	60+4	64
	64+4	68
	68+4	72

$Liste1 + Liste2 \Rightarrow liste$

$Matrix1 + Matrix2 \Rightarrow matrix$

Returnerer en liste (eller matrix), der indeholder summerne af tilsvarende elementer i *Liste1* og *Liste2* (eller *Matrix1* og *Matrix2*).

Argumenternes dimensioner må være ens.

$15 + \{10, 15, 20\}$	$\{25, 30, 35\}$
$\{10, 15, 20\} + 15$	$\{25, 30, 35\}$

Bemærk: Anvend .+ (punktum plus) til at addere et udtryk til hvert element.

$20 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
---	--

- (subtraher)		- -tast
---------------	--	---------

$Liste1 - Liste2 \Rightarrow liste$

$Matrix1 - Matrix2 \Rightarrow matrix$

Subtraherer hvert element i *Liste2* (eller *Matrix1*) fra det tilsvarende element i *Liste1* (eller *Matrix1*), og returnerer resultaterne.

Argumenternes dimensioner må være ens.

$15 - \{10, 15, 20\}$	$\{5, 0, -5\}$
$\{10, 15, 20\} - 15$	$\{-5, 0, 5\}$

-(subtraher)**-**-tast

Bemærk: Anvend .- (punktum minus) til at subtrahere et udtryk fra hvert element.

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

·(multiplicer)**×**-tast

Returnerer produktet af de to argumenter.

$Liste1 \cdot Liste2 \Rightarrow liste$

Returnerer en liste med produkterne af de tilsvarende elementer i *Liste1* og *Liste2*.

Listernes dimensioner må være ens.

$Matrix1 \cdot Matrix2 \Rightarrow matrix$

Returnerer matrixproduktet af *Matrix1* og *Matrix2*.

Antallet af kolonner i *Matrix1* skal være lig med antallet af rækker i *Matrix2*.

Bemærk: Anvend .· (punktum gangetegn) til at gange et udtryk med hvert element.

/ (divider)**÷**-tast

Bemærk: Se også **Brøkskabelon**, side 1.

$Liste1 / Liste2 \Rightarrow liste$

Returnerer en liste med kvotienterne af *Liste1* divideret med *Liste2*.

$$\frac{\begin{Bmatrix} 1, 2, 3 \\ 4, 5, 6 \end{Bmatrix}}{\begin{Bmatrix} 1, 2, 3 \\ 4, 5, 6 \end{Bmatrix}} \quad \left\{ 0.25, \frac{2}{5}, \frac{1}{2} \right\}$$

Listernes dimensioner må være ens.

Bemærk: Anvend ./ (punktum divisionstegn) til at dividere et udtryk med hvert element.

^ (potens)**^**-tast

$Liste1 \wedge Liste2 \Rightarrow liste$

Returnerer det første argument opløftet til potensen af det andet argument.

$\wedge$ (potens)

$\boxed{\wedge}$ -tast

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.

kvadratMatrix1 $\wedge$ *heltal* $\Rightarrow$ matrix

Returnerer *kvadratMatrix1* opløftet til *heltal*spotensen.

kvadratMatrix1 skal være en kvadratisk matrix.

Hvis *heltal* = -1, beregnes den inverse matrix.

Hvis *heltal* < -1, beregnes den inverse matrix til en passende positiv potens.

$$\{1, 2, 3, 4\}^{-2} \quad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \quad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

x^2 (kvadrat)

$\boxed{x^2}$ -tast

Returnerer kvadratet af argumentet.

*Liste1*² $\Rightarrow$ *liste*

Returnerer en liste med kvadraterne på elementerne i *Liste1*.

*kvadratMatrix1*² $\Rightarrow$ matrix

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.

$$4^2 \quad 16$$

$$\{2, 4, 6\}^2 \quad \{4, 16, 36\}$$

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2 \quad \begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^{\wedge 2} \quad \begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$$

.+ (punktum plustegn)

  -taster

$Matrix1 .+ Matrix2 \Rightarrow matrix$

$Matrix1 .+ Matrix2$ returnerer en matrix, der er summen af hvert par tilsvarende elementer i $Matrix1$ og $Matrix2$.

.- (punktum minus.)

  -taster

$Matrix1 .- Matrix2 \Rightarrow matrix$

$Matrix1 .- Matrix2$ returnerer en matrix, der er differensen mellem hvert par af tilsvarende elementer i $Matrix1$ og $Matrix2$.

.· (punktum mult.)

  -taster

$Matrix1 .· Matrix2 \Rightarrow matrix$

$Matrix1 .· Matrix2$ returnerer en matrix, der er produktet af hvert par tilsvarende elementer i $Matrix1$ og $Matrix2$.

. / (punktum divider)

  -taster

$Matrix1 . / Matrix2 \Rightarrow matrix$

$Matrix1 . / Matrix2$ returnerer en matrix, der er kvotient af hvert par af tilsvarende elementer i $Matrix1$ og $Matrix2$.

.^ (punktum potens)

  -taster

$Matrix1 ^ Matrix2 \Rightarrow matrix$

$Tal.^Matrix1 \Rightarrow matrix$

$Matrix1 .^ Matrix2$ returnerer en matrix, hvor hvert element i $Matrix2$ er eksponent til det tilsvarende element i $Matrix1$.

-(neger)

 -tast

$-Liste1 \Rightarrow liste$

I binær tilstand:

-(neger)

-tast

-Matrix1 $\Rightarrow$ *matrix*

Returnerer negationen til argumentet.

Ved en liste eller matrix, returneres alle elementer negeret.

Hvis argumentet er et binært eller hexadecimalt heltal, giver negationen 2's komplement.

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

Liste1 % $\Rightarrow$ *liste*

Matrix1 % $\Rightarrow$ *matrix*

argument

Returnerer 100

For en liste eller matrix returneres en liste eller matrix med hvert element divideret med 100.

Bemærk: Sådan gennemtvinges et tilnærmet resultat,

Håndholdt: Tryk på  .

Windows®: Tryk **Ctrl+Enter**.

Macintosh®: Tryk på **⌘+Enter**.

iPad®: Hold **ENTER** nede, og vælg .

13%	0.13
-----	------

$\{\{1,10,100\}\}\%$	$\{0.01,0.1,1.\}$
----------------------	-------------------

= (lig med)

-tast

Udtr1 = *Udtr2* $\Rightarrow$ *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*.

Alt andet returnerer en reduceret form af ligningen.

For lister og matricer returneres sammenligninger element for element.

Eksempelfunktion, der anvender matematiske testsymboler: =, $\neq$, $\leq$, $<$, $>$, $\geq$

```
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
```

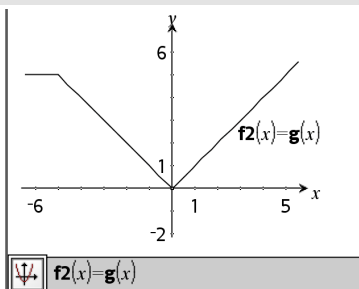
Done

Tegnet resultat af grafen g(x)

= (lig med)

 -tast

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



≠ (forskellig fra)

  -taster

$Udtr1 \neq Udtr2 \Rightarrow$ Boolsk udtryk

Se "=" (lig med)-eksemplet.

$Liste1 \neq Liste2 \Rightarrow$ Boolsk liste

$Matrix1 \neq Matrix2 \Rightarrow$ Boolsk matrix

Returnerer true, hvis $Udtr1$ bestemmes til at være forskellig fra $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være lig med $Udtr2$.

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive /=

< (mindre end)

  taster

$Udtr1 < Udtr2 \Rightarrow$ Boolsk udtryk

Se "=" (lig med)-eksemplet.

$Liste1 < Liste2 \Rightarrow$ Boolsk liste

$Matrix1 < Matrix2 \Rightarrow$ Boolsk matrix

Returnerer true, hvis $Udtr1$ bestemmes til at være mindre end $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være større end $Udtr2$.

< (mindre end)

  **taster**

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)

ctrl **=** taster

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

Se “=” (lig med)-eksemplet.

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

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

Returnerer true, hvis $Udtr1$ bestemmes til at være større end eller lig med $Udtr2$.

Returnerer false, hvis $Udtr1$ bestemmes til at være mindre end $Udtr2$.

Alt andet returnerer en reduceret form af uligheden.

For lister og matricer returneres sammenligninger element for element.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive $\geq$

$\Rightarrow$ (medfører)

ctrl **=** -taster

$BoolskUdtryk1 \Rightarrow BoolskUdtryk2$
returnerer *Boolsk udtryk*

$5 > 3$ or $3 > 5$	true
--------------------	------

$5 > 3 \Rightarrow 3 > 5$	false
---------------------------	-------

$BoolskListe1 \Rightarrow BoolskListe2$
returnerer *Boolsk liste*

3 or 4	7
------------	-----

$3 \Rightarrow 4$	-4
-------------------	------

$BoolskMatrix1 \Rightarrow BoolskMatrix2$
returnerer *Boolsk matrix*

$\{1, 2, 3\}$ or $\{3, 2, 1\}$	$\{3, 2, 3\}$
--------------------------------	---------------

$\{1, 2, 3\} \Rightarrow \{3, 2, 1\}$	$\{-1, -1, -3\}$
---------------------------------------	------------------

$Heltal1 \Rightarrow Heltal2$ returnerer *Heltal*

Beregner udtrykket **not** <argument1> **or** <argument2> og returnerer true, false eller en forenklet form af ligningen.

For lister og matricer returneres sandhedsværdier element for element.

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive $\Rightarrow$

⇔ (ensbetydende med)

  -taster

BoolskUdtryk1 ⇔ *BoolskUdtryk2*
returnerer *Boolsk udtryk*

$5 > 3 \text{ xor } 3 > 5$ true

BoolskListe1 ⇔ *BoolskListe2*
returnerer *Boolsk liste*

$5 > 3 \Leftrightarrow 3 > 5$ false

$3 \text{ xor } 4$ 7

BoolskMatrix1 ⇔ *BoolskMatrix2*
returnerer *Boolsk matrix*

$3 \Leftrightarrow 4$ -8

$\{1, 2, 3\} \text{ xor } \{3, 2, 1\}$ $\{2, 0, 2\}$

$\{1, 2, 3\} \Leftrightarrow \{3, 2, 1\}$ $\{-3, -1, -3\}$

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 $\Leftrightarrow$

! (fakultet)

 -tast

Liste1! ⇒ *liste*

$5!$ 120

Matrix1! ⇒ *matrix*

$\{\{5, 4, 3\}\}!$ $\{120, 24, 6\}$

Returnerer fakultetværdien af argumentet.

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

For en liste eller matrix returneres en liste eller matrix af elementerne.

& (tilføj)

  -taster

Streng1 & *Streng2* ⇒ *streng*

"Hello "&"Nick" "Hello Nick"

Returnerer en tekststreng, der er *Streng2* adderet til *Streng1*.

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

 $\sqrt{}$ (kvadratrod)ctrl x^2 -taster
 $\sqrt{Liste1} \Rightarrow liste$

Returnerer kvadratroden af argumentet.

For en liste returneres kvadratrødderne af alle elementer i *Liste1*.

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

Bemærk: Se også **Kvadratrods-kabelon**, side 1.

 $\Pi()$ (prodSeq)Katalog > 
 $\Pi(Udtr1, Var, Lav, Høj) \Rightarrow udtryk$

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

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 $Høj < Lav-1$

$$\prod_{k=4}^3 (k) = 1$$

De anvendte produktformler stammer fra følgende reference:

Ronald L. Graham, Donald E. Knuth og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) = 6$$

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \cdot \prod_{k=2}^4 \left(\frac{1}{k}\right) = \frac{1}{4}$$

$\Sigma(Udtr1, Var, Lav, Høj) \Rightarrow udtryk$

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 $Høj < Lav-1$

$$\sum_{k=4}^3 (k) = 0$$

De anvendte summationsformler stammer fra følgende reference:

Ronald L. Graham, Donald E. Knuth og Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\sum_{k=4}^1 (k) = -5$$

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) = 4$$

$\Sigma Int()$

$\Sigma Int(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [afrundVærdi]) \Rightarrow værdi$

$\Sigma Int(1, 3, 12, 4, 75, 20000, , 12, 12) = -218.11$

$\Sigma Int(NPmt1, NPmt2, amortTabel) \Rightarrow værdi$

Amortiseringsfunktion, der beregner summen af renter under en angivet række af betalinger.

NPmt1 og *NPmt2* definerer starten og slutningen af betalingsrækken.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* og *PmtAt* er beskrevet i tabellen over TVM-argumenter på side 161.

- Hvis du udelader *Pmt*, sættes den som standard til *Pmt=tvmpmt(N, I, PV, FV, PpY, CpY, PmtAt)*.

<i>tbl:=amortTbl(12,12,4.75,20000,,,12,12)</i>				
	0	0.	0.	20000.
1	-79.17	-1630.69	18369.3	
2	-72.71	-1637.15	16732.2	
3	-66.23	-1643.63	15088.5	
4	-59.73	-1650.13	13438.4	
5	-53.19	-1656.67	11781.7	
6	-46.64	-1663.22	10118.5	
7	-40.05	-1669.81	8448.7	
8	-33.44	-1676.42	6772.28	
9	-26.81	-1683.05	5089.23	
10	-20.14	-1689.72	3399.51	
11	-13.46	-1696.4	1703.11	
12	-6.74	-1703.12	-0.01	
$\Sigma Int(1, 3, tbl) = -218.11$				

- 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.

ΣInt(*NPmt1*,*NPmt2*,*amortTabel*)
beregner summen af renten baseret på amortiseringstabellen *amortTabel*.
amortTabel-argumentet skal være en matrix i formen beskrevet under **amortTbl()**, side 6.

Bemærk: Se også ΣPrn() nedenfor og Bal(), side 14.

ΣPrn(*NPmt1*,*NPmt2*,*N*,*I*,*PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*afrundVærdi*])⇒*værdi*

ΣPrn(1,3,12,4.75,20000,,,12,12) -4911.47

ΣPrn

(*NPmt1*,*NPmt2*,*amortTabel*)⇒*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 161.

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

tbl:=amortTbl(12,12,4.75,20000,,,12,12)

0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

ΣPrn(1,3,*tbl*) -4911.47

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 6.

Bemærk: Se også ΣInt() ovenfor og Bal(), side 14.

(henvisning)

  -taster

varNavnestreng

Opretter eller kalder variablen xyz.

Kalder variablen, hvis navn er *varNavnestreng*. Dermed kan du anvende strenge til at oprette variabelnavne fra en funktion.

$10 \rightarrow r$	10
"r" $\rightarrow s1$	"r"
#s1	10

Returnerer værdien af variablen (r), hvis navn er lagret i variablen s1.

E (videnskabelig notation)

 -tast

mantisseEksponent

Indtaster et tal i videnskabelig notation. Tallet fortolkes som *mantisse* × 10^{eksponent}.

23000.	23000.
2300000000.+4.1E15	4.1E15
3·10 ⁴	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)

1-tast

Udtr1 $\Rightarrow$ *udtryk*

I vinkeltilstanden Grader, Nygrader eller Radianer:

Udtr1 $\Rightarrow$ *udtryk*

Liste1 $\Rightarrow$ *liste*

Matrix1 $\Rightarrow$ *matrix*

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)

1-tast

Liste1 $\Rightarrow$ *liste*

I vinkeltilstanden Grader, Nygrader eller Radianer:

Matrix1 $\Rightarrow$ *matrix*

Denne funktion giver mulighed for at angive en vinkel i radianer, mens programmet er i vinkeltilstanden Grader eller Nygrader.

I vinkeltilstanden Grader ganges argumentet med $180/\pi$.

I vinkeltilstanden Radian, returneres argument uændret.

I vinkeltilstanden Nygrader ganges argumentet med $200/\pi$.

Tip: Anvend **r**, hvis du vil fremtvinge radianer i en funktionsdefinition uanset den aktuelle tilstand, når funktionen anvendes.

Bemærk: Du kan indsætte dette symbol fra computerens tastatur ved at skrive @r.

° (grader)

1-tast

Liste $l^\circ \Rightarrow$ *liste*

I vinkeltilstanden Grader, Nygrader eller Radianer:

Matrix $l^\circ \Rightarrow$ *matrix*

I vinkeltilstanden Radianer:

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.

°, ', " (grader/minutter/sekunder)

ctrl -taster

gg°*mm*'*ss*.*ss*" $\Rightarrow$ *udtryk*

I vinkeltilstanden Grader:

*gg*Et positivt eller negativt tal

25°13'17.5" 25.2215

*mm*Et ikke-negativt tal

25°30' 51
2

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 (').

∠ (vinkel)

ctrl -taster

[*Radius*, ∠_θ *Vinkel*] $\Rightarrow$ *vektor*
(polært input)

I vinkeltilstanden Radian og vektorformatet indstillet til:

rektangulær

∠ (vinkel)

ctrl  -taster

[Radius, ∠θ_Vinkel, Z_Koordinat]
⇒vektor
(cylindrisk input)

cylindrisk

[Radius, ∠θ_Vinkel, ∠θ_Vinkel]
⇒vektor
(sfærisk input)

sfærisk

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 @<.

(Størrelse ∠ Vinkel)⇒kompleksVærdi
(polært input)

I vinkeltilstanden radian og rektangulært komplekst format:

Indtaster en kompleks værdi i den polære form ($r\angle\theta$). *Vinkel* fortolkes efter den aktuelle indstilling af vinkeltilstand.

$$5+3\cdot i - \left(10 \angle \frac{\pi}{4}\right) \quad -2.07107-4.07107\cdot i$$

_ (understregning som et tomt element)

Se |“Tomme (ugyldige) elementer” side 208, .

_ (understregning som en enhedsbenævnelse)

ctrl  -taster

*Variabel*_

Det antages, at z er udefineret:

Når *Variabel* ikke har nogen værdi, behandles den, som om den repræsenterer et komplekst tal. Uden_ behandles variabelen som reel.

$\text{real}(z)$	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.

10^()

Katalog > 

10^ (ListeI)⇒liste

Returnerer 10 opløftet til potensen af argumentet.

I en liste returneres 10 opløftet til potensen af elementerne i *Liste1*.

10^(*kvadratMatrix1*) ⇒ *kvadratmatrix*

Returnerer 10 opløftet til potensen af *kvadratMatrix1*. Dette er ikke det samme som at beregne 10 opløftet til potensen af hvert element. Oplysninger om beregningsmetoden findes i **cos()**.

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} = \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

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

Liste1 ^-1 ⇒ *liste*

Returnerer den reciprokke værdi af argumentet.

For en liste returneres de reciprokke værdier af elementerne i *Liste1*.

kvadratMatrix1 ^-1 ⇒ *kvadratMatrix*

Returnerer den inverse værdi af *kvadratMatrix1*.

kvadratMatrix1 skal være en ikke-singulær kvadratisk matrix.

Udtryk | *BoolskUdtryk1*
[**and***BoolskUdtryk2*]...

Udtryk | *BoolskUdtryk1*
[**or***BoolskUdtryk2*]...

("|")-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:

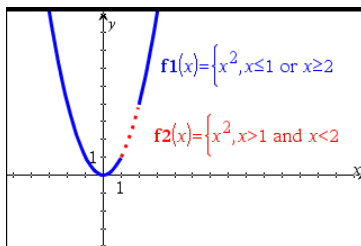
- Substitutioner
- Intervalafgrænsninger
- Udelukkelser

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.



Udelukkelser benytter "forskellig fra"-kommandoen (/= eller $\neq$) til at udelukke en specifik værdi fra reduktionen.

$$\text{solve}(x^2-1=0, x) | x \neq 1 \quad x = -1$$

→ (lagring)

Bemærk: Du kan indsætte denne operator fra tastaturet ved at skrive `=:` som en genvej. Skriv for eksempel `pi/4 =:` **myvar**.

:= (tildel)

  -taster

Var := Liste

Var := Matrix

Funktion(Param1,...) := Udr

Funktion(Param1,...) := Liste

Funktion(Param1,...) := Matrix

© (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 $g(n)$ = Func

© Declare variables

Local $i, result$

$result := 0$

For $i, 1, n, 1$ © Loop n times

$result := result + i^2$

EndFor

Return $result$

EndFunc

Done

$g(3)$

14

0b, 0h

  -taster,   -taster

0b *binærtTal*

I decimal tilstand:

0b10+0hF+10

27

0h *hexadecimaltTal*

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).

I binær tilstand:

0b10+0hF+10

0b11011

Resultatet vises i den valgte tilstand for talsystem.

I hexadecimal tilstand:

0b10+0hF+10

0h1B

TI-Nspire™ CX II - Tegn kommandoer

Dette er et supplerende dokument til TI-Nspire™ Referencevejledningen og TI-Nspire™ CAS Referencevejledningen. Alle TI-Nspire™ CX II-kommandoer vil blive inkorporerede og offentliggjort i version 5.1 i TI-Nspire™ Referencevejledningen og TI-Nspire™ CAS Referencevejledningen.

Grafikprogrammering

Nye kommandoer er blevet tilføjet på TI-Nspire™ CX II håndholdte og TI-Nspire™ desktop applikationer til grafisk programmering.

TI-Nspire™ CX II håndholdte skifter til denne grafiktilstand, mens du udfører grafikkommandoer og skifter tilbage til den kontekst, hvor programmet blev udført efter programmets afslutning.

Skærmen viser "Running ..." i øverste linje mens programmet udføres. Det vil vise "Finished", når programmet fuldendes. Enhver tastetryk vil tage systemet ud af grafikfunktionen.

- Overgangen til grafiktilstand udløses automatisk, når en af Draw (grafik) - kommandoerne opstår under udførelsen af TI-Basic-programmet.
- Denne overgang vil kun ske, når der udføres et program fra lommeregneren; i et dokument eller en lommeregner i scratchpad.
- Overgangen fra grafiktilstand sker ved afslutning af programmet.
- Grafiktilstanden er kun tilgængelig på visninger på TI-Nspire™ CX II håndholdte og TI-Nspire™ CX II håndholdte. Det betyder, at den ikke er tilgængelig i computerdokumentvisningen på skrivebordet eller iOS.
 - Hvis der opstår en grafikkommando under udførelsen af et TI-Basic-program fra den forkerte kontekst, vises en fejlmeddelelse, og TI-Basic-programmet afsluttes.

Grafikskærm

Grafikskærmen vil indeholde en overskrift øverst på skærmen, der ikke kan skrives til af grafikkommandoer.

Grafikskærmens tegningsområde bliver ryddet (farve = 255.255.255), når grafikskærmen initialiseres.

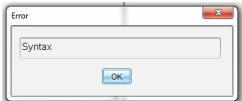
Grafikskærm	Standard
Højde	212
Bredde	318
Farve	hvid: 255,255,255

Standardvisning og indstillinger

- Statusikonerne i øverste bjælke (batteristatus, tryk-til-test-status, netværksindikator osv.) vises ikke, mens et grafikprogram kører.
- Standard tegningsfarve: Sort (0,0,0)
- Standard pennefacon - normal, glat
 - Tykkelse: 1 (tynd), 2 (normal), 3 (tykke)
 - Typografi 1 (glat), 2 (punkteret), 3 (stiplet)
- Alle tegningskommandoer bruger de nuværende indstillinger for farve og pen enten standardværdier eller dem, der blev indstillet via TI-Basic-kommandoer.
- Tekstens skrifttype er fast og kan ikke ændres.
- Enhver udgang til grafiskskærmen tegnes inden for et klipvindue, som er størrelsen på grafiskskærmens tegneområde. Enhver tegnet output, der strækker sig uden for dette klippede grafiske skærmområde, tegnes ikke. Der vises ingen fejlmeddelelse.
- Alle x, y-koordinater, der er angivet til tegningskommandoer, er defineret således, at 0,0 er øverst til venstre i grafiskskærmens tegneområde.
 - **Undtagelser:**
 - **DrawText** bruger koordinaterne som nederste venstre hjørne af grænsekassen til teksten.
 - **SetWindow** bruger nederste venstre hjørne af skærmen
- Alle parametre for kommandoerne kan gives som udtryk, der evalueres til et tal, som derefter afrundes til nærmeste heltal.

Fejlmeddelelser på grafikskærmen

Hvis validering mislykkes, vises en fejlmeddelelse.

Fejlmeddelelse	Beskrivelse	Vis
Fejl Syntaks	Hvis syntakskontrollen finder syntaksfejl, viser den en fejlmeddelelse, og prøver at placere markøren tæt på den første fejl, så du kan rette den.	
Fejl For få argumenter	Funktionen eller kommandoen mangler et eller flere argumenter	Error Too few arguments The function or command is missing one or more arguments. 
Fejl For mange argumenter	Funktionen eller kommandoen indeholder for mange argumenter og kan ikke beregnes.	Error Too many arguments The function or command contains an excessive number of arguments and cannot be evaluated. 
Fejl Ugyldig datatype	Et argument er af forkert datatype.	Error Invalid data type An argument is of the wrong data type. 

Ugyldige kommandoer i grafiktilstand

Nogle kommandoer er ikke tilladt, når programmet skifter til grafiktilstand. Hvis disse kommandoer opstår i grafiktilstand og fejl vises, og programmet afsluttes.

Ikke-tilladt kommando	Fejlmeddelelse
Anmodning	Anmodning kan ikke udføres i grafiktilstand
RequestStr	RequestStr kan ikke udføres i grafiktilstand
tekst	Tekst kan ikke udføres i grafiktilstand

Kommandoerne, der udskriver tekst til lommeregneren - **disp** and **dispAt** - vil være understøttede kommandoer i grafikkonteksten. Teksten fra disse kommandoer vil blive sendt til lommeregnerskærmen (ikke på grafik) og vil være synlig, når programmet udgår, og systemet skifter tilbage til Lommeregner-appen

Ryd (Clear)Katalog > 
CXII**Ryd** x , y , *bredde*, *højde*

Ryd

Rydder hele skærmen, hvis der ikke er angivet nogen parametre.

Rydder hele skærmen

Hvis x , y , *bredde* og *højde* er angivet, vil rektanglet der defineres af parametrene vil blive ryddet.

Ryd 10,10,100,50

Rydder et rektangelområde med øverste venstre hjørne på (10, 10) og med bredde 100, højde 50

DrawArc
 Katalog > 
CXII

DrawArc *x, y, bredde, højde, startAngle, arcAngle*

Tegner en bue i det afgrænsende rektangel med de tilvejebragte start- og buevinkler.

x, y: øverste venstre koordinat af afgrænsende rektangel

bredde, højde: dimensioner af et afgrænset rektangel

"Buevinklen" definerer buens størrelse.

Disse parametre kan gives som udtryk, der evalueres til et tal, som derefter afrundes til nærmeste heltal.

DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180



Se også: [FillArc](#)

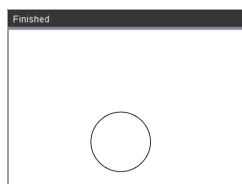
DrawCircle
 Katalog > 
CXII

DrawCircle *x, y, radius*

x, y: koordinat af centrum

radius: cirkelns radius

DrawCircle 150,150,40



Se også: [FillCircle](#)

DrawLine

Katalog > 
CXII

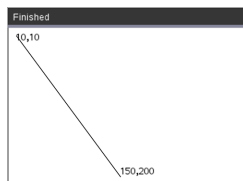
DrawLine $x1, y1, x2, y2$

Tegn en linje fra $x1, y1, x2, y2$.

Udtryk der evalueres til et tal, som derefter afrundes til nærmeste heltal.

Skærmgrænser: Hvis de angivne koordinater forårsager, at en del af linjen tegnes uden for grafiskskærmen, bliver den del af linjen klippet, og der vises ingen fejlmeddelelse.

DrawLine 10,10,150,200



DrawPoly

Katalog > 
CXII

Kommandoerne har to varianter:

DrawPoly $xlist, ylist$

or

DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$

Bemærk: DrawPoly $xlist, ylist$

Form vil forbinde $x1, y1$ til $x2, y2, x2, y2$ til $x3, y3$ og så videre.

Bemærk: DrawPoly $x1, y1, x2, y2, x3,$

$y3...xn, yn$

xn, yn vil **IKKE** blive automatisk forbundet med $x1, y1$.

Udtryk der evalueres til en liste over reelle floats

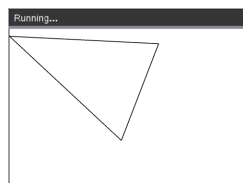
$xlist, ylist$

Udtryk der evalueres til en enkelt reel float
 $x1, y1...xn, yn$ = koordinater for polygonens hjørner

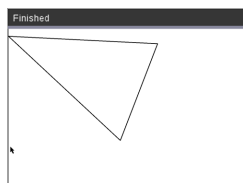
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

DrawPoly $xlist,ylist$



DrawPoly 0,10,200,20,150,150,0,10



Bemærk: DrawPoly: Indtast størrelsesdimensioner (bredde/højde) i forhold til tegnede linjer. Linjerne er tegnet i en afgrænsningskasse omkring det angivne koordinat og dimensioner således, at den faktiske størrelse af den tegnede polygon vil være større end bredden og højden.

Se også: [FillPoly](#)

DrawRect

DrawRect *x, y, bredde, højde*

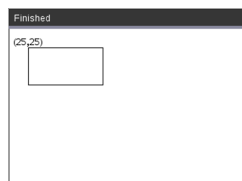
x, y: øverste venstre koordinat af rektangel

bredde, højde: bredde og højde af rektangel (rektangel trukket ned og til højre fra startkoordinat).

Bemærk: Linjerne er tegnet i en afgrænsningskasse omkring den angivne koordinat og dimensioner således, at den faktiske størrelse af det tegnede rektangel vil være større end bredden og højden der er indikeret.

Se også: [FillRect](#)

DrawRect 25,25,100,50



DrawText

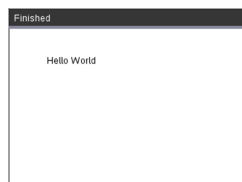
DrawText *x, y, exprOrString1*
[,exprOrString2]...

x, y: koordinat af tekstoutput

Tegner teksten i *exprOrString* ved den specificerede *x, y* koordinatplacering.

Reglerne for *exprOrString* er de samme som for **Disp** – **DrawText** kan tage flere argumenter.

DrawText 50,50,"Hello World"



FillArc

 Katalog > 
 CXII

FillArc *x, y, bredde, højde startAngle, arcAngle*

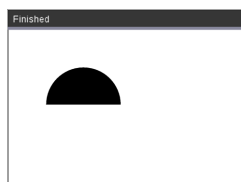
x, y: øverste venstre koordinat af afgrænsende rektangel

Tegn og fyld en bue i det afgrænsende rektangel med de tilvejebragte start- og buevinkler.

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen

"Buevinklen" definerer buens svej.

FillArc 50,50,100,100,0,180



FillCircle

 Katalog > 
 CXII

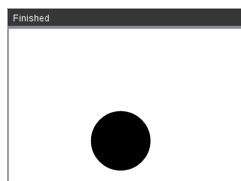
FillCircle *x, y, radius*

x, y: koordinat af centrum

Tegn og fyld en cirkel på det specificerede center med den angivne radius.

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen.

FillCircle 150,150,40



Her !

FillPoly

 Katalog > 
 CXII

FillPoly *xlist, ylist*

or

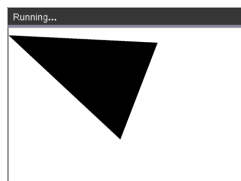
FillPoly *x1, y1, x2, y2, x3, y3...xn, yn*

Bemærk: Linjen og farven er angivet af [SetColor](#) og [SetPen](#)

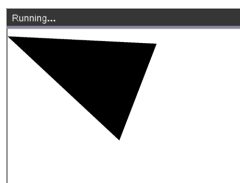
xlist:={0,200,150,0}

ylist:={10,20,150,10}

FillPoly xlist,ylist



FillPoly 0,10,200,20,150,150,0,10



FillRect

FillRect *x, y, bredde, højde*

x, y: øverste venstre koordinat af rektangel

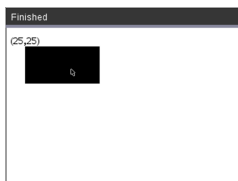
bredde, højde: bredde og højde af rektangel

Tegn og udfyld et rektangel med øverste venstre hjørne ved koordinaten angivet af (*x,y*)

Standardfyldfarve er sort. Fyldfarven kan indstilles af [SetColor](#)-kommandoen

Bemærk: Linjen og farven er angivet af [SetColor](#) og [SetPen](#)

FillRect 25,25,100,50



getPlatform()Katalog > 
CXII**getPlatform()**

getPlatform()

"dt"

Returnerer:

"dt" på desktop-softwareapplikationer

"hh" på TI-Nspire™ CX håndholdte

"ios" på TI-Nspire™ CX iPad® app

PaintBufferKatalog > 
CXII**PaintBuffer**

Grafisk buffer til skærmen

Denne kommando bruges sammen med UseBuffer for at øge visningshastighed på skærmen, når programmet genererer flere grafiske objekter.

UseBuffer

For n,1,10

x:=randInt(0,300)

y:=randInt(0,200)

radius:=randInt(10,50)

Wait 0,5

DrawCircle x,y,radius

EndFor

PaintBuffer

Dette program viser alle de 10
cirkler på én gang.

Hvis kommandoen "UseBuffer" fjernes,
vises hver cirkel, som den er
tegnet.

Se også: [UseBuffer](#)

PlotXY $x, y, shape$ x, y : koordinat til at tegne form $shape$: et tal mellem 1 og 13, der angiver formen

1 - Udfyldt cirkel

2 - Tom cirkel

3 - Udfyldt firkant

4 - Tom firkant

5 - Kryds

6 - Plus

7 - Tynd

8 - medium punkt, udfyldt

9 - medium punkt, tomt

10 - større punkt, udfyldt

11 - større punkt, tomt

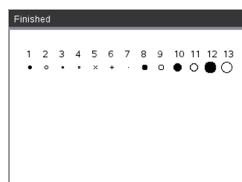
12 - største punkt, udfyldt

13 - største punkt, tomt

PlotXY 100,100,1

For $n, 1, 13$ DrawText $1+22*n, 40, n$ PlotXY $5+22*n, 50, n$

EndFor



SetColor
Katalog > 
CXII
SetColor

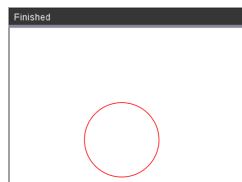
Rød-værdi, Grøn-værdi, Blå-værdi

Værdierne for Rød, Grøn, Blå skal være mellem 0 og 255

Indstiller farven til efterfølgende tegningskommandoer

SetColor 255,0,0

DrawCircle 150,150,100

**SetPen**
Katalog > 
CXII
SetPen

tykkelse, stil

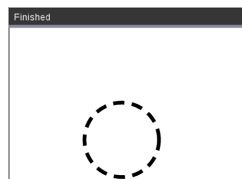
tykkelse: 1 <= tykkelse <= 3 | 1 er tyndest, 3 er tykkest

stil: 1 = Glat, 2 = Punkteret, 3 = Stiplet

Indstiller pennefaconen til efterfølgende tegningskommandoer

SetPen 3,3

DrawCircle 150,150,50

**SetWindow**
Katalog > 
CXII
SetWindow

xMin, xMax, yMin, yMax

Etablerer et logisk vindue, der er knyttet til det grafiske tegneområde. Alle parametre er påkrævede.

Hvis delen af det tegnede objekt er uden for vinduet, vil resultatet blive klippet (ikke vist), og der vises ingen fejlmeddelelse.

SetWindow 0,160,0,120

vil indstille outputvinduet til 0,0 i nederste venstre hjørne med en bredde på 160 og en højde på 120

DrawLine 0,0,100,100

SetWindow 0,160,0,120

SetPen 3,3

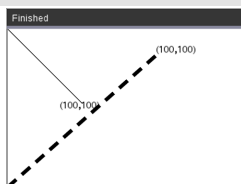
DrawLine 0,0,100,100

Hvis $x_{\min}$ er større end eller lig med $x_{\max}$ eller $y_{\min}$ er større end eller lig med $y_{\max}$, vises en fejlmeddelelse.

Eventuelle objekter tegnet før en SetWindow-kommando vil ikke blive tegnet igen i den nye konfiguration.

For at nulstille vindueparametrene til standardværdien skal du bruge:

SetWindow 0,0,0,0



UseBuffer

 Katalog > 
 CXII

UseBuffer

Tegn til grafikbuffer i stedet for skærm (for at øge ydeevnen)

Denne kommando bruges sammen med PaintBuffer for at øge visningshastighed på skærmen, når programmet genererer flere grafiske objekter.

Med UseBuffer vises al grafikken først, når den næste PaintBuffer-kommando er udført.

UseBuffer skal kun kaldes en gang i programmet, dvs. hver brug af PaintBuffer behøver ikke en tilsvarende UseBuffer

Se også: [PaintBuffer](#)

UseBuffer

```

For n,1,10
  x:=randInt(0,300)
  y:=randInt(0,200)
  radius:=randInt(10,50)
  Wait 0,5
  DrawCircle x,y,radius
EndFor
PaintBuffer

```

Dette program viser alle de 10 cirkler på én gang.

Hvis kommandoen "UseBuffer" fjernes, vises hver cirkel, som den er tegnet.

Tomme (ugyldige) elementer

Ved analyse af data fra virkeligheden har du muligvis ikke et komplet datasæt. TI-Nspire™ tillader tomme eller ugyldige dataelementer, så du kan fortsætte med det næsten komplette sæt data i stedet for at skulle starte forfra eller kassere de ukomplette forekomster.

Du kan finde et eksempel på data, der medtager tomme elementer i kapitlet Lister og regneark under *“Tegning af regnearkdata”*.

Med funktionen **delVoid()** kan du fjerne tomme elementer fra en liste. Med funktionen **isVoid()** kan du teste for et tomt element. Nærmere beskrivelse findes under **delVoid()**, side 37, og **isVoid()**, side 74.

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 **ctrl** **⏏**.

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

sum({2,_,3,5,6,6})	16.6
median({1,2,_,_,3})	2
cumulativeSum({1,2,_,4,5})	{1,3,_,7,12}
cumulativeSum($\begin{bmatrix} 1 & 2 \\ 3 & - \\ 5 & 6 \end{bmatrix}$)	$\begin{bmatrix} 1 & 2 \\ 4 & - \\ 9 & 8 \end{bmatrix}$

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>
$\Sigma()$ (Sumskabelon)	<code>sumSeq (...)</code>
$\Pi()$ (Produktskabelon)	<code>prodSeq (...)</code>
$\sin^{-1}()$, $\cos^{-1}()$, ...	<code>arcsin (...)</code> , <code>arccos (...)</code> , ...
$\Delta\text{List}()$	<code>deltaList (...)</code>

På computerens tastatur

Indsæt dette:	- ved at skrive denne genvej:
i (imaginær konstant)	<code>@i</code>
e (naturlig logbase e)	<code>@e</code>
E (videnskabelig notation)	<code>@E</code>
T (transponere)	<code>@t</code>
r (radianer)	<code>@r</code>
$^\circ$ (grader)	<code>@d</code>

Indsæt dette:	- ved at skrive denne genvej:
g (nygrader)	@ g
$\angle$ (vinkel)	@<
► (konvertering)	@>
► Decimal , ► approxFraction() , osv.	@> Decimal , @> approxFraction() , osv.

Hierarkiet i EOS™ ligningsoperativsystemet (Equation Operating System)

I dette afsnit beskrives EOS™ (Equation Operating System) der anvendes af TI-Nspire™ læringsteknologien til matematik og naturfag. Tal, variable og funktioner indtastes i en nem og enkel rækkefølge. EOS™ softwaren beregner udtryk og ligninger ved hjælp af parenteser og efter de prioriteringer, der er beskrevet nedenfor.

Operatorhierarkiet

Niveau	Operator
1	Parenteser (), kantparenteser [], krølparenteser { }
2	Henviſning (nr.)
3	Funktionskald
4	Postoperatorer: grader-minutter-sekunder ([°] , ', "), fakultet (!), procentsats (%), radian (^r), lavtstillet ([]), skifte side for lighedstegnet (^T)
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™ 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.

Eksponentiering

Eksponentiering (^) og elementvis eksponentiering (.^) 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 eksponentiering udføres før negation. For eksempel er resultatet af $-x^2$ et negativt tal, og $-9^2 = -81$. Med parenteser kvadreres et negativt tal som $(-9)^2$, så det giver 81.

Betingelse ("|")

Det argument, der følger efter ("|")-betingelses-operatoren, angiver et sæt betingelser, der skal tages hensyn til under beregningen af det argument, der kommer før operatoren.

TI-Nspire CX II - TI-Grundlæggende programmeringsfunktioner

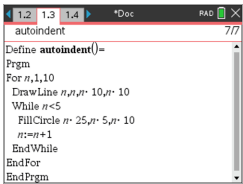
Automatisk indrykning i programmeringseditoren

TI-Nspire™-programeditor indrykker nu automatisk sætninger inde i en blokkommando.

Blokkommandoer er If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

Editoren forbereder automatisk mellemrum for programkommandoer inde i en blokkommando. Lukkekommandoen for blokken vil blive justeret med åbningskommandoen.

Eksemplet herunder viser automatisk indrykning i indlejrede blokkommandoer.



Kodefragmenter, der kopieres og indsættes, beholder den originale indrykning.

Åbning af et program, der er oprettet i en tidligere version af softwaren, bevarer den oprindelige indrykning.

Forbedrede fejlmeddelelser til TI-Basic

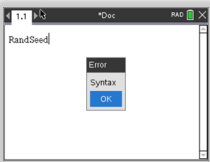
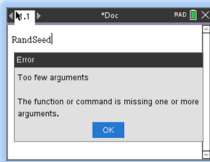
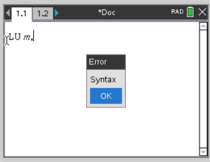
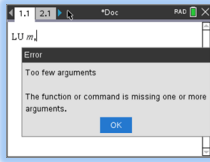
Fejlmeddelelser

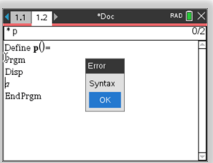
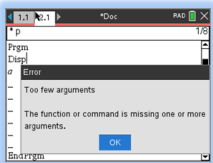
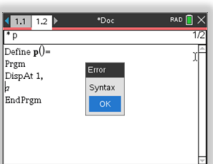
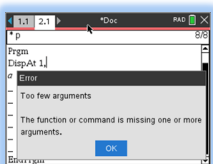
Fejltilstand	Ny meddelelse
Fejl i betingelseserklæring (If/While)	En betinget erklæring løstes ikke til TRUE eller FALSE BEMÆRK: Med ændringen for at placere markøren på linjen med fejlen, behøver vi ikke længere at angive, om fejlen er i en "If"-erklæring eller en "While"-erklæring.
Manglende EndIf	Forventede EndIf , men fandt en anden sluterklæring
Manglende EndFor	Forventede EndFor , men fandt en anden sluterklæring
Manglende EndWhile	Forventede EndWhile , men fandt en anden sluterklæring
Manglende EndLoop	Forventede EndLoop , men fandt en anden

Fejltilstand	Ny meddelelse
	sluterklæring
Manglende EndTry	Forventede EndTry , men fandt en anden sluterklæring
“ Then ” udeladt efter If <condition>	Manglende If..Then
“ Then ” udeladt efter Elseif <condition>	Then mangler i blok: Elseif .
Når “ Then ”, “ Else ” og “ Elseif ” opstod uden for kontrolblokke	Else ugyldig uden for blokke: If..Then..EndIf eller Try..EndTry
“ Elseif ” vises uden for “ If..Then..EndIf ”-blokke	Elseif ugyldig uden for blok: If..Then..EndIf
“ Then ” vises uden for “ If....EndIf ”-blokke	Then ugyldig uden for blok: If..EndIf

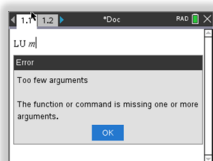
Syntaksfejl

Hvis kommandoer, der forventer en eller flere argumenter, kaldes med en ufuldstændig liste over argumenter, vil en “**Too few argument-fejl**” blive udstedt i stedet for en “**syntax**”-fejl

Aktuel adfærd	Ny CX II-adfærd
 The screenshot shows the TI-84 Plus CE calculator interface. The cursor is on the line 'RandSeed'. An error dialog box is displayed with the text 'Error Syntax' and an 'OK' button.	 The screenshot shows the TI-84 Plus CE calculator interface. The cursor is on the line 'RandSeed'. An error dialog box is displayed with the text 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.
 The screenshot shows the TI-84 Plus CE calculator interface. The cursor is on the line 'L1 U m'. An error dialog box is displayed with the text 'Error Syntax' and an 'OK' button.	 The screenshot shows the TI-84 Plus CE calculator interface. The cursor is on the line 'L1 U m'. An error dialog box is displayed with the text 'Error Too few arguments' and 'The function or command is missing one or more arguments.' and an 'OK' button.

Aktuel adfærd	Ny CX II-adfærd
	
	

Bemærk: Når en ufuldstændig liste over argumenter ikke følges af et komma, er fejlmeddelelsen: "too few arguments". Dette er det samme som tidligere udgivelser.



Konstanter og værdier

Det følgende skema oplister konstanter og deres værdier, som er tilgængelige, når der udføres enhedskonvertering. De kan indtastes manuelt eller vælges via **Konstant** listen i **Værktøjer > Enhedskonverteringer** (Håndholdt: Tryk  3).

Konstant	Navn	Værdi
_c	Lysets hastighed	299792458 _m/_s
_Cc	Coulombs konstant	8987551792.261 _m/_F
_Fc	Faradays konstant	96485.33212 _coul/_mol
_g	Tyngdeaccelerationen	9.80665 _m/_s ²
_Gc	Tyngdekonstant	6.6743E-11 _m ³ /_kg/_s ²
_h	Plancks konstant	6.62607015E-34 _J _s
_k	Boltzmanns konstant	1.380649E-23 _J/_°K
_μ0	Vakuumpermeabiliteten	1.25663706212E-6 _N/_A ²
_μb	Bohr-magneton	9.274009994E-24 _J _m ² /_Wb
_Me	Elektronens hvilemasse	9.1093837015E-31 _kg
_Mμ	Myonens masse	1.883531627E-28 _kg
_Mn	Neutronens hvilemasse	1.67492749804E-27 _kg
_Mp	Protonens hvilemasse	1.67262192369E-27 _kg
_Na	Avogadros konstant	6.02214076E23 /_mol
_q	Elementarladningen	1.602176634E-19 _coul
_Rb	Bohr-radius	5.29177210903E-11 _m
_Rc	Gaskonstanten	8.314462618 _J/_mol/_°K
_Rdb	Rydbergs konstant	10973731.568160/_m
_Re	Elektronradius	2.8179403262E-15 _m
_u	Atommasse	1.6605390666E-27 _kg
_Vm	Det molare volumen	2.241396954E-2 _m ³ /_mol
_ε0	Permittivitet for et vakuum (konstant)	8.8541878128E-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 variablen `errCode`. Brugerdefinerede programmer og funktioner kan undersøge `errCode` og bestemme årsagen til en fejl. Du finder et eksempel på anvendelse af `errCode` i Eksempel 2 under kommandoen **Try**, side 157.

Bemærk: Visse fejtilstande 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

Fejlkode	Beskrivelse
	Nedre grænse skal være mindre end øvre grænse for at definere søgeintervallet.
180	Afbrudt Der er blevet trykket på  eller  -tasten under en længere beregning eller under en programeksekvierung.
190	Cirkulær definition Denne meddelelse vises for at undgå at løbe tør for hukommelse under uendelig erstatning af variabelværdier under reduktion. For eksempel vil $a+1 \rightarrow a$, hvor a er en 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 matricerne indeholder et andet antal elementer.
250	Division med 0
260	Domænefejl Et argument skal være i et angivet domæne. For eksempel er rand(0) ikke gyldig.
270	Variabelnavn findes allerede
280	Else og Elseif er ikke gyldige uden for blokken If..EndIf
290	EndTry mangler den tilsvarende Else-erklæring
295	For stor iteration
300	Forventede en liste med 2 eller 3 elementer eller en matrix

Fejlkode	Beskrivelse
310	Første argument af nSolve skal være en ligning i en enkelt variabel. Det kan ikke indeholde en variabel uden en værdi bortset fra den variabel, der skal undersøges.
320	Første argument for solve eller cSolve skal være en ligning eller en ulighed For eksempel er solve($3x^4 - 4$, x) ugyldig, fordi det første argument ikke er en ligning.
345	Inkonsistente enheder
350	Indeks uden for området
360	Henvisningsstrengen er ikke et gyldigt variabelnavn
380	Udefineret Ans Enten gav den foregående beregning ikke en Ans, eller der blev ikke indtastet en tidligere beregning.
390	Ugyldig tildeling
400	Ugyldig tildelingsværdi
410	Ugyldig kommando
430	Ugyldigt for de aktuelle indstillinger
435	Ugyldigt gæt
440	Ugyldig implicit multiplikation for eksempel er $x(x+1)$ ugyldig, mens $x*(x+1)$ er den korrekte syntaks. Dette er for at undgå forveksling mellem implicit multiplikation og funktionskald.
450	Ugyldig i en funktion eller i det aktuelle udtryk Kun visse kommandoer er gyldige i en brugerdefineret funktion.
490	Ugyldig i Try..EndTryin Try..EndTry-blok
510	Ugyldig liste eller matrix
550	Ugyldig uden for en funktion eller et program Et antal kommandoer er ugyldige uden for en funktion eller et program. For eksempel kan Local ikke bruges, medmindre det er i en funktion eller program.
560	Ugyldig uden for blokkene Loop..EndLoop, For..EndFor eller While..EndWhile For eksempel er Exit-kommandoen kun gyldig inde i disse to løkkeblokke.
565	Ugyldig uden for et program
570	Ugyldigt stinavn For eksempel er \var ugyldigt.

Fejlkode	Beskrivelse
575	Ugyldigt pol-kompleks
580	Ugyldig programreference Programmer kan ikke kaldes i funktioner eller udtryk som $1+p(x)$, hvor p er et program.
600	Ugyldig tabel
605	Ugyldig brug af enheder
610	Ugyldigt variabelnavn i en Local-sætning
620	Ugyldigt variabel- eller funktionsnavn
630	Ugyldig variabelhenvisning
640	Ugyldig vektorsyntaks
650	Link-overførsel En overførsel mellem to enheder blev ikke gennemført. Kontroller, at forbindelseskablet er godt tilsluttet i begge ender.
665	Matricen kan ikke diagonaliseres
670	Lav hukommelse 1. Slet nogle data i dette dokument 2. Gem og luk dette dokument Hvis 1 og 2 ikke lykkes, skal batterierne tages ud og sættes på plads igen
672	Resourcer opbrugt
673	Resourcer opbrugt
680	Mangler (
690	Mangler)
700	Mangler “
710	Mangler]
720	Mangler }
730	Mangler starten eller slutningen af bloksyntaksen
740	Mangler Then i blokken If..Endif
750	Navnet er ikke en funktion eller et program
765	Ingen funktion er valgt

Fejlkode	Beskrivelse
780	Ingen løsning blev fundet
800	Ikke-reelt resultat Hvis softwaren for eksempel er i indstillingen Reel, er $\sqrt{-1}$ ugyldig. For at kunne tillade komplekse resultater skal du ændre tilstanden Reel eller Kompleks til REKTANGULÆR eller POLÆR
830	Overløb
850	Programmet blev ikke fundet Et programkald inde i et andet program blev ikke fundet i den angivne sti under eksekvering.
855	Rand funktioner er ikke tilladt graftegning
860	For mange rekursioner
870	Reserveret navn eller systemvariabel
900	Argumentfejl Median-median-modellen kunne ikke anvendes på datasættet.
910	Syntaksfejl
920	Teksten blev ikke fundet
930	For få argumenter Funktionen eller kommandoen mangler et eller flere argumenter.
940	For mange argumenter Udtrykket eller ligningen indeholder for mange argumenter og kan ikke beregnes.
950	For mange lavtstillede tegn
955	For mange udefinerede variable
960	Variablen er ikke defineret Der er ikke tildelt en værdi til variablen. Anvend en af følgende kommandoer: • <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

Fejlkode	Beskrivelse
980	Variablen er beskyttet
990	Ugyldigt variabelnavn Kontroller, at navnet ikke overskrider begrænsningerne for længden
1000	Talområde for Vindue variabel
1010	Zoom
1020	Intern fejl
1030	Overskridelse af beskyttet hukommelse
1040	Funktionen understøttes ikke. Denne funktion kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1045	Operatoren understøttes ikke. Denne operator kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1050	Funktionen understøttes ikke. Denne operator kræver Computer Algebra System. Prøv TI-Nspire™ CAS.
1060	Input-argumentet skal være et tal. Der er kun tilladt input med numeriske værdier.
1070	Trig-funktionsargumentet er for stort til en nøjagtig reduktion
1080	Ikke-understøttet anvendelse af Ans. Denne applikation understøtter ikke Ans.
1090	Funktionen er ikke defineret. Anvend en af følgende kommandoer: • 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 Komplex til REKTANGULÆR eller POLÆR
1110	Ugyldige grænser
1120	Intet fortegnsskift
1130	Argument kan ikke være en liste eller matrix
1140	Argumentfejl Det første argument skal være et polynomielt udtryk i det andet argument. Hvis det andet argument udelades, vil softwaren forsøge at vælge en standardværdi.
1150	Argumentfejl

Fejlkode	Beskrivelse
	De første to argumenter skal være polynomielle udtryk i det tredje argument. Hvis det tredje argument udelades, vil softwaren forsøge at vælge en standardværdi.
1160	Ugyldigt biblioteksstinavn Et stinavn skal være på formen <code>xxx\yyy</code>, hvor: <code>xxx</code>-delen kan have 1 til 16 tegn. <code>yyy</code>-delen kan have 1 til 15 tegn. Se flere oplysninger i afsnittet Bibliotek i dokumentationen.
1170	Ugyldig brug af biblioteksstinavn - En værdi kan ikke tildeles et stinavn med Define, <code>:=</code> eller <code>sto</code> →. - 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:

Fejlkode	Beskrivelse
	Funktionerne tangentLine og normalLine understøtter kun reelle funktioner.
1230	Domænefejl. Trigonometriske konverteringsoperatorer understøttes ikke i vinkeltilstandene Grader eller Nygrader.
1250	Argumentfejl Brug et system af lineære ligninger . Eksempel på et system med to lineære ligninger med variablerne x og y: $3x+7y=5$ $2y-5x=-1$
1260	Argumentfejl: Det første argument til nfMin eller nfMax skal være et udtryk i en enkelt variabel. Det kan ikke indeholde en variabel uden en værdi bortset fra den variabel, der skal undersøges.
1270	Argumentfejl Differentialkvotientens orden skal være 1 eller 2.
1280	Argumentfejl Brug et polynomium i udvidet form i en variabel.
1290	Argumentfejl Brug et polynomium i en variabel.
1300	Argumentfejl Koefficienter i polynomiet skal beregnes til talværdier.
1310	Argumentfejl: En funktion kunne ikke beregnes i et eller flere af den argumenter.
1380	Domænefejl Indlejrede funktionskald er ikke tilladt i domain()-funktionen.

Fejlkode- og meddelelser

Du kan bruge funktionen **warnCodes()** til at gemme fejlkode genereret ved beregning af et udtryk. Denne tabel viser de numeriske fejlkode og de tilhørende meddelelser. Se **warnCodes()** for at få et eksempel på lagring af fejlkode, side 165.

Fejlkode	Meddelelse
10000	Operationen kan måske indføre falske løsninger. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10001	Differentiation af en ligning kan give en falsk ligning.
10002	Tvivlsom løsning Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10003	Tvivlsom præcision Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10004	Operationen kan miste løsninger. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10005	cSolve kan give flere nulpunkter.
10006	Solve kan give flere nulpunkter. Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10007	Der kan være flere løsninger. Forsøg at specificere relevante nedre og øvre grænser og/eller et gæt. Eksempler med brug af solve(): • <code>solve(Ligning, Var=Gæt) nedreGrænse<Var<øvreGrænse</code> • <code>solve(Ligning, Var) nedreGrænse<Var<øvreGrænse</code> • <code>solve(Ligning,Var=Gæt)</code> Prøv at bruge grafiske metoder til at bekræfte resultaterne, når det er relevant.
10008	Domænet af resultatet kan være mindre end domænet af inputtet.
10009	Domænet af resultatet kan være større end domænet af inputtet.
10012	Ikke-reel beregning
10013	∞^0 eller undef^0 erstattet med 1
10014	undef^0 erstattes med 1
10015	1^∞ eller 1^undef erstattet med 1
10016	1^undef erstattes med 1

Fejlkode	Meddelelse
10017	Overløb erstattes 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 parameterværdier.
10022	Angivelse af korrekte nedre og øvre grænser kan give en løsning.
10023	Skalar er blevet multipliceret med identitetsmatrixen.
10024	Resultatet er opnået ved hjælp af approksimativ aritmetik.
10025	Ækvivalensen kan ikke verificeres i tilstanden EXACT.
10026	Begrænsning ignoreres muligvis. Specificer begrænsning på formen "\" Variable MathTestSymbol Constant" eller en konjunktion af disse formler eksempelvis "x<3 and x>-12"

Generelle oplysninger

Online hjælp

education.ti.com/eguide

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 for at få oplysninger om varigheden og betingelserne for garantien, eller om produktservice.

Begrænset reklamationsret. Denne garanti påvirker ikke dine lovbestemte rettigheder.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Indeks

-		-	
-, subtrahere	174	_, enhedsbenævnelse	189
!			
!, fakultet	182	, betingelses-operator	190
"		,	
", sekundnotation	188	', minutnotation	188
#		+	
#, henvisning	186	+, adder	174
#, henvisningsoperator	213	=	
%		≠, forskellig fra	179
%, procent	178	=, lig med	178
&		>	
&, tilføj	182	>, større end	180
*		Π	
*, multiplicere	175	Π, produkt	183
.		Σ	
., punktum subtraktion	177	Σ(), sum	184
., punktum multiplikation	177	ΣInt()	184
./, punkt division	177	ΣPrn()	185
., punktum potens	177	√	
., punktum addition	177	√, kvadratrods	183
/		≤	
/, dividere	175	≤, mindre end eller lig med	180
:		≥	
:=, tildel	192	≥, større end eller lig med	181
^		►	
^-1, reciprok	190	►approxFraction()	12
^, potens	175	►Base10, vise som decimalt heltal	16
		►Base16, vise som hexadecimalt	17
		►Base2, vise som binær	15

►Cylind, vise som cylindrisk vektor ...	33	Æ	
►DD, vise som decimal vinkel	34		
►Decimal, vise resultat som decimal	34	ægte brøk, propFrac	115
►DMS, vise som		A	
grader/minutter/sekunder	41		
►Grad, konvertere til vinkelmål i		afrund, round()	131
nygrader	66	A	
►Polar, vise som polær vektor	111		
►Rad, konverter til vinkelmål i		afslut løkke, EndLoop	88
radianer	119	A	
►Rect, vis som rektangulær vektor ...	122		
►Sphere, vise som en kuglevektor ...	145	afslut, Exit	47
→		A	
→, lagring	191		
⇒		afslutte funktion, EndFunc	56
⇒ , medfører[*]	181, 210	A	
↔		amortiseringstabel, amortTbl()	6, 14
↔ , ensbetydende med[*]	182	A	
©		amortTbl(), amortiseringstabel	6, 14
© , kommentar	192	A	
°		and, Boolsk operator	7
°, grader/minutter/sekunder	188	A	
°, gradnotation	188		
0		anden afledede	
Ob, binær indikator	192	skabelon til	5
Oh, hexadecimal indikator	192	A	
1		andengradsregression, QuadReg ...	116
10^(), tiende potens	189	A	
A			
abs(), absolut værdi	6	angle(), vinkel	8
Absolut værdi		A	
skabelon til	3	Anmodning	125
adder, +	174	A	
		ANOVA, envejs variansanalyse	8

A		A	
ANOVA2way, to-vejs variansanalyse	9	arcsinh()	13
A		A	
ans, sidste resultat	11	arctan()	13
A		A	
approx(), tilnærmet	11	arctanh()	13
A		A	
approxRational()	12	arcuscosinus, $\cos^{-1}()$	26
A		A	
arccos()	12	arcussinus, $\sin^{-1}()$	141
A		A	
arccosh()	12	arcustangens, $\tan^{-1}()$	152
A		A	
arccot()	12	argumenter i TVM-funktioner	161
A		A	
arccoth()	12	augment(), udvid/sammenkæd	13
A		A	
arccsc()	12	avgRC(), gennemsnitlig ændringshastighed	14
A		B	
arccsch()	13	beregne polynomium, polyEval()	112
A		beregning, rækkefølge af	212
arcsec()	13	Bestemt integral	
A		skabelon til	5
arcsech()	13	betingelses-operator "I"	190
A		betingelses-operator, rækkefølge for en beregning	212
arcsin()	13	Bibliotek (Library)	
A		danner genveje til objekter	76
		binær	
		indikator, 0b	192
		vise, ►Base2	15
		binomCdf()	18, 72

binomPdf()	18	csch(), hyperbolsk cosecans	30
Boolske operatører		CubicReg, tredjegradsregression	31
⇒	181, 210	cumulative sum, cumulativeSum()	32
⇔	182	cumulativeSum(), cumulative sum	32
and	7	Cycle, cyklus	32
nand	98	cyklus, Cycle	32
nor	102	cylindrisk vektorvisning, ►Cylind	33
not	104		
or	108	D	
xor	167		
brøker		d(), første differentialkvotient	183
ægteBrøk	115	dage mellem datoer, dbd()	33
skabelon til	1	dbd(), dage mellem datoer	33
brugerdefinerede funktioner	34	decimal	
brugerdefinerede funktioner og programmer	36	heltal vise, ►Base10	16
		vinkelvisning, ►DD	34
C		Define	34
		Define LibPriv	36
Cdf()	50	Define LibPub	36
ceiling(), rund op	18	Define, definere	34
centralDiff()	19	definere	
char(), tegnstreng	19	offentlig funktion eller program	36
ClearAZ	21	privat funktion eller program	36
ClrErr, slet fejl	22	definere, Define	34
colAugment	22	delmatrix, subMat()	149, 151
colDim(), matrix kolonnedimension	22	deltaList()	37
colNorm(), matrix kolonnenorm	23	DelVar, slette variabel	37
conj(), komplekst konjugat	23	delVoid(), fjerner ugyldige elementer	37
constructMat(), konstruer matrix	23	det(), matrixdeterminant	37
corrMat(), korrelationsmatrix	24	diag(), matrixdiagonal	38
cos ⁻¹ , arcuscosinus	26	differentialkvotienter	
cos(), cosinus	24	første differentialkvotient, d()	183
cosh ⁻¹ (), hyperbolsk arcuscosinus	27	numerisk differentialkvotient,	
cosh(), hyperbolsk cosinus	26	nDeriv()	100-101
cosinus, cos()	24	numerisk differentialkvotient,	
cot ⁻¹ (), arcuscotangens	27	nDerivative()	99
cot(), cotangens	27	dim(), dimension	38
cotangens, cot()	27	dimension, dim()	38
coth ⁻¹ (), hyperbolsk arcuscotangens	28	Disp, vis data	39, 134
coth(), hyperbolsk cotangens	28	DispAt	39
count(), tælle elementer i en liste	28	distributionsfunktioner	
countif(), tælle elementer i en liste betinget	29	binomCdf()	18
cPolyRoots()	29	binomPdf()	18
crossP(), vektorprodukt	29	normCdf()	103
csc ⁻¹ (), invers cosecans	30	normPdf()	103
csc(), cosecans	30	poissCdf()	111
csch ⁻¹ (), invers hyperbolsk cosecans	31	poissPdf()	111
		tCdf()	154

tPdf()	156	F	
χ^2 2way()	20		
χ^2 Cdf()	20	F-test- med 2 målinger	55
χ^2 GOF()	21	factor(), opløse i faktorer	50
χ^2 Pdf()	21	fakultet, !	182
dividere, P	175	fejlkode og -meddelelser	226
division af heltal, intDiv()	70	fejlmeddelelser og fejlfinding	
dot		slette fejl, ClrErr	22
produkt, dotP()	41	videresende fejl, PassErr	110
dotP(), prikprodukt	41	Fill, matrixudfyldning	51
E		finansfunktioner, tvnFV()	159
e eksponent		finansfunktioner, tvnI()	160
skabelon til	2	finansfunktioner, tvnN()	160
e til en potens, e^()	42, 47	finansfunktioner, tvnPmt()	160
E, eksponent	186	finansfunktioner, tvnPV()	161
e^(), e til en potens	42	FiveNumSammendrag	51
eff), omregn nominal til effektiv		fjerdegradsregression, QuartReg ...	117
rente	42	floor(), runde ned	52
effektiv rente, eff()	42	flyt, shift()	138
egenværdi, eigVl()	43	For	52
egenvektor, eigVc()	43	for, For	52
eigVc(), egenvektor	43	For, for	52
eigVl(), egenværdi	43	fordelingsfunktioner	
eksponent, E	186	binomCdf()	72
eksponenter		invNorm()	72
skabelon til	1	invt()	73
eksponentiel regression, ExpReg	48	Invx ² ()	71
ellers hvis, Elseif	44	format(), formatere streng	53
ellers, Else	67	formatere streng, format()	53
Elseif, ellers hvis	44	forskellig fra, ≠	179
end		første orden	
for, EndFor	52	skabelon til	5
løkke...EndLoop	88	fortegn, sign()	139
EndWhile, slut while-kompleks	167	fpart(), funktionsdel	53
enhedsvektor, unitV()	163	freqTable()	54
ensbetydende med, ⇔	182	frequency()	55
EOS ligningsoperativsystem		Frobenius-norm, norm()	103
(Equation Operating		Func, funktion	56
System)	212	Func, programfunktion	56
Equation Operating System (EOS)		funktioner	
ligningsoperativsystem	212	brugerdefinerede	34
etiket, Lbl	74	del, fpart()	53
euler(), Euler function	45	programfunktion, Func	56
Exit, afslut	47	funktioner og variable	
exp(), e til en potens	47	kopiere	24
expr(), streng til udtryk	48	fyld	200-201
ExpReg, eksponentiel regression	48	G	
		g, nygrader	187

gå til, Goto	66	arcussinus, $\sinh^{-1}()$	142
gcd(), største fælles divisor	57	cosinus, cosh()	26
gennemsnitlig ændringshastighed, avgRC()	14	sinus, sinh()	142
genveje, tastatur	210	tangens, tanh()	153
geomCdf()	57	I	
geomPdf()	58	i streng, inString()	70
Get	58, 202	identitetsmatrix, identity()	67
getDenom(), hente/returnere nævner	59	identity(), identitetsmatrix	67
getKey()	59	If, hvis	67
getLangInfo(), hent/returner sproginformation	63	imag(), imaginær del	69
getLockInfo(), tester låsestatus af en variabel eller variabelgruppe	63	imaginær del, imag()	69
getModel(), hente tilstandsindstillinger	63	indstille tilstand, setMode()	136
getNum(), hente/returnere tal	64	indstillinger, hente aktuelle	63
GetStr	64	inString(), i streng	70
getType(), get type of variable	65	int(), heltal	70
getVarInfo(), hent/returner variabeloplysninger	65	intDiv(), division af heltal	70
Goto, gå til	66	interpolate(), interpolation	71
grader/minutter/sekunder-notation	188	invers kumuleret normalfordeling (invNorm()	72
gradnotation, g	188	invers, $\wedge^{-1}$	190
grupper, låse og låse op	85, 163	invF()	71
grupper, teste låsestatus	63	invNorm(), invers kumuleret normalfordeling)	72
H		invt()	73
heltal, int()	70	$\text{Inv}\chi^2()$	71
heltalsdel, iPart()	73	iPart(), heltalsdel	73
hente/returnere nævner, getDenom()	59	irr(), intern rente intern rente, irr()	73
tal, getNum()	64	isPrime(), primtalstest	74
variabeloplysninger, getVarInfo()	63, 65	isVoid(), test for ugyldighed	74
henvisning, #	186	K	
henvisningsoperator (#)	213	kombinationer, nCr()	99
hexadecimal indikator, Oh	192	kommentar, ©	192
vise, ►Base16	17	komplekst konjugat, conj()	23
højre, right()	71, 128	Konstruer matrix, constructMat()	23
hvis, If	67	konverter ►Rad	119
hvisFn()	68	konvertere 4Grad	66
hyperbolsk arctangens, $\tanh^{-1}()$	153	kopiere variabel eller funktion, CopyVar	24
arcuscosinus, $\cosh^{-1}()$	27	korrelationsmatrix, corrMat()	24
		kuglevektor vise, ►Sphere	145

kvadratrod		tomme elementer i	208
skabelon til	1	udvid/sammenkæd, augment()	13
kvadratrod, v()	145, 183	vektorprodukt, crossP()	29
L		ln(), naturlig logaritme	82
længde af streng	38	LnReg, logaritmisk regression	83
lagre		Local, lokal variabel	84
symbol, &	191-192	Lock, låse variabel eller variabelgruppe	85
låse variable og variabelgrupper	85	Log	
Lbl, etiket	74	skabelon til	2
lcm, mindste fælles multiplum	75	logaritmer	82
left(), venstre	75	logaritmisk regression, LnReg	83
LibPriv	36	Logistik, logistisk regression	86
LibPub	36	LogistikD, logistisk regression	87
libShortcut(), danner genveje til biblioteksobjekter	76	logistisk regression, Logistik	86
lig med, =	178	logistisk regression, LogistikD	87
ligningssystem (2-ligninger)		lokal variabel, Local	84
skabelon til	3	lokal, Local	84
ligningssystem (N-ligning)		løkke, Loop	88
skabelon til	3	Loop, løkke	88
Lineær reg t-test	80	LU, matrix opløsning i øvre-nedre	89
lineær regression, LinRegAx	77	M	
lineær regression, LinRegBx	76, 78	maksimum, max()	90
LinRegBx, lineær regression	76	mat ► list(), matrix til liste	89
LinRegMx, lineær regression	77	matricer	
LinRegIntervals, lineær regression	78	cumulative sum, cumulativeSum	
linSolve()	81	()	32
list ► mat(), liste til matrix	82	delmatrix, subMat()	149, 151
liste til matrix, list ► mat()	82	determinant, det()	37
liste, tælle elementer betinget	29	diagonal, diag()	38
liste, tælle elementer i	28	dimension, dim()	38
lister		egenværdi, eigVl()	43
cumulative sum, cumulativeSum		egenvektor, eigVc()	43
()	32	identitet, identity()	67
differens, @list()	82	kolonnedimension, colDim()	22
differenser i en liste, @list()	82	kolonnenorm, colNorm()	23
liste til matrix, list ► mat()	82	liste til matrix, list ► mat()	82
maksimum, max()	90	maksimum, max()	90
matrix til liste, mat ► list()	89	matrix til liste, mat ► list()	89
midt-streng, mid()	93	minimum, min()	93
minimum, min()	93	multiplikation og addition af	
ny, newList()	100	rækker, mRowAdd()	95
prikprodukt, dotP()	41	ny, newMat()	100
produkt, product()	114	opløsning i øvre-nedre, LU	89
sortere faldende, SortD	145	produkt, product()	114
sortere stigende, SortA	144	punktum addition, .+	177
summation, sum()	150	punktum division, ./	177

punktum multiplikation, .*	177	modificeret intern forrentning, mirr()	94
punktum potens, .^	177	modulo, mod()	95
punktum subtraktion, .-	177	mRow(), matrixrækkeoperation	95
QR opløse i faktorer, QR	115	mRowAdd(), multiplikation og addition af matrixrækker	95
rækkeoperation, mRow()	95	Multipel lineær regression ttest	97
reduceret række-echelonform, rref()	132	multipliere, *	175
summation, sum()	150	MultReg	95
transponere, T	151	MultRegIntervals()	96
udfylde, Fill	51	MultRegTests()	97
udvid/sammenkæd, augment()	13		
vilkårlige, randMat()	121		
matrix (1 × 2)		N	
skabelon til	4	nand, Boolsk operator	98
matrix (2 × 1)		når, when()	166
skabelon til	4	naturalig logaritme, ln()	82
matrix (2 × 2)		nCr(), kombinationer	99
skabelon til	4	nDerivative(), numerisk differentialkvotient	99
matrix (m × n)		negation, indtaste negative tal	213
skabelon til	4	nettonutidsværdi, npv()	105
matrix til liste, mat2list()	89	newList(), ny liste	100
matrixer		newMat(), ny matrix	100
række-echelonform, ref()	123	nfMax(), numerisk funktionsmaksimum	100
rækkeaddition, rowAdd()	132	nfMin(), numerisk funktionsminimum	101
rækkedimension, rowDim()	132	nInt(), numerisk integral	101
rækenorm, rowNorm()	132	nom), omregn effektiv til nominel rente	102
rækkeskift, rowSwap()	132	nominel rente, nom()	102
max(), maksimum	90	nor, Boolsk operator	102
mean(), middel	90	norm(), Frobenius-norm	103
med,	190	normalsandsynlighedsfordeling, normCdf()	103
medfører, ⇒	181, 210	normCdf()	103
median(), median	91	normPdf()	103
median, median()	91	not, Boolsk operator	104
medium-medium linjeregression, MedMed	92	nPr(), permutationer	104
MedMed, medium-medium linjeregression	92	npv(), nettonutidsværdi	105
mens, While	167	nSolve(), numerisk løsning	105
mid(), midt-streng	93	nte rod	
middel, mean()	90	skabelon til	1
midt-streng, mid()	93	numerisk	
min(), minimum	93	differentialkvotient, nDerivative()	99
mindre end eller lig med, {	180	differentialkvotient, nDeriv()	100-101
mindste fælles multiplum, lcm	75	integral, nInt()	101
minimum, min()	93	løsning, nSolve()	105
minutnotation,	188		
mirr(), modificeret intern forrentning	94		
mod(), modulo	95		

fjerdegrads, QuartReg	117	sech(), hyperbolsk sekans	133
lineær regression, LinRegAx	77	sekundnotation, "	188
lineær regression, LinRegBx	76	sekvens, seq()	134
logistiske, Logistik	87	seq(), sekvens	134
potensregression,		seqGen()	135
PowerReg ... 112, 125, 127, 154		seqn()	136
sinus, SinReg	143	sequence, seq()	135-136
tredegrads, CubicReg, CubicReg	31	setMode(), indstil tilstand	136
Regressioner		shift(), flyt	138
lineær regression, LinRegBx	78	sign(), fortegn	139
logaritmisk, LnReg	83	simult(), sammenhørende ligninger	139
Logistisk	86	sin ⁻¹ (), arcussinus	141
medium-medium linje, MedMed	92	sin(), sinus	141
MultReg	95	sinh ⁻¹ (), hyperbolsk arcussinus	142
rektangulær vektorvisning, ►Rect	122	sinh(), hyperbolsk sinus	142
rektangulært x-koordinat, P►Rx()	109	SinReg, sinusregression	143
rektangulært y-koordinat, P►Ry()	109	sinus, sin()	141
remain(), rest	125	sinusregression, SinReg	143
RequestStr	127	skabeloner	
rest, remain()	125	Absolut værdi	3
resultat (sidste), ans	11	anden afledede	5
resultater, statistiske	146	bestemt integral	5
resultatværdier, statistik	147	brøk	1
Return, returner	128	e eksponent	2
returner, Return	128	eksponent	1
right, right()	45, 165	første orden	5
rk23(), Runge Kutta-funktion	128	kvadratrods	1
rotate(), roter	130	ligningssystem (2-ligninger)	3
roter, rotate()	130	ligningssystem (N-ligning)	3
round(), afrund	131	Log	2
rowAdd(), addition af matrixrækker	132	matrix (1 × 2)	4
rowDim(), matrixrækkedimension	132	matrix (2 × 1)	4
rowNorm(), matrixrækkenorm	132	matrix (2 × 2)	4
rowSwap(), matrixrækkeskift	132	matrix (m × n)	4
rref(), reduceret række-		nte rod	1
echelonform	132	produkt $\prod$ ()	5
rund op, ceiling()	18	stykkevis-skabelon (N stykker)	2
runde ned, floor()	52	stykkevis funktion (2 stykker)	2
runde op, ceiling()	19, 29	sum Σ ()	5
Ryd	196	slet	
		fejl, ClrErr	22
S		slette	
sammenhørende ligninger, simult()	139	ugyldige elementer fra liste	37
sandsynlighedsdensitet, normPdf		variabel, DelVar	37
()	103	slut	
sec ⁻¹ (), invers sekans	133	funktion, EndFunc	56
sec(), sekans	133	hvis, EndIf	67
sech ⁻¹ (), invers hyperbolsk sekans	134	mens, EndWhile	167
		slut på hvis, EndIf	67

slut while-kompleks, EndWhile	167	streng til udtryk, expr()	48
som af rentebetalinger	184	tegnkode, ord()	109
SortA, sorter stigende	144	tegnstreng, char()	19
SortD, sortere faldende	145	tilføj, &	182
sortere		udtryk til streng, string()	149
faldende, SortD	145	venstre, left()	75
stigende, SortA	144	string(), udtryk til streng	149
sprog		strings	
hent sproginformation	63	right, right()	45, 165
sqrt(), kvadratrodd	145	student-t sandsynlighedsfordeling,	
standardafvigelse, stdDev()	147-148, 164	tCdf()	154
stat.results	146	student-t sandsynlighedstæthed,	
stat.values	147	tPdf()	156
statistik		stykkevis-skabelon (N stykker)	
fakultet, !	182	skabelon til	2
kombinationer, nCr()	99	stykkevis funktion (2 stykker)	
median, median()	91	skabelon til	2
middel, mean()	90	subMat(), delmatrix	149, 151
permutationer, nPr()	104	substitution med " " -operator	190
standardafvigelse, stdDev()	147-148, 164	subtrahere, -	174
statistik med en variabel,		sum $\sum$ ()	
OneVar	106	skabelon til	5
to-variabel-resultater, TwoVar	161	sum af hovedstolsbetalinger	185
varians, variance()	164	sum(), summation	150
vilkårlig norm, randNorm()	121	sum, S()	184
vilkårligt tal-seed, RandSeed	122	sumlf()	150
statistik med en variabel, OneVar	106	summation, sum()	150
stdDevPop(), population		sumSeq()	151
standardafvigelse	147		
stdDevSamp(), standardafvigelse for		T	
måling	148	T (transponere)	151
Stop-kommando	149	t test, tTest	158
større end eller lig med, 	181	tælle dage mellem datoer, dbd()	33
større end, >	180	tælle elementer i en liste betinget,	
største fælles divisor, gcd()	57	countif()	29
streng		tælle elementer i en liste, count()	28
dimension, dim()	38	tan ⁻¹ (), arcustangens	152
længde	38	tan(), tangens	152
strenge		tangens, tan()	152
anvende til at skabe		tanh ⁻¹ (), hyperbolsk arcustangens	153
variabelnavne	213	tanh(), hyperbolsk tangens	153
flyt, shift()	138	tastaturgenveje	210
formatere	53	tCdf(), student t	
formatere, format()	53	sandsynlighedsfordeling	154
henvisning, #	186	tegn	197-199
højre, right()	71, 128	numerisk kode, ord()	109
i, inString	70	streng, char()	19
midt-streng, mid()	93	tegnstreng, char()	19
roter, rotate()	130		

test for ugyldighed, isVoid()	74	unitV(), enhedsvektor	163
Test_2S, F test med 2 målinger	55	unLock, oplåse variabel eller variabel gruppe	163
Text-kommando	154		
tidsdiskonterede pengeværdier, aktuel værdi	161	V	
tidsdiskonterede pengeværdier, antal betalinger	160	variabel	
tidsdiskonterede pengeværdier, betalingsbeløb	160	oprette navn fra en tegnstreng	213
tidsdiskonterede pengeværdier, Fremtidig værdi	159	slette alle enkeltbogstaver	21
tidsdiskonterede pengeværdier, Rente	160	variable	
tiende potens, 10^()	189	lokal, Local	84
tilføj, &	182	slette, DelVar	37
tilnærmet, approx()	11	variable og funktioner	
tilstande		kopiere	24
indstille, setMode()	136	variable, låse og låse op	63, 85, 163
tilstandsindstillinger, getMode()	63	varians, variance()	164
tInterval, t konfidensinterval	155	varPop()	164
tInterval_2Samp, tomålingers t konfidensinterval	155	varSamp(), målingsvarians	164
to-variabel-resultater, TwoVar	161	vektorer	
tomme (ugyldige) elementer	208	cylindrisk vektorvisning, ►Cylind	33
tPdf(), studentt		enhed, unitV()	163
sandsynlighedstæthed	156	prikprodukt, dotP()	41
transponere, T	151	vektorprodukt, crossP()	29
tredjegradsregression, CubicReg	31	vektorprodukt, crossP()	29
Try, fejlhåndteringskommando	157	venstre, left()	75
tTest, t test	158	Vent på kommando	165
tTest_2Samp, to-målingers t test	159	videresende fejl, PassErr	110
TVM-argumenter	161	vilkårlig	
tvmFV()	159	matrix, randMat()	121
tvmI()	160	vilkårlig norm, randNorm()	121
tvmN()	160	vilkårlig stikprøve	121
tvmPmt()	160	vilkårligt	
tvmPV()	161	tal-seed, RandSeed	122
TwoVar, to-variabel-resultater	161	vinkel, angle()	8
		vis data, Disp	39, 134
U		vis som	
udelukkelse med " " -operator	190	rektangulær vektor, ►Rect	122
udtryk		vise som	
streng til udtryk, expr()	48	binær, ►Base2	15
udvid/sammenkæd, augment()	13	cylindrisk vektor, ►Cylind	33
ugyldige elementer	208	decimal vinkel, ►DD	34
ugyldige elementer, fjerne	37	decimalt heltal, ►Base10	16
ugyldighed, test for	74	grader/minutter/sekunder, ►DMS	41
understregning, _	189	hexadecimal, ►Base16	17
		kuglevektor, ►Sphere	145
		polær vektor, ►Polar	111
		Viser grader/minutter/sekunder, ►DMS	41

W

warnCodes(), Warning codes	165
when(), når	166
While, mens	167

X

χ^2 , kvadrat	176
XNOR	182
xor, Boolsk exclusive or	167

Z

zInterval, z konfidensinterval	168
zInterval_1Prop, en-proportionalt z konfidensinterval	169
zInterval_2Prop, to-proportionalt z konfidensinterval	169
zInterval_2Samp, z konfidensinterval med to-målinger	170
zTest	171
zTest_1Prop, enproportional z test ..	171
zTest_2Prop, toproportional z test ..	172
zTest_2Samp, to-prøvers z test	173

Δ

Δ list(), listedifferens	82
--	----

X

χ^2 2way	20
χ^2 Cdf()	20
χ^2 GOF	21
χ^2 Pdf()	21