



TI-Nspire™ CX CAS Referentiehandleiding

Voor meer informatie over de technologie van TI kunt u de online hulppagina raadplegen op education.ti.com/eguide.

Belangrijke informatie

Tenzij uitdrukkelijk anders vermeld in de bij een programma behorende Licentie, geeft Texas Instruments betreffende programma's of boekmateriaal geen uitdrukkelijke noch impliciete garantie, daaronder mede begrepen maar niet beperkt tot impliciete garanties met betrekking tot verkoopbaarheid en geschiktheid voor een bepaald doel, en maakt zulk materiaal uitsluitend beschikbaar op een "as-is" basis. In geen geval is Texas Instruments tegenover wie dan ook aansprakelijk voor enige speciale, indirecte, bijkomende of gevolgschade verband houdend met of voortvloeiend uit de aankoop of het gebruik van dit materiaal en, ongeacht de vorm van proces, zal de enige en uitsluitende aansprakelijkheid van Texas Instruments niet hoger zijn dan het in de bij een programma behorende licentie vermelde bedrag. Daarenboven wijst Texas Instruments elke aansprakelijkheid van de hand voor vorderingen van welke aard dan ook tegen het gebruik van dit materiaal door derden.

© 2006 - 2019 Texas Instruments Incorporated

Inhoud

Uitdrukkingstemplates	1
Alfabetische lijst	8
A	8
B	17
C	21
D	47
E	61
F	71
G	81
I	92
L	101
M	117
N	126
O	135
P	138
Q	148
R	151
S	167
T	194
U	210
V	211
W	212
X	214
Z	215
Symbolen	224
TI-Nspire™ CX II - Tekenopdrachten	251
Programmeren van grafische weergaven	251
Grafisch scherm	251
Standaardweergave en instellingen	252
Foutmeldingen op het grafische scherm	253
Ongeldige opdrachten in de grafische modus	253
C	255
D	256
F	259
G	261
P	262
S	264
U	266

Lege elementen	267
Snelkoppelingen voor het invoeren van wiskundige uitdrukkingen	269
Constanten en waarden	271
EOS (Equation Operating System)-hiërarchie	272
TI-Nspire CX II - Functies van de TI-Basic-programmering	274
Auto-inspringen in de programmeereditor	274
Verbeterde foutmeldingen voor TI-Basic	274
Foutcodes en meldingen	277
Waarschuwingcodes en berichten	285
Algemene informatie	287
Online Help	287
Neem contact op met TI Ondersteuning	287
Service- en garantie-informatie	287
Index	288

Uitdrukkingstemplates

Uitdrukkingstemplates bieden u een makkelijke manier om wiskundige uitdrukkingen in standaard wiskundige notatie in te voeren. Wanneer u een template invoegt, verschijnt deze op de invoerregel met kleine blokjes op de posities waarop u elementen kunt invoeren. Een cursor geeft aan welk element u kunt invoeren.

Gebruik de pijltoetsen of druk op **tab** om de cursor te verplaatsen naar de positie van elk element, en typ een waarde of uitdrukking voor het element in. Druk op **enter** of **ctrl enter** om de uitdrukking uit te werken.

Breukentemplate

ctrl **÷** -toetsen



Opmerking: zie ook / (delen), pag. 226.

Voorbeeld:

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

Exponent-template

^ -toetsen



Opmerking: typ de eerste waarde, typ op **^** en typ dan de exponent. Om de cursor terug te brengen naar de basisregel drukt u op de pijl naar rechts (►).

Opmerking: zie ook ^ (macht), pag. 227.

Voorbeeld:

$$2^3 = 8$$

Worteltemplate

ctrl **x²** -toetsen



Opmerking: zie ook $\sqrt{()}$ (wortel), pag. 238.

Voorbeeld:

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

N-de wortel-template

ctrl **^** -toetsen



Opmerking: zie ook wortel (), pag. 163.

Voorbeeld:



N-de wortel-template

ctrl **^** -toetsen

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

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

e-macht-template

e^x -toetsen

e

Het getal e verheven tot een macht

Opmerking: zie ook **e^()**, pag. 61.

Voorbeeld:

$$e^1 \quad e$$

$$e^{1.} \quad 2.71828182846$$

Log-template

ctrl **10^x** -toets

log

Berekent de log ten opzichte van een gespecificeerd grondtal. Voor het standaard grondtal 10 laat u het grondtal weg.

Opmerking: zie ook **log()**, pag. 113.

Voorbeeld:

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

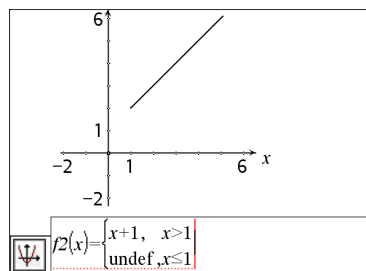
Template voor stuksgewijs gedefinieerde functies (2-stuks)

Catalogus > **10^x**

Hiermee kunt u uitdrukkingen en condities voor een in twee stukken- stuksgewijs gedefinieerde functie creëren. Om een stuk toe te voegen klikt u in de template en herhaalt u de template.

Opmerking: zie ook **piecewise()**, pag. 140.

Voorbeeld:



Template voor stuksgewijs gedefinieerde functies (N-stuks)

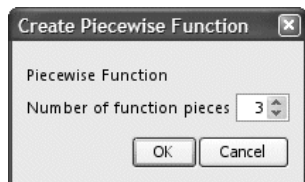
Catalogus >



Hiermee kunt u uitdrukkingen en condities voor een -stuksgewijs gedefinieerde functie in N stukken creëren. Vraagt om N .

Voorbeeld:

Zie het voorbeeld bij Template voor stuksgewijs gedefinieerde functies (2-stuks).



Opmerking: zie ook `piecewise()`, pag. 140.

Stelsel van 2 vergelijkingen-template

Catalogus >



Creëert een stelsel van twee vergelijkingen. Om een rij toe te voegen aan een bestaand stelsel, klikt u in de template en herhaalt u de template.

Opmerking: zie ook `system()`, pag. 194.

Voorbeeld:

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

Stelsel van N vergelijkingen-template

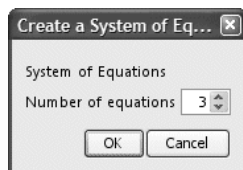
Catalogus >



Hiermee kunt u een stelsel van N vergelijkingen creëren. Vraagt om N .

Voorbeeld:

Zie het voorbeeld bij Stelsel van vergelijkingen-template (2 vergelijkingen).



Opmerking: zie ook `system()`, pag. 194.

Absolute waarde-template

Catalogus > 

 **Opmerking:** zie ook **abs()**, pag. 8.

Voorbeeld:

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

dd°mm'ss.ss" template

Catalogus > 



Voorbeeld:

Hiermee kunt u hoeken in **dd°mm'ss.ss"**-opmaak invoeren, waarbij **dd** het aantal decimale graden, **mm** het aantal minuten en **ss.ss** het aantal seconden is.

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

Matrixtemplate (2 x 2)

Catalogus > 



Voorbeeld:

Creëert een 2 x 2 matrix.

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

Matrixtemplate (1 x 2)

Catalogus > 



Voorbeeld:

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

Matrixtemplate (2 x 1)

Catalogus > 



Voorbeeld:

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

Matrixtemplate (m x n)

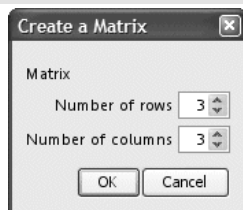
Catalogus > 

De template verschijnt nadat u het aantal rijen en kolommen heeft ingevoerd.

Voorbeeld:

Matrixtemplate (m x n)

Catalogus > 



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

Opmerking: als u een matrix creëert met een groot aantal rijen en kolommen, kan het even duren voordat deze verschijnt.

Somtemplate (Σ)

Catalogus > 

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

Voorbeeld:

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

Opmerking: zie ook $\Sigma()$ (sumSeq), pag. 239.

Product-template (Π)

Catalogus > 

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

Voorbeeld:

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

Opmerking: zie ook $\Pi()$ (prodSeq), pag. 238.

Eerste afgeleide-template

Catalogus > 

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

Voorbeeld:

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

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

De eerste afgeleide-template kan ook worden gebruikt om de eerste afgeleide in een punt te berekenen.

Opmerking: zie ook $d()$ (afgeleide), pag. 235.

Tweede afgeleide-template

Catalogus > 

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

De tweede afgeleide-template kan ook worden gebruikt om de tweede afgeleide in een punt te berekenen.

Opmerking: zie ook **d()** (afgeleide), pag. 235.

Voorbeeld:

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

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

N-de afgeleide-template

Catalogus > 

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

De *n*de afgeleide-template kan worden gebruikt om de *n*de afgeleide te berekenen.

Opmerking: zie ook **d()** (afgeleide), pag. 235.

Voorbeeld:

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

Bepaalde integraal-template

Catalogus > 

$$\int_a^b \square dx$$

Opmerking: zie ook **f()** **integral()**, pag. 224.

Voorbeeld:

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

Onbepaalde integraal-template

Catalogus > 

$$\int \square dx$$

Opmerking: zie ook **f()** **integral()**, pag. 224.

Voorbeeld:

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

Limiet-template

Catalogus > 

$$\lim_{\square \rightarrow \square} \square$$

Voorbeeld:

Gebruik – of (–) voor de linkerlimiet.
 Gebruik + voor de rechterlimiet.

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

Opmerking: zie ook `limit()`, pag. 103.

Alfabetische lijst

Elementen waarvan de namen niet alfabetische zijn (zoals +, ! en >) staan aan het eind van dit hoofdstuk, pag. 224. Tenzij anders gespecificeerd zijn alle voorbeelden in dit hoofdstuk uitgevoerd in de standaard reset-modus, en wordt van alle variabelen aangenomen dat ze onbepaald zijn.

A

abs()

Catalogus >

abs(*UitdrI*)⇒*uitdrukking*

abs(*LijstI*)⇒*lijst*

abs(*MatrixI*)⇒*matrix*

Geeft de absolute waarde van het argument.

Opmerking: zie ook **Absolute waarde-template**, pag. 4.

Als het argument een complex getal is, dan wordt de modulus van dat getal gegeven.

Opmerking: alle onbepaalde variabelen worden behandeld als reële variabelen.

$\left\lceil \frac{\pi}{2}, \frac{\pi}{3} \right\rceil$	$\left\{ \frac{\pi}{2}, \frac{\pi}{3} \right\}$
$ 2-3\cdot i $	$\sqrt{13}$
$ z $	$ z $
$ x+y\cdot i $	$\sqrt{x^2+y^2}$

amortTbl()

Catalogus >

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

Aflossingsfunctie die een matrix als aflossingstabel genereert voor een serie TVM-argumenten.

NPmt is het aantal betalingen dat in de tabel moet worden opgenomen. De tabel begint met de eerste betaling.

N, I, PV, Pmt, FV, PpY, CpY en *PmtAt* worden beschreven in de tabel met TVM-argumenten, pag. 208.

- Als u *Pmt* weglaat, dan wordt de standaardwaarde ***Pmt=tvmpmt(N,I,PV,FV,PpY,CpY,PmtAt)*** gebruikt.
- Als u *FV* weglaat, dan wordt de

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

standaardwaarde $FV=0$ gebruikt.

- De standaardwaarden voor PpY , CpY en $PmtAt$ zijn hetzelfde als voor de TVM-functies.

afgerondeWaarde specificeert het aantal decimalen voor afronding.

Standaardwaarde=2.

De kolommen in de resulterende matrix zijn in de volgorde: Aantal betalingen, bedrag betaald aan rente, bedrag betaald aan de hoofdsom (aflossing) en balans.

De balans die getoond wordt in rij n is de balans na betaling n .

U kunt de uitvoermatrix gebruiken als invoer voor de andere aflossingsfuncties $\Sigma Int()$ en $\Sigma Prn()$, pag. 239, en **bal()**, pag. 17.

and

BooleaanseUitdr1 **and**

BooleaanseUitdr2 $\Rightarrow$ *Booleaanse uitdrukking*

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

BooleaanseLijst1 **and**

BooleaanseLijst2 $\Rightarrow$ *Booleaanse lijst*

BooleaanseMatrix1 **and**

BooleaanseMatrix2 $\Rightarrow$ *Booleaanse matrix*

Geeft waar of onwaar of een vereenvoudigde vorm van de oorspronkelijke invoer.

Geheel getal1 **and** *Geheel getal2* $\Rightarrow$ *geheel getal*

In de Hex-grondtalmodus:

$$0h7AC36 \text{ and } 0h3D5F \quad 0h2C16$$

Belangrijk: nul, niet de letter O.

In de Bin-grondtalmodus:

$$0b100101 \text{ and } 0b100 \quad 0b100$$

In de Dec-grondtalmodus:

Vergelijkt twee reële gehele getallen bit-voor-bit met behulp van een **and**-bewerking. Intern worden beide gehele getallen geconverteerd naar 64-bits binaire getallen met een teken (positief of negatief). Wanneer overeenkomstige bits vergeleken worden, is het resultaat 1 als beide bits 1 zijn; anders is het resultaat 0. De geretournerde waarde geeft de bitresultaten, en wordt weergegeven volgens de ingestelde grondtal-modus.

U kunt de gehele getallen invoeren in elk grondtal. Voor een binaire of hexadecimale invoer moet u respectievelijk het prefix 0b of 0h gebruiken. Zonder prefix worden gehele getallen behandeld als decimaal (grondtal 10).

Als u een decimaal geheel getal invoert dat te groot is voor een 64-bits binaire vorm met een teken (positief of negatief), dan wordt er een symmetrische modulo-bewerking gebruikt om de waarde binnen het betreffende bereik te brengen.

37 and 0b100

4

Opmerking: een binaire invoer kan maximaal 64 cijfers hebben (het prefix 0b niet meegeteld). Een hexadecimale invoer kan maximaal 16 cijfers hebben.

angle()

angle(*Uitdr1*) $\Rightarrow$ *uitdrukking*

Geeft de hoek van het argument, waarbij het argument als een complex getal wordt geïnterpreteerd.

Opmerking: alle onbepaalde variabelen worden behandeld als reële variabelen.

In de hoekmodus Graden:

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

In de hoekmodus Decimale graden:

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

In de hoekmodus Radialen:

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

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

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

angle()Catalogus > **angle(Lijst1)**⇒lijst $\text{angle}(\{1+2\cdot i, 3+0\cdot i, 0-4\cdot i\})$ **angle(Matrix1)**⇒matrix $\left\{\frac{\pi}{2}-\tan^{-1}\left(\frac{1}{2}\right), 0, \frac{-\pi}{2}\right\}$

Geeft een lijst of matrix met de hoeken van de elementen in *Lijst1* of *Matrix1*, waarbij elk element geïnterpreteerd wordt als een complex getal dat een punt in een rechthoekig twee-dimensionaal assenstelsel voorstelt.

ANOVACatalogus > **ANOVA** *Lijst1, Lijst2[, Lijst3, ..., Lijst20][, Vlag]*

Voert een eenwegs-variantieanalyse uit voor het vergelijken van de gemiddelden van twee tot 20 populaties. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Vlag=0 voor gegevens, *Vlag=1* voor statistieken

Uitvoervariabele	Beschrijving
stat.F	Waarde van de F-statistiek
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.df	Vrijheidsgraden van de groepen
stat.SS	Som van de kwadraten van de groepen
stat.MS	Gemiddelde van de kwadraten van de groepen
stat.dfError	Vrijheidsgraden van de fouten
stat.SSError	Som van de kwadraten van de fouten
stat.MSError	Gemiddelde kwadraat van de fouten
stat.sp	Gepoolde standaarddeviatie
stat.xbarlist	Gemiddelde van de invoer van de lijsten
stat.CLowerList	95% betrouwbaarheidsintervallen voor het gemiddelde van elke invoerlijst
stat.CUpperList	95% betrouwbaarheidsintervallen voor het gemiddelde van elke invoerlijst

ANOVA2wayCatalogus > **ANOVA2way** *Lijst1, Lijst2[, Lijst3, ..., Lijst10]*

[,NivRij]

Berekent een tweewegs variantieanalyse voor het vergelijken van de gemiddelden van twee tot 10 populaties. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

NivRij=0 voor blok

NivRij=2,3,...,*Len*-1, voor tweeweg, waarbij
Len=*lengte(Lijst1)*=*lengte(Lijst2)* = ... = *lengte(Lijst10)* en *Len* / *NivRij* ∈ {2,3,...}

Uitvoer: Blokopmaak

Uitvoervariabele	Beschrijving
stat.F	F-statistiek van de kolomfactor
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.df	Vrijheidsgraden van de kolomfactor
stat.SS	Som van de kwadraten van de kolomfactor
stat.MS	Gemiddelde van de kwadraten van de kolomfactor
stat.FBlock	F-statistiek voor de factor
stat.PValBlock	Kleinste kans waarbij de nulhypothese verworpen kan worden
stat.dfBlock	Vrijheidsgraden van de factor
stat.SSBlock	Som van de kwadraten van de factor
stat.MSBlock	Gemiddelde van de kwadraten van de factor
stat.dfError	Vrijheidsgraden van de fouten
stat.SSError	Som van de kwadraten van de fouten
stat.MSError	Gemiddelde van de kwadraten van de fouten
stat.s	Standaarddeviatie van de fout

Uitvoer van KOLOMFACTOR

Uitvoervariabele	Beschrijving
stat.Fcol	F-statistiek van de kolomfactor
stat.PValCol	Kanswaarde van de kolomfactor
stat.dfCol	Vrijheidsgraden van de kolomfactor

Uitvoervariabele	Beschrijving
stat.SSCol	Som van de kwadraten van de kolomfactor
stat.MSCol	Gemiddelde van de kwadraten van de kolomfactor

Uitvoer van RIJFACTOR

Uitvoervariabele	Beschrijving
stat.FRow	F-statistiek van de rijfactor
stat.PValRow	Kanswaarde van de rijfactor
stat.dfRow	Vrijheidsgraden van de rijfactor
stat.SSRow	Som van de kwadraten van de rijfactor
stat.MSRow	Gemiddelde van de kwadraten van de rijfactor

Uitvoer van INTERACTIE

Uitvoervariabele	Beschrijving
stat.FInteract	F-statistiek van de interactie
stat.PValInteract	Kanswaarde van de interactie
stat.dfInteract	Vrijheidsgraden van de interactie
stat.SSInteract	Som van de kwadraten van de interactie
stat.MSInteract	Gemiddelde van de kwadraten van de interactie

Uitvoer van FOUT

Uitvoervariabele	Beschrijving
stat.dfError	Vrijheidsgraden van de fouten
stat.SSError	Som van de kwadraten van de fouten
stat.MSError	Gemiddelde van de kwadraten van de fouten
s	Standaarddeviatie van de fout

Ans	ctrl (-) -toetsen	
Ans⇒waarde	56	56
Geeft het resultaat van de meest recent uitgewerkte uitdrukking.	56+4	60
	60+4	64

approx()Catalogus > **approx(Uitdr I) ⇒ uitdrukking**

Geeft de uitwerking van het argument als een uitdrukking met decimale waarden, indien mogelijk, ongeacht de huidige **Automatische of Benaderende** modus.

Dit is hetzelfde als het argument invoeren en op   drukken.

approx(Lijst I) ⇒ lijst**approx(Matrix I) ⇒ matrix**

Geeft een lijst of *matrix* waarin elk element uitgewerkt is naar een decimale waarde, indien mogelijk.

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

►approxFraction()Catalogus > 

Uitdr ► **approxFraction([Tol])**
⇒ uitdrukking

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

Geeft de invoer weer als een breuk, gebruikt een tolerantie van *Tol*. Als *tol* wordt weggelaten, wordt er een tolerantie van 5.E-14 gebruikt.

Opmerking: u kunt deze functie vanaf het toetsenbord van de computer invoeren door **@>approxFraction(...)** in te typen.

$\frac{1}{2} + \frac{1}{3} + \tan(\pi)$	0.833333
0.8333333333333333 ► $\text{approxFraction}(5 \cdot 10^{-14})$	$\frac{5}{6}$
{ $\pi, 1.5$ } ► $\text{approxFraction}(5 \cdot 10^{-14})$	$\left\{ \frac{5419351}{1725033}, \frac{3}{2} \right\}$

approxRational()Catalogus > **approxRational(Uitdr[, tol])** ⇒ uitdrukking**approxRational(Lijst[, tol])** ⇒ lijst**approxRational(Matrix[, Tol])** ⇒ matrix

Geeft het argument als een breuk met een tolerantie van *tol*. Als *tol* wordt weggelaten, wordt er een tolerantie van 5.E-14 gebruikt.

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

arccos() Zie $\cos^{-1}()$, pag. 33.

arccosh() Zie $\cosh^{-1}()$, pag. 35.

arccot() Zie $\cot^{-1}()$, pag. 36.

arccoth() Zie $\coth^{-1}()$, pag. 37.

arccsc() Zie $\csc^{-1}()$, pag. 39.

arccsch() Zie $\csch^{-1}()$, pag. 40.

arcLen() Catalogus > 

arcLen(<i>Uitdr1</i>,<i>Var</i>,<i>Start</i>,<i>Eind</i>) $\Rightarrow$<i>uitdrukking</i> Geeft de booglengte van <i>Uitdr1</i> van <i>Start</i> tot <i>Eind</i> ten opzichte van variabele <i>Var</i>. Booglengte wordt berekend als een integraal, waarbij een functiemodusdefinitie wordt verondersteld. arcLen(<i>Lijst1</i>,<i>Var</i>,<i>Start</i>,<i>Eind</i>)$\Rightarrow$<i>lijst</i> Geeft een lijst van de booglengtes van elk element van <i>Lijst1</i> van <i>Start</i> tot <i>Eind</i> ten opzichte van <i>Var</i>.	<table border="0" style="width: 100%;"><tr><td style="border-bottom: 1px solid black; padding-bottom: 5px;">$\text{arcLen}(\cos(x), x, 0, \pi)$</td><td style="text-align: right; padding-right: 10px;">3.8202</td></tr><tr><td style="border-bottom: 1px solid black; padding-bottom: 5px;">$\text{arcLen}(f(x), x, a, b)$</td><td style="text-align: right; padding-right: 10px;">$\int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} dx$</td></tr><tr><td style="border-bottom: 1px solid black; padding-bottom: 5px;">$\text{arcLen}(\{\sin(x), \cos(x)\}, x, 0, \pi)$</td><td style="text-align: right; padding-right: 10px;">$\{3.8202, 3.8202\}$</td></tr></table>	$\text{arcLen}(\cos(x), x, 0, \pi)$	3.8202	$\text{arcLen}(f(x), x, a, b)$	$\int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} dx$	$\text{arcLen}(\{\sin(x), \cos(x)\}, x, 0, \pi)$	$\{3.8202, 3.8202\}$
$\text{arcLen}(\cos(x), x, 0, \pi)$	3.8202						
$\text{arcLen}(f(x), x, a, b)$	$\int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2 + 1} dx$						
$\text{arcLen}(\{\sin(x), \cos(x)\}, x, 0, \pi)$	$\{3.8202, 3.8202\}$						

arcsec() Zie $\sec^{-1}()$, pag. 167.

arcsech()

Zie $\text{sech}^{-1}()$, pag. 168.

arcsin()

Zie $\sin^{-1}()$, pag. 179.

arcsinh()

Zie $\sinh^{-1}()$, pag. 180.

arctan()

Zie $\tan^{-1}()$, pag. 195.

arctanh()

Zie $\tanh^{-1}()$, pag. 196.

augment()

Catalogus > 

augment(Lijst1, Lijst2)⇒lijst

Geeft een nieuwe lijst die bestaat uit *Lijst2* aan het eind van *Lijst1*.

augment(Matrix1, Matrix2)⇒matrix

Geeft een nieuwe matrix die bestaat uit *Matrix2* toevoegd aan *Matrix1*. Wanneer het teken “,” wordt gebruikt, moeten de matrices gelijke rijafmetingen hebben, en wordt *Matrix2* toegevoegd aan *Matrix1* als nieuwe kolommen. Augment verandert *Matrix1* en *Matrix2* niet.

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

{1,-3,2,5,4}

12

34

→ m1

12

34

5

6

→ m2

5

6

augment(m1,m2)

125

346

16 Alfabetische lijst

avgRC()	Catalogus > 	
avgRC (<i>Uitdr1</i> , <i>Var</i> [= <i>Waarde</i>] [, <i>Stap</i>]) ⇒ <i>uitdrukking</i>	$\text{avgRC}(f(x), x, h)$	$\frac{f(x+h)-f(x)}{h}$
avgRC (<i>Uitdr1</i> , <i>Var</i> [= <i>Waarde</i>] [, <i>Lijst1</i>]) ⇒ <i>lijst</i>	$\text{avgRC}(\sin(x), x, h) x=2$	$\frac{\sin(h+2)-\sin(2)}{h}$
avgRC (<i>Lijst1</i> , <i>Var</i> [= <i>Waarde</i>] [, <i>Stap</i>]) ⇒ <i>lijst</i>	$\text{avgRC}(x^2-x+2, x)$	$2 \cdot (x-0.4995)$
	$\text{avgRC}(x^2-x+2, x, 0.1)$	$2 \cdot (x-0.45)$
avgRC (<i>Matrix1</i> , <i>Var</i> [= <i>Waarde</i>] [, <i>Stap</i>]) ⇒ <i>matrix</i>	$\text{avgRC}(x^2-x+2, x, 3)$	$2 \cdot (x+1)$

Geeft het differentiequotiënt (gemiddelde veranderingssnelheid).

Uitdr1 kan een door de gebruiker gedefinieerde functienaam zijn (zie **Func**).

Wanneer *Waarde* gespecificeerd is, wordt elke eerdere variabeletoekenning of elke huidige “|”-substitutie voor de variabele onderdrukt.

Stap is de stapgrootte. Als *Stap* wordt weggelaten, is de standaardwaarde 0,001.

Merk op dat de soortgelijke functie **centralDiff()** het centraal-differentiequotiënt gebruikt.

B

bal()		Catalogus > 																													
bal (<i>NPmt</i> , <i>N</i> , <i>I</i> , <i>PV</i> , [<i>Pmt</i>], [<i>FV</i>], [<i>PpY</i>], [<i>CpY</i>], [<i>PmtAt</i>], [<i>afgerondeWaarde</i>]) ⇒ <i>waarde</i>		bal(5,6,5.75,5000,,12,12) 833.11																													
bal (<i>NPmt</i> , <i>amortTable</i>)⇒ <i>waarde</i>		<i>tbl</i> :=amortTbl(6,6,5.75,5000,,12,12)																													
Aflossingsfunctie die de geplande balans berekent na een gespecificeerde betaling. <i>N</i> , <i>I</i> , <i>PV</i> , <i>Pmt</i> , <i>FV</i> , <i>PpY</i> , <i>CpY</i> en <i>PmtAt</i> worden beschreven in de tabel met TVM-argumenten. pag. 208.		<table><tr><td>0</td><td>0.</td><td>0.</td><td>5000.</td></tr><tr><td>1</td><td>-23.35</td><td>-825.63</td><td>4174.37</td></tr><tr><td>2</td><td>-19.49</td><td>-829.49</td><td>3344.88</td></tr><tr><td>3</td><td>-15.62</td><td>-833.36</td><td>2511.52</td></tr><tr><td>4</td><td>-11.73</td><td>-837.25</td><td>1674.27</td></tr><tr><td>5</td><td>-7.82</td><td>-841.16</td><td>833.11</td></tr><tr><td>6</td><td>-3.89</td><td>-845.09</td><td>-11.98</td></tr></table>		0	0.	0.	5000.	1	-23.35	-825.63	4174.37	2	-19.49	-829.49	3344.88	3	-15.62	-833.36	2511.52	4	-11.73	-837.25	1674.27	5	-7.82	-841.16	833.11	6	-3.89	-845.09	-11.98
		0	0.	0.	5000.																										
		1	-23.35	-825.63	4174.37																										
		2	-19.49	-829.49	3344.88																										
		3	-15.62	-833.36	2511.52																										
		4	-11.73	-837.25	1674.27																										
		5	-7.82	-841.16	833.11																										
6	-3.89	-845.09	-11.98																												
bal(4, <i>tbl</i>) 1674.27																															

Aflossingsfunctie die de geplande balans berekent na een gespecificeerde betaling.

N, *I*, *PV*, *Pmt*, *FV*, *PpY*, *CpY* en *PmtAt* worden beschreven in de tabel met TVM-argumenten, pag. 208.

NPmt specificeert het nummer van de betaling waarna u de gegevens berekend wilt hebben.

N , I , PV , Pmt , FV , PpY , CpY en $PmtAt$ worden beschreven in de tabel met TVM-argumenten, pag. 208.

- Als u Pmt weglaat, dan wordt de standaardwaarde $Pmt = \mathbf{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ gebruikt.
- Als u FV weglaat, dan wordt de standaardwaarde $FV = 0$ gebruikt.
- De standaardwaarden voor PpY , CpY en $PmtAt$ zijn hetzelfde als voor de TVM-functies.

afgerondeWaarde specificeert het aantal decimalen voor afronding.
Standaardwaarde=2.

bal($NPmt, amortTable$) berekent de balans na het nummer van de betaling $NPmt$, op basis van de aflossingstabel $amortTable$. Het argument $amortTable$ moet een matrix zijn in de vorm die beschreven wordt onder **amortTbl**(), pag. 8.

Opmerking: zie ook $\Sigma Int()$ en $\Sigma Prn()$, pag. 239.

►Base2 ►Grondtal2

Geheel getal1 ►Base2 ⇒ *geheel getal*

256 ►Base2	0b100000000
------------	-------------

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @►Base2 in te typen.

0h1F ►Base2	0b11111
-------------	---------

Converteert *Geheel getal1* naar een binair getal. Binaire of hexadecimale getallen hebben altijd respectievelijk het prefix 0b of 0h. Nul, niet de letter O, gevolgd door b of h.

0b *binairGetal*

0h *hexadecimaalGetal*

Een binair getal kan maximaal 64 cijfers hebben. Een hexadecimaal getal kan maximaal 16 cijfers hebben.

Zonder prefix wordt *Geheel getal* behandeld als decimaal (grondtal 10). Het resultaat wordt binair weergegeven, ongeacht de Grondtal-modus.

Negatieve getallen worden weergegeven in de "twee-complement"-vorm. Bijvoorbeeld,

-1 wordt weergegeven als
 0hFFFFFFFFFFFFFFF in de hexadecimale
 modus 0b111...111 (64 enen) in de
 binaire modus

-2^{63} wordt weergegeven als
 0h8000000000000000 in de hexadecimale
 modus 0b100...000 (63 nullen) in de
 binaire modus

Als u een decimaal geheel getal invoert dat buiten het bereik van een 64-bits binaire vorm met een teken (positief of negatief) valt, dan wordt er een symmetrische modulo-bewerking gebruikt om de waarde binnen het betreffende bereik te brengen. In de volgende voorbeelden vallen de waarden buiten het bereik.

2^{63} wordt -2^{63} en wordt weergegeven als
 0h8000000000000000 in de hexadecimale
 modus 0b100...000 (63 nullen) in de
 binaire modus

2^{64} wordt 0 en wordt weergegeven als 0h0
 in de hexadecimale modus 0b0 in de
 binaire modus

$-2^{63} - 1$ wordt $2^{63} - 1$ en wordt
 weergegeven als 0h7FFFFFFFFFFFFFFF in
 de hexadecimale modus 0b111...111
 (64 enen) in de binaire modus

Geheel getal ►Base10⇒*geheel getal*

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @►Base10 in te typen.

0b10011►Base10	19
0h1F►Base10	31

Converteert *Geheel getal* naar een decimaal (grondtal 10) getal. Een binair of hexadecimaal getal moet altijd respectievelijk het prefix 0b of 0h hebben.

0b *binairGetal*

0h *hexadecimaalGetal*

Nul, niet de letter O, gevolgd door b of h.

Een binair getal kan maximaal 64 cijfers hebben. Een hexadecimaal getal kan maximaal 16 cijfers hebben.

Zonder prefix wordt *Geheel getal* behandeld als decimaal. Het resultaat wordt als decimaal getal weergegeven, ongeacht de Grondtal-modus.

►Base16 (►Grondtal16)

Geheel getal ►Base16⇒*geheel getal*

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>►Base16 in te typen.

Converteert *Geheel getal* naar een hexadecimaal getal. Binaire of hexadecimale getallen hebben altijd respectievelijk het prefix 0b of 0h.

0b *binairGetal*

0h *hexadecimaalGetal*

Nul, niet de letter O, gevolgd door b of h.

Een binair getal kan maximaal 64 cijfers hebben. Een hexadecimaal getal kan maximaal 16 cijfers hebben.

Zonder prefix wordt *Geheel getal* behandeld als decimaal (grondtal 10). Het resultaat wordt als hexadecimaal getal weergegeven, ongeacht de Grondtal-modus.

256►Base16	0h100
0b111100001111►Base16	0hF0F

Als u een decimaal geheel getal invoert dat te groot is voor een 64-bits binaire vorm met een teken (positief of negatief), dan wordt er een symmetrische modulo-bewerking gebruikt om de waarde binnen het betreffende bereik te brengen. Zie voor meer informatie ►Base2, pag. 18.

binomCdf()

binomCdf(n,p)⇒lijst

binomCdf($n,p,ondergrens,bovengrens$)⇒getal als *ondergrens* en *bovengrens* getallen zijn, *lijst* als *ondergrens* en *bovengrens* lijsten zijn

binomCdf($n,p,bovengrens$)voor $P(0 \leq X \leq bovengrens)$ ⇒getal als *bovengrens* een getal is, *lijst* als *bovengrens* een lijst is

Berekent de cumulatieve kans voor de discrete binomiale verdeling met aantal pogingen n en succeskans p bij iedere poging.

Voor $P(X \leq bovengrens)$ stelt u *ondergrens*=0 in

binomPdf()

binomPdf(n,p)⇒lijst

binomPdf($n,p,XWaarde$)⇒getal als *XWaarde* een getal is, *lijst* als *XWaarde* een lijst is

Berekent de kans voor de discrete binomiale verdeling met aantal pogingen n en succeskans p bij iedere poging.

C

ceiling()

ceiling(*Uitdr1*)⇒geheel getal

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

Geeft het dichtstbijliggende gehele getal dat $\geq$ is aan het argument.

Het argument kan een reëel of complex getal zijn.

Opmerking: zie ook `floor()`.

`ceiling(Lijst1)` $\Rightarrow$ *lijst*

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

Geeft een lijst of matrix met de 'plafondwaarde' van elk element.

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

centralDiff()

`centralDiff(Uitdr1, Var [=Waarde][, Stap])`
 $\Rightarrow$ *uitdrukking*

`centralDiff(Uitdr1, Var [, Stap])`
 $| Var = Waarde \Rightarrow$ *uitdrukking*

`centralDiff(Uitdr1, Var [=Waarde][, Lijst])`
 $\Rightarrow$ *lijst*

`centralDiff(Lijst1, Var [=Waarde][, Stap])`
 $\Rightarrow$ *lijst*

`centralDiff(Matrix1, Var [=Waarde][, Stap])`
 $\Rightarrow$ *matrix*

Geeft de numerieke afgeleide met behulp van de centraal-differentiequotiëntformule.

Wanneer *Waarde* gespecificeerd is, wordt elke eerdere variabeletoekenning of elke huidige “|”-substitutie voor de variabele onderdrukt.

Stap is de stapgrootte. Als *Stap* wordt weggelaten, is de standaardwaarde 0,001.

Wanneer u *Lijst1* of *Matrix1* gebruikt, dan wordt de bewerking toegepast op de waarden in de lijst of op de elementen in de matrix.

Opmerking: zie ook `en d()`.

$$\begin{array}{l} \text{centralDiff}(\cos(x), x, h) \\ \frac{-\cos(x-h) - \cos(x+h)}{2 \cdot h} \\ \lim_{h \rightarrow 0} (\text{centralDiff}(\cos(x), x, h)) \quad -\sin(x) \\ \text{centralDiff}(x^3, x, 0.01) \\ 3 \cdot (x^2 + 0.000033) \\ \text{centralDiff}(\cos(x), x) \Big|_{x=\frac{\pi}{2}} \quad -1. \\ \text{centralDiff}(x^2, x, \{0.01, 0.1\}) \\ \{2 \cdot x, 2 \cdot x\} \end{array}$$

cFactor(*Uitdr1*[,*Var*]) $\Rightarrow$ *uitdrukking*

cFactor(*Lijst1*[,*Var*]) $\Rightarrow$ *lijst*

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

cFactor(*Uitdr1*) geeft *Uitdr1* ontbonden ten opzichte van al zijn variabelen boven een gemeenschappelijke noemer.

Uitdr1 wordt zoveel mogelijk ontbonden in lineaire rationale factoren, zelfs als dit nieuwe niet-reële getallen oplevert. Dit alternatief is geschikt als u een ontbinding in factoren ten opzichte van meer dan één variabele wilt.

cFactor(*Uitdr1*,*Var*) geeft *Uitdr1* ontbonden ten opzichte van variabele *Var*.

Uitdr1 wordt zoveel mogelijk ontbonden in factoren die lineair zijn in *Var*, met misschien niet-reële constanten, zelfs als dit irrationale constanten of subuitdrukkingen die irrationaal zijn in andere variabelen oplevert.

De factoren en hun termen worden gesorteerd met *Var* als de hoofdvariabele. In elke factor worden gelijksoortige machten van *Var* samengenomen. Neem *Var* op als argument als u een ontbinding ten opzichte van alleen die variabele nodig heeft, en u bereid bent om irrationale uitdrukkingen in andere variabelen te accepteren om een ontbinding ten opzichte van *Var* te vergroten. Er kan wat onbedoelde ontbinding ten opzichte van andere variabelen optreden.

$\text{cFactor}(a^3 \cdot x^2 + a \cdot x^2 + a^3 + a \cdot x)$	$a \cdot (a^2 + 1) \cdot (x - i) \cdot (x + i)$
$\text{cFactor}\left(x^2 + \frac{4}{9}\right)$	$\frac{(3 \cdot x - 2 \cdot i) \cdot (3 \cdot x + 2 \cdot i)}{9}$
$\text{cFactor}(x^2 + 3)$	$x^2 + 3$
$\text{cFactor}(x^2 + a)$	$x^2 + a$

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

cFactor()

Catalogus >

Bij de automatische instelling van de **Automatische of Benaderende** modus kunt u door *Var* op te nemen tevens een benadering met drijvende komma-coëfficiënten mogelijk maken, waar irrationale coëfficiënten niet expliciet beknopt uitgedrukt kunnen worden in termen van de ingebouwde functies. Ook als er maar één variabele is, kan het opnemen van *Var* een vollediger ontbinding opleveren.

Opmerking: zie ook **factor()**.

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

Om het hele resultaat te zien drukt u op en gebruikt u vervolgens en om de cursor te verplaatsen.

char()

Catalogus >

char(*Geheel getal*)⇒*teken*

Geeft een tekenreeks die het teken met het nummer *Geheel getal* van de tekenserie van de rekenmachine bevat. Het geldige bereik voor *Geheel getal* is 0–65535.

$$\begin{array}{ll} \text{char}(38) & "&" \\ \text{char}(65) & "A" \end{array}$$

charPoly()

Catalogus >

charPoly(*vierkanteMatrix*,*Var*)
⇒*veelterm-uitdrukking*

charPoly(*vierkanteMatrix*,*Uitdr*)
⇒*veelterm-uitdrukking*

charPoly(*vierkanteMatrix1*,*Matrix2*)
⇒*veelterm-uitdrukking*

Geeft de karakteristieke veelterm van *vierkanteMatrix*. De karakteristieke veelterm van $n \times n$ matrix *A*, aangeduid door $p_A(\lambda)$, is de veelterm die gedefinieerd wordt door

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

waarbij *I* de $n \times n$ eenheidsmatrix aanduidt.

vierkanteMatrix1 en *vierkanteMatrix2* moeten gelijke afmetingen hebben.

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

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

Berekent een χ^2 -toets voor afhankelijkheid op de kruistabel van aantallen in de geobserveerde matrix *ObsMatrix*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Zie voor informatie over het effect van lege elementen in een matrix “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat. χ^2	Chi-kwadraat-statistiek: som (geobserveerd - verwacht) ² /verwacht.
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.df	Vrijheidsgraden van de chi-kwadraat-statistieken
stat.ExpMat	Matrix van de verwachte tabel met aantallen elementen, waarbij wordt uitgegaan van de nulhypothese
stat.CompMat	Matrix van chi-kwadraat-statistiekbijdragen van elementen

 χ^2 Cdf()

χ^2 Cdf(*ondergrens*,*bovengrens*,*df*) $\Rightarrow$ getal als *ondergrens* en *bovengrens* getallen zijn, *lijst* als *ondergrens* en *bovengrens* lijsten zijn

chi2Cdf(*ondergrens*,*bovengrens*,*df*) $\Rightarrow$ getal als *ondergrens* en *bovengrens* getallen zijn, *lijst* als *ondergrens* en *bovengrens* lijsten zijn

Berekent de χ^2 -verdelingskans tussen *ondergrens* en *bovengrens* voor de gespecificeerde vrijheidsgraden *df*.

Voor $P(X \leq \text{bovengrens})$ stelt u *ondergrens* = 0 in.

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267.)

 χ^2 GOF χ^2 GOF *obsLijst*,*explLijst*,*df*

chi2GOF *obsLijst,expLijst,df*

Voert een toets uit om te bevestigen dat de steekproefgegevens afkomstig zijn uit een populatie met de gespecificeerde verdeling. *obsLijst* is een lijst met aantallen en moet gehele getallen bevatten. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat. χ^2	Chi-kwadraat-statistiek: som (geobserveerd - verwacht) ² /verwacht
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.df	Vrijheidsgraden van de chi-kwadraat-statistieken
stat.ComplList	Chi-kwadraat-statistiekbijdragen van elementen

 χ^2 Pdf()

χ^2 Pdf(*XWaarde*,*df*) $\Rightarrow$ getal als *XWaarde* een getal is, *lijst* als *XWaarde* een lijst is

chi2Pdf(*XWaarde*,*df*) $\Rightarrow$ getal als *XWaarde* een getal is, *lijst* als *XWaarde* een lijst is

Berekent de kansdichtheidsfunctie (pdf) voor de χ^2 -verdeling bij een gespecificeerde *XWaarde* voor de gespecificeerde vrijheidsgraden *df*.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

ClearAZ**ClearAZ**

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

Wist alle variabelen die bestaan uit één teken in de huidige opgave.

<i>b</i>	5
----------	---

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

Als een aantal van de variabelen vergrendeld is, veroorzaakt deze opdracht een foutmelding en worden alleen de niet-vergrendelde variabelen gewist. Zie **unLock**, pag. 210.

<i>b</i>	<i>b</i>
----------	----------

ClrErr

Zie voor een voorbeeld van **ClrErr** Voorbeeld 2 onder het commando **Try** (pag. 203).

Wist de foutstatus en zet de systeemvariabele *errCode* op nul.

De **Else**-bepaling van het **Try...Else...EndTry**-blok moet **ClrErr** of **PassErr** gebruiken. Als de fout verwerkt of genegeerd moet worden, gebruik dan **ClrErr**. Als onbekend is wat er met de fout gedaan moet worden, gebruik dan **PassErr** om hem te verzenden naar de volgende foutenafhandelaar. Als er geen onbesliste **Try...Else...EndTry**-foutenafhandelaars meer zijn, wordt het foutendialoogvenster weergegeven zoals normaal is.

Opmerking: zie ook **PassErr**, pag. 139 en **Try**, pag. 203.

Opmerking bij het invoeren van het voorbeeld:

Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

colAugment()

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

Geeft een nieuwe matrix die bestaat uit *Matrix2* toegevoegd aan *Matrix1*. De matrices moeten evenveel kolommen hebben, en *Matrix2* wordt toegevoegd aan *Matrix1* als nieuwe rijen. Dit verandert *Matrix1* of *Matrix2* niet.

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

colDim()

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

Geeft het aantal kolommen in *Matrix*.

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

Opmerking: zie ook **rowDim()**.

colNorm()

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

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

Geeft het maximum van de sommen van de absolute waarden van de elementen in de kolommen in *Matrix*.

Opmerking: onbepaalde matrixelementen zijn niet toegestaan. Zie ook **rowNorm()**.

comDenom()

comDenom(*Uitdr1*[, *Var*]) $\Rightarrow$ uitdrukking

comDenom(*Lijst1*[, *Var*]) $\Rightarrow$ lijst

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

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

$$\frac{x^2 \cdot y^2 + x^2 \cdot y + 2 \cdot x \cdot y^2 + 2 \cdot x \cdot y + 2 \cdot y^2 + 2 \cdot y}{x^2 + 2 \cdot x + 1}$$

comDenom(*Uitdr1*) geeft een vereenvoudigde breuk met een volledig uitgewerkte teller boven een volledig uitgewerkte noemer.

comDenom(*Uitdr1*, *Var*) geeft een vereenvoudigde breuk met teller en noemer die uitgewerkt zijn ten opzichte van *Var*. De termen en hun factoren worden gesorteerd met *Var* als de hoofdvariabele. Gelijke machten van *Var* worden samengenomen. Er kan wat onbedoelde ontbinding ten opzichte van de samengenomen coëfficiënten optreden. Vergeleken met het weglaten van *Var* scheelt dit vaak tijd, geheugen en schermruimte, terwijl de uitdrukking begrijpelijker wordt. Ook worden volgende bewerkingen op het resultaat sneller en putten deze het geheugen minder uit.

Als *Var* niet voorkomt in *Uitdr1*, dan geeft **comDenom**(*Uitdr1*, *Var*) een vereenvoudigde breuk met een niet-uitgewerkte teller boven een niet-uitgewerkte noemer. Dergelijke resultaten besparen meestal nog meer tijd, geheugen en schermruimte. Dergelijke gedeeltelijk ontbonden resultaten maken volgende bewerkingen op het resultaat tevens sneller en minder belastend voor het geheugen.

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

$$\frac{x^2 \cdot y \cdot (y+1) + 2 \cdot x \cdot y \cdot (y+1) + 2 \cdot y \cdot (y+1)}{x^2 + 2 \cdot x + 1}$$

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

$$\frac{y^2 \cdot (x^2 + 2 \cdot x + 2) + y \cdot (x^2 + 2 \cdot x + 2)}{x^2 + 2 \cdot x + 1}$$

$$\text{Define } \text{comden}(\text{exprn}) = \text{comDenom}(\text{exprn}, abc)$$

Done

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

Ook wanneer er geen noemer is, is de **comden**-functie vaak een snelle manier om gedeeltelijke ontbinding te bereiken als **factor()** te langzaam is of als deze het geheugen uitput.

Tip: Voer deze **comden()**-functiedefinitie in en probeer hem volgens de regels uit als alternatief voor **comDenom()** en **factor()**.

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

completeSquare ()

completeSquare(UitdrOfVgl, Var)

⇒ *uitdrukking of vergelijking*

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

completeSquare(UitdrOfVgl, Var^Macht)

⇒ *uitdrukking of vergelijking*

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

completeSquare(UitdrOfVgl, Var1, Var2

[...]) ⇒ *uitdrukking of vergelijking*

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

completeSquare(UitdrOfVgl, {Var1, Var2

[...]) ⇒ *uitdrukking of vergelijking*

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

Converteert een kwadratische veeltermuitdrukking van de vorm $a \cdot x^2 + b \cdot x + c$ naar de vorm $a \cdot (x-h)^2 + k$

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

- of -

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

Converteert een kwadratische vergelijking van de vorm $a \cdot x^2 + b \cdot x + c = d$ naar de vorm $a \cdot (x-h)^2 = k$

Het eerste argument moet een kwadratische uitdrukking of vergelijking in standaardvorm zijn gerelateerd aan het tweede argument.

Het tweede argument moet een enkelvoudige term in één variabele of een enkelvoudige term in één variabele die verheven is tot een rationale macht zijn, bijvoorbeeld x , y^2 of $z^{1/3}$.

De derde en vierde syntax proberen het kwadraat af te splitsen met betrekking tot de variabelen *Var1*, *Var2* [...]).

conj()		Catalogus > 
conj (<i>Uitdr1</i>) $\Rightarrow$ <i>uitdrukking</i>	$\text{conj}(1+2\cdot i)$	$1-2\cdot i$
conj (<i>Lijst1</i>) $\Rightarrow$ <i>lijst</i>	$\text{conj}\left(\begin{bmatrix} 2 & 1-3\cdot i \\ -i & -7 \end{bmatrix}\right)$	$\begin{bmatrix} 2 & 1+3\cdot i \\ i & -7 \end{bmatrix}$
conj (<i>Matrix1</i>) $\Rightarrow$ <i>matrix</i>	$\text{conj}(z)$	$\bar{z}$
Geeft de complex geconjugeerde van het argument.	$\text{conj}(x+i\cdot y)$	$x-y\cdot i$
Opmerking: alle onbepaalde variabelen worden behandeld als reële variabelen.		

constructMat()		Catalogus > 
constructMat (<i>Uitdr</i> , <i>Var1</i> , <i>Var2</i> , <i>aantalRijen</i> , <i>aantalKolommen</i>) $\Rightarrow$ <i>matrix</i>	$\text{constructMat}\left(\frac{1}{i+j}, i, j, 3, 4\right)$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$
Geeft een matrix op basis van de argumenten.		
<i>Uitdr</i> is een uitdrukking in de variabelen <i>Var1</i> en <i>Var2</i> . Elementen in de resulterende matrix worden gevormd door <i>Uitdr</i> uit te werken voor elke opgehoogde waarde van <i>Var1</i> en <i>Var2</i> .		
<i>Var1</i> wordt automatisch verhoogd van 1 tot en met <i>aantalRijen</i> . Binnen elke rij wordt <i>Var2</i> verhoogd van 1 tot en met <i>aantalKolommen</i> .		

CopyVar()		Catalogus > 
CopyVar <i>Var1</i> , <i>Var2</i>	Define $a(x)=\frac{1}{x}$	<i>Done</i>
CopyVar <i>Var1</i> ., <i>Var2</i> .	Define $b(x)=x^2$	<i>Done</i>
CopyVar <i>Var1</i> , <i>Var2</i> kopieert de waarde van variabele <i>Var1</i> naar variabele <i>Var2</i> , waarbij <i>Var2</i> indien nodig gecreëerd wordt. Variabele <i>Var1</i> moet een waarde hebben.	CopyVar <i>a</i> , <i>c</i> : $c(4)$	$\frac{1}{4}$
	CopyVar <i>b</i> , <i>c</i> : $c(4)$	16

Als *Var1* de naam van een bestaande, door de gebruiker gedefinieerde functie is, kopieert CopyVar de definitie van die functie naar functie *Var2*. Functie *Var1* moet gedefinieerd zijn.

Var1 moet voldoen aan de naamgevingsvereisten of moet een indirecte uitdrukking zijn die vereenvoudigd wordt tot een variabelenaam die voldoet aan de vereisten.

CopyVar *Var1.*, *Var2.* kopieert alle leden van de variabelegroep *Var1.* naar de groep *Var2.*, waarbij *Var2.* indien nodig wordt gecreëerd.

Var1 moet de naam van een bestaande variabelegroep zijn, zoals de statistische *stat.nn*-resultaten of variabelen die gecreëerd zijn met de **LibShortcut()**-functie. Als *Var2* reeds bestaat, dan vervangt deze opdracht alle elementen die beide groepen gemeenschappelijk hebben, en worden de elementen die nog niet bestaan toegevoegd. Als één of meer elementen van *Var2.* vergrendeld zijn, dan blijven alle elementen van *Var2.* ongewijzigd.

<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(*Lijst1,Lijst2[,...[,Lijst20]]*)

Berekent de correlatiematrix voor de matrix bestaande uit [*Lijst1, Lijst2, ..., Lijst20*].

Uitdr ►cos

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>cos in te typen.

Geeft *Uitdr* weer in termen van cosinus. Dit is een operator voor weergaveconversie. Deze kan alleen op het eind van de invoerregel gebruikt worden.

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

►cos verlaagt alle machten van sin(...) modulo $1 - \cos(\dots)^2$ zodat alle resterende machten van cos(...) exponenten in het bereik (0, 2) hebben. Het resultaat zal dus vrij zijn van sin(...) dan en slechts dan als sin(...) uitsluitend voorkomt in de gegeven uitdrukking met even exponenten.

Opmerking: deze conversie-operator wordt niet ondersteund in de hoekmodi Graden en Decimale graden. Voordat u deze operator gebruikt, dient u ervoor te zorgen dat de hoekmodus is ingesteld op Radialen, en dat *Uitdr* geen expliciete verwijzingen naar graden of decimale graden bevat.

cos()

 -toets

cos(*Uitdr1*) ⇒ *uitdrukking*

In de hoekmodus Graden:

cos(*Lijst1*) ⇒ *lijst*

$$\cos\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

cos(*Uitdr1*) geeft de cosinus van het argument als een uitdrukking.

$$\cos(45) \quad \frac{\sqrt{2}}{2}$$

cos(*Lijst1*) geeft een lijst van de cosinussen van alle elementen in *Lijst1*.

$$\cos\{0,60,90\} \quad \left\{1, \frac{1}{2}, 0\right\}$$

Opmerking: het argument wordt geïnterpreteerd als een hoek in graden, decimale graden of radialen volgens de huidige hoekmodus-instelling. U kunt °, G of r gebruiken om de hoekmodus tijdelijk te onderdrukken.

In de hoekmodus Decimale graden:

$$\cos\{0,50,100\} \quad \left\{1, \frac{\sqrt{2}}{2}, 0\right\}$$

In de hoekmodus Radialen:

$$\cos\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2}$$

$$\cos(45^\circ) \quad \frac{\sqrt{2}}{2}$$

cos(*vierkanteMatrix1*) ⇒ *vierkanteMatrix*

In de hoekmodus Radialen:

Geeft de matrixcosinus van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de cosinus van elk element.

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

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

Wanneer een scalaire functie $f(A)$ werkt op *vierkanteMatrix1* (A), dan wordt het resultaat berekend door het volgende algoritme:

Bereken de eigenwaarden (λ_i) en de eigenvectoren (V_i) van A .

vierkanteMatrix1 moet diagonaliseerbaar zijn. Bovendien kan hij geen symbolische variabelen hebben die geen waarde toegekend hebben gekregen.

Vorm de matrices:

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

Vervolgens $A = X B X^{-1}$ en $f(A) = X f(B) X^{-1}$.
Bijvoorbeeld $\cos(A) = X \cos(B) X^{-1}$ waarbij:

$\cos(B) =$

$$\begin{bmatrix} \cos(\lambda_1) & 0 & \dots & 0 \\ 0 & \cos(\lambda_2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \cos(\lambda_n) \end{bmatrix}$$

Alle berekeningen worden uitgevoerd met behulp van drijvende komma-rekenkunde.

$\cos^{-1}(\text{Uitdr1}) \Rightarrow \text{uitdrukking}$

In de hoekmodus Graden:

$\cos^{-1}(\text{Lijst1}) \Rightarrow \text{lijst}$

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

$\cos^{-1}(\text{Uitdr1})$ geeft de hoek waarvan de cosinus *Uitdr1* als een uitdrukking is.

In de hoekmodus Decimale graden:

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

cos⁻¹(Lijst1) geeft een lijst van de inverse cosinussen van elk element van *Lijst1*.

Opmerking: de uitkomst wordt in graden, decimale graden of radialen gegeven, volgens de ingestelde hoekmodus.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arccos (...)** in te typen.

cos⁻¹(vierkanteMatrix1)⇒vierkanteMatrix

Geeft de inverse matrixcosinus van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de inverse cosinus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Radialen:

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

In de hoekmodus Radialen en rechthoekige complexe opmaak:

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

Om het hele resultaat te zien drukt u op **▲** en gebruikt u vervolgens **◀** en **▶** om de cursor te verplaatsen.

cosh(Uitdr1)⇒uitdrukking

cosh(Lijst1)⇒lijst

cosh(Uitdr1) geeft de cosinus hyperbolicus van het argument als een uitdrukking.

cosh(Lijst1) geeft een lijst van de cosinussen hyperbolicus van elk element van *Lijst1*.

cosh(vierkanteMatrix1)⇒vierkanteMatrix

Geeft de matrixcosinus hyperbolicus van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de cosinus hyperbolicus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Graden:

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

In de hoekmodus Radialen:

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

cosh⁻¹()**cosh⁻¹(Uitdr1)** ⇒ uitdrukking

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

cosh⁻¹(Lijst1) ⇒ lijst

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

cosh⁻¹(Uitdr1) geeft de inverse cosinus hyperbolicus van het argument als een uitdrukking.

cosh⁻¹(Lijst1) geeft een lijst van de inverse cosinussen hyperbolicus van elk element van *Lijst1*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arccosh (...)** in te typen.

cosh⁻¹(vierkanteMatrix1)
⇒ vierkanteMatrix

Geeft de inverse matrixcosinus hyperbolicus van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de inverse cosinus hyperbolicus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Radialen en rechthoekige complexe opmaak:

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

Om het hele resultaat te zien drukt u op **►** en gebruikt u vervolgens **◀** en **►** om de cursor te verplaatsen.

cot()**cot(Uitdr1)** ⇒ uitdrukking

In de hoekmodus Graden:

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

cot(Lijst1) ⇒ lijst

Geeft de cotangens van *Uitdr1* of geeft een lijst van de cotangensen van alle elementen in *Lijst1*.

In de hoekmodus Decimale graden:

$$\cot(50) \quad 1$$

In de hoekmodus Radialen:

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

Opmerking: het argument wordt geïnterpreteerd als een hoek in graden, decimale graden of radialen volgens de huidige hoekmodus-instelling. U kunt $^{\circ}$, $^{\text{G}}$ of $^{\text{r}}$ gebruiken om de hoekmodus tijdelijk te onderdrukken.

 $\cot^{-1}()$

$\cot^{-1}(\text{Uitdr1}) \Rightarrow \text{uitdrukking}$

In de hoekmodus Graden:

$\cot^{-1}(\text{Lijst1}) \Rightarrow \text{lijst}$

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

Geeft de hoek waarvan de cotangens *Uitdr1* is of geeft een lijst met de inverse cotangens van elk element in *Lijst1*.

In de hoekmodus Decimale graden:

Opmerking: de uitkomst wordt in graden, decimale graden of radialen gegeven, volgens de ingestelde hoekmodus.

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

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arccot (...)** in te typen.

In de hoekmodus Radialen:

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

coth()

$\text{coth}(\text{Uitdr1}) \Rightarrow \text{uitdrukking}$

$\text{coth}(1.2)$	1.19954
--------------------	---------

$\text{coth}(\text{Lijst1}) \Rightarrow \text{lijst}$

$\text{coth}(\{1, 3.2\})$	$\left\{ \frac{1}{\tanh(1)}, 1.00333 \right\}$
---------------------------	--

Geeft de cotangens hyperbolicus van *Uitdr1* of geeft een lijst van de cotangensen hyperbolicus van alle elementen in *Lijst1*.

coth⁻¹()Catalogus > **coth⁻¹(Uitdr1)**⇒uitdrukkingcoth⁻¹(3,5)

0.293893

coth⁻¹(Lijst1)⇒lijstcoth⁻¹({-2,2,1,6})

$$\left\{ \frac{-\ln(3)}{2}, 0.518046, \frac{\ln\left(\frac{7}{5}\right)}{2} \right\}$$

Geeft de inverse cotangens hyperbolicus van *Uitdr1* of geeft een lijst met de inverse cotangensen hyperbolicus van *Lijst1*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arccoth(...)** in te typen.

count()Catalogus > **count(Waarde1 of Lijst1 [,Waarde2 of Lijst2 [,...]])**⇒waarde

count(2,4,6)

3

count({2,4,6})

3

Geeft het samengenomen aantal van alle elementen in de argumenten die uitgewerkt worden tot numerieke waarden.

count(2,{4,6},{8 10
12 14})

7

Elk argument kan een uitdrukking, waarde, lijst of matrix zijn. U kunt gegevenstypen mengen en argumenten met verschillende afmetingen gebruiken.

count(1/2,3+4*i,undef,"hello",x+5,sign(0))

2

Bij een lijst, matrix of reeks cellen wordt elk element uitgewerkt om te bepalen of het moet worden opgenomen in de telling.

In het laatste voorbeeld worden alleen 1/2 en 3+4*i geteld. De resterende argumenten worden niet uitgewerkt naar numerieke waarden, als we aannemen dat *x* onbepaald is.

In de toepassing Lijsten & Spreadsheet kunt u een reeks cellen op de plaats van elk argument gebruiken.

Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

countif()Catalogus > **countif(Lijst,Criteria)**⇒waarde

countif({1,3,"abc",undef,3,1},3)

2

Geeft het samengenomen aantal van alle elementen in *Lijst* die voldoen aan de gespecificeerde *Criteria*.

Telt het aantal elementen dat gelijk is aan 3.

Criteria kan zijn:

countif({"abc","def","abc",3},"def")

1

- Een waarde, uitdrukking of tekenreeks.
Bijvoorbeeld: **3** telt alleen die elementen

Telt het aantal elementen dat gelijk is aan "def".

in *Lijst* die vereenvoudigd worden tot de waarde 3.

- Een Booleaanse uitdrukking met het symbool ? als tijdelijke plaatsaanduiding voor elk element. Bijvoorbeeld, ?<5 telt alleen die elementen in *Lijst* die kleiner zijn dan 5.

In de toepassing Lijsten & Spreadsheet kunt u een reeks cellen gebruiken op de plaats van *Lijst*.

Lege elementen in de lijst worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

Opmerking: zie ook **sumlf()**, pag. 192 en **frequency()**, pag. 79.

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

Telt het aantal elementen dat gelijk is aan x ; in dit voorbeeld wordt ervan uitgegaan dat de variabele x onbepaald is.

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

Telt 1 en 3.

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

Telt 3, 5 en 7.

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

Telt 1, 3, 7 en 9.

cPolyRoots()

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

cPolyRoots(*LijstVanCoëff*) $\Rightarrow$ *lijst*

De eerste syntax, **cPolyRoots**(*Poly*, *Var*), geeft een lijst met complexe oplossingen van de veelterm (polynoom) *Poly* voor de variabele *Var*.

Poly moet een veelterm met één variabele zijn.

De tweede syntax, **cPolyRoots**(*LijstVanCoëff*), geeft een lijst met complexe oplossingen voor de coëfficiënten in *LijstVanCoëff*.

Opmerking: zie ook **polyRoots()**, pag. 144.

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

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

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

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

crossP()Catalogus > **crossP(Lijst1, Lijst2) ⇒ lijst**

Geeft het uitwendige product van *Lijst1* en *Lijst2* als een lijst.

Lijst1 en *Lijst2* moeten gelijke afmetingen hebben, en de afmeting moet 2 of 3 zijn.

crossP(Vector1, Vector2) ⇒ vector

Geeft een rij- of kolomvector (afhankelijk van de argumenten) die het uitwendig product is van *Vector1* en *Vector2*.

Zowel *Vector1* als *Vector2* moeten rijvectoren zijn, of beide moeten kolomvectoren zijn. Beide vectoren moeten gelijke afmetingen hebben, en de afmeting moet 2 of 3 zijn.

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

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

$$\text{crossP}(\{0.1, 2.2, -5\}, \{1, -0.5, 0\})$$

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

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

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

csc() -toets**csc(Uitdr1) ⇒ uitdrukking**

In de hoekmodus Graden:

csc(Lijst1) ⇒ lijst

Geeft de cosecans van *Uitdr1* of geeft een lijst met de cosecans van alle elementen in *Lijst1*.

$$\text{csc}(45)$$

$$\sqrt{2}$$

In de hoekmodus Decimale graden:

$$\text{csc}(50)$$

$$\sqrt{2}$$

In de hoekmodus Radialen:

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

$$\left\{\frac{1}{\sin(1)}, 1, \frac{2 \cdot \sqrt{3}}{3}\right\}$$

csc⁻¹() -toets**csc⁻¹(Uitdr1) ⇒ uitdrukking**

In de hoekmodus Graden:

csc⁻¹(Lijst1) ⇒ lijst

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

$$90.$$

Geeft de hoek waarvan de cosecans *Uitdr1* is of geeft een lijst met de inverse cosecans van elk element in *Lijst1*.

In de hoekmodus Decimale graden:

csc⁻¹()**trig-toets**

Opmerking: de uitkomst wordt in graden, decimale graden of radialen gegeven, volgens de ingestelde hoekmodus.

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

100.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arccsc(...)** in te typen.

In de hoekmodus Radialen:

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

csch()**Catalogus > **

csch(Uitdr1) ⇒ uitdrukking

$$\text{csch}(3)$$

$$\frac{1}{\sinh(3)}$$

csch(Lijst1) ⇒ lijst

$$\text{csch}\left(\left\{1,2,1,4\right\}\right)$$

$$\left\{\frac{1}{\sinh(1)}, 0.248641, \frac{1}{\sinh(4)}\right\}$$

Geeft de cosecans hyperbolicus van *Uitdr1* of geeft een lijst van de cosecansen hyperbolicus van alle elementen in *Lijst1*.

csch⁻¹()**Catalogus > **

csch⁻¹(Uitdr1) ⇒ uitdrukking

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

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

csch⁻¹(Lijst1) ⇒ lijst

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

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

Geeft de inverse cosecans hyperbolicus van *Uitdr1* of geeft een lijst met de inverse cosecans hyperbolicus van elk element in *Lijst1*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arccsch(...)** in te typen.

cSolve()**Catalogus > **

cSolve(Vergelijking, Var) ⇒ Booleaanse uitdrukking

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

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

cSolve(Vergelijking, Var=Gok) ⇒ Booleaanse uitdrukking

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

$$x=-1$$

cSolve(Ongelijkheid, Var) ⇒ Booleaanse uitdrukking

Geeft mogelijke complexe oplossingen van een vergelijking of ongelijkheid voor *Var*. Het doel is het produceren van mogelijkheden voor alle reële en niet-reële oplossingen. Ook als *Vergelijking* reëel is, staat **cSolve()** niet-reële resultaten toe in complexe opmaak van reële resultaten.

cSolve() stelt het domein tijdelijk in op complex tijdens het oplossen, zelfs als het huidige domein reëel is. In het complexe domein gebruiken gebroken machten met oneven noemers de principaal in plaats van de reële tak. Daardoor zijn oplossingen vanuit **solve()** van vergelijkingen met dergelijke gebroken machten niet noodzakelijk een deelverzameling van die van **cSolve()**.

cSolve() start met exacte symbolische methodes. **cSolve()** gebruikt indien nodig ook een iteratieve benaderende complexe ontbinding.

Opmerking: zie ook **cZeros()**, **solve()** en **zeros()**.

cSolve(*Vgl1* and *Vgl2* [and...], *VarOfGok1*, *VarOfGok2* [, ...])
⇒ *Booleaanse uitdrukking*

cSolve(*StelselVanVgl*, *VarOfGok1*, *VarOfGok2* [, ...])
⇒ *Booleaanse uitdrukking*

Geeft mogelijke complexe oplossingen voor stelsels algebraïsche vergelijkingen, waarbij elke *VarOfGok* een variabele specificeert waarnaar u wilt oplossen.

U kunt optioneel een begingok voor een variabele specificeren. Elke *VarOfGok* moet de volgende vorm hebben:

variabele

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

In de cijferweergavemodus Vast 2:

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

Om het hele resultaat te zien drukt u op  en gebruikt u vervolgens  en  om de cursor te verplaatsen.

– of –

variabele = reëel of niet-reëel getal

Bijvoorbeeld: x is geldig en ook $x=3+i$.

Als alle vergelijkingen veeltermen zijn en als u GEEN begingokken specificeert, dan gebruikt **cSolve()** de lexicale Gröbner/Buchberger-eliminatiemethode om te proberen **alle** complexe oplossingen te bepalen.

Complexe oplossingen kunnen zowel reële als niet-reële oplossingen bevatten, zoals te zien is in het voorbeeld rechts.

$$\text{cSolve}(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v\})$$

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

Om het hele resultaat te zien drukt u op **►** en gebruikt u vervolgens **◀** en **►** om de cursor te verplaatsen.

Stelsels van veeltermvergelijkingen kunnen extra variabelen hebben die geen waarden bevatten, maar die gegeven numerieke waarden voorstellen die later gesubstitueerd kunnen worden.

$$\text{cSolve}(u \cdot v - u = c \cdot v \text{ and } v^2 = -u, \{u, v\})$$

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

U kunt ook oplossingsvariabelen opnemen die niet voorkomen in de vergelijkingen. Deze oplossingen laten zien hoe families van oplossingen willekeurige constanten zouden kunnen bevatten van de vorm ck , waarbij k een geheel getal-suffix van 1 tot en met 255 is.

$$\text{cSolve}(u \cdot v - u = v \text{ and } v^2 = -u, \{u, v, w\})$$

$$u = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } w = c13 \text{ or } u = \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \text{ and } v = \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \text{ and } w = c13$$

Om het hele resultaat te zien drukt u op **►** en gebruikt u vervolgens **◀** en **►** om de cursor te verplaatsen.

Bij stelsels veeltermen kan de berekeningstijd of de belasting van het geheugen sterk afhangen van de volgorde waarin u de oplossingsvariabelen plaatst. Als uw eerste keuze het geheugen uitput of teveel van uw geduld vraagt, probeer de variabelen in de vergelijkingen en/of *VarOfGok* dan te herschikken.

Als u geen gokken opneemt en als een vergelijking in enige variabele geen veelterm is, maar als alle vergelijkingen lineair zijn in alle oplossingsvariabelen, dan gebruikt **cSolve()** Gaussische eliminatie om te proberen alle oplossingen te bepalen.

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

Als een stelsel noch in al zijn variabelen veeltermen bevat, noch lineair in zijn oplossingsvariabelen is, dan bepaalt **cSolve()** maximaal één oplossing met behulp van een benaderende iteratieve methode. Om dit te doen moet het aantal oplossingsvariabelen gelijk zijn aan het aantal vergelijkingen, en moeten alle andere variabelen in de vergelijkingen vereenvoudigd worden tot getallen.

$$\text{cSolve}\left(e^z=w \text{ and } w=z^2, \{w, z\}\right) \\ w=0.494866 \text{ and } z=0.703467$$

Er is dikwijls een niet-reële gok nodig om een niet-reële oplossing te bepalen. Voor convergentie moet een gok mogelijk vrij dicht bij een oplossing liggen.

$$\text{cSolve}\left(e^z=w \text{ and } w=z^2, \{w, z=1+i\}\right) \\ w=0.149606+4.8919 \cdot i \text{ and } z=1.58805+1.5402 \cdot i$$

Om het hele resultaat te zien drukt u op  en gebruikt u vervolgens  en  om de cursor te verplaatsen.

CubicReg

CubicReg $X, Y, [Freq] [, Categorie, Opnemen]$

Berekent de derdegraads veeltermregressie $y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$ op de lijsten X en Y met frequentie $Freq$. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en Y zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig X - en Y -punt voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriëcodes voor de overeenkomstige X - en Y -gegevens.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a \cdot x^3 + b \cdot x^2 + c \cdot x + d$
stat.a, stat.b, stat.c, stat.d	Regressiecoëfficiënten
stat.R ²	Determinatiecoëfficiënt
stat.Resid	Residuen uit de regressie
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

cumulativeSum()

cumulativeSum(Lijst1) ⇒ lijst

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

Geeft een lijst met de cumulatieve sommen van de elementen in *Lijst1*, beginnend bij element 1.

cumulativeSum(Matrix1) ⇒ matrix

Geeft een matrix van de cumulatieve sommen van de elementen van *Matrix1*. Elk element is de cumulatieve som van de kolom, van boven naar beneden.

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

Een leeg element in *Lijst1* of *Matrix1* levert een leeg element in de resulterende lijst of matrix op. Voor meer informatie over lege elementen, zie pag. 267.

Cycle

Brengt de besturing onmiddellijk naar de volgende iteratie van de huidige lus (**For**, **While** of **Loop**).

Cycle is niet toegestaan buiten de drie lusstructuren (**For**, **While** of **Loop**).

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Functies die de gehele getallen van 1 tot 100 optelt, waarbij 50 wordt overgeslagen.

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

Vector ► **Cylind**

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>**Cylind** in te typen.

Geef de rij- of kolomvector in cilindrische vorm weer $[r, \angle \theta, z]$.

Vector moet exact drie elementen hebben. De vector kan een rij of een kolom zijn.

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

cZeros()

cZeros(Uitdr, Var) ⇒ lijst

Geeft een lijst met mogelijke reële en niet-reële waarden van *Var* die ervoor zorgen dat *Uitdr*=0 is. **cZeros()** doet dit door

explist(cSolve(Uitdr=0,Var),Var) te berekenen. Voor het overige is **cZeros()** gelijk aan **zeros()**.

Opmerking: zie ook **cSolve()**, **solve()** en **zeros()**.

cZeros({Uitdr1, Uitdr2 [, ...] }, {VarOfGok1, VarOfGok2 [, ...] }) ⇒ matrix

In de cijferweergavemodus Vast 3:

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

Om het hele resultaat te zien drukt u op $\blacktriangle$ en gebruikt u vervolgens $\blacktriangleleft$ en $\blacktriangleright$ om de cursor te verplaatsen.

Geeft mogelijke posities waarbij de uitdrukkingen gelijktijdig nul zijn. Elke *VarOfGok* specificeert een onbekende waarvan u de waarde zoekt.

U kunt optioneel een begingok voor een variabele specificeren. Elke *VarOfGok* moet de volgende vorm hebben:

variabele

– of –

variabele = *reëel of niet-reëel getal*

Bijvoorbeeld: x is geldig en ook $x=3+i$.

Als alle uitdrukkingen veeltermen zijn en als u GEEN begingokken specificeert, dan gebruikt **cZeros()** de lexicale Gröbner/Buchberger-eliminatiemethode om te proberen **alle** complexe nulpunten te bepalen.

Complexe oplossingen kunnen zowel reële als niet-reële nulpunten bevatten, zoals te zien is in het voorbeeld rechts.

Elke rij van de resulterende matrix representeert een alternatief nulpunt, met de componenten op dezelfde manier geordend als in de lijst *VarOfGok*. Om een rij te extraheren, indexeert u de matrix met $[rij]$.

Stelsels van veeltermen kunnen extra variabelen hebben die geen waarden hebben, maar die gegeven numerieke waarden voorstellen die later gesubstitueerd kunnen worden.

U kunt ook onbekende variabelen opnemen die niet voorkomen in de uitdrukkingen. Deze nulpunten laten zien hoe families van nulpunten willekeurige constanten zouden kunnen bevatten van de vorm ck , waarbij k een geheel getal-suffix van 1 tot en met 255 is.

$$\text{cZeros}\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \{u, v\}\right)$$

$$\begin{bmatrix} 0 & 0 \\ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i \end{bmatrix}$$

Extraheer rij 2:

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

$$\text{cZeros}\left(\left\{u \cdot v - u - c \cdot v^2, v^2 + u\right\}, \{u, v\}\right)$$

$$\begin{bmatrix} 0 & 0 \\ -(c-1)^2 & -(c-1) \end{bmatrix}$$

$$\text{cZeros}\left(\left\{u \cdot v - u - v, v^2 + u\right\}, \{u, v, w\}\right)$$

$$\text{cZero}\left(\left\{u \cdot (v-1) - v, u + v^2\right\}, \{u, v, w\}\right)$$

$$\begin{bmatrix} 0 & 0 & c4 \\ \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & c4 \\ \frac{1}{2} + \frac{\sqrt{3}}{2} \cdot i & \frac{1}{2} - \frac{\sqrt{3}}{2} \cdot i & c4 \end{bmatrix}$$

Bij stelsels veeltermen kan de berekeningstijd of de belasting van het geheugen sterk afhangen van de volgorde waarin u de onbekende variabelen plaatst. Als uw eerste keuze het geheugen uitput of teveel van uw geduld vraagt, probeer de variabelen in de uitdrukkingen en/of de lijst *VarOfGok* dan te herschikken.

Als u geen gokken opneemt en als een uitdrukking geen veelterm is in enige variabele, maar alle vergelijkingen lineair zijn in alle onbekende variabelen, dan gebruikt **cZeros()** Gaussische eliminatie om te proberen alle nulpunten te bepalen.

Als een stelsel noch veeltermen bevat in al zijn variabelen, noch lineair in zijn onbekenden is, dan bepaalt **cZeros()** maximaal één nulpunt met behulp van een benaderende iteratieve methode. Om dit te doen moet het aantal onbekende variabelen gelijk zijn aan het aantal uitdrukkingen, en moeten alle andere variabelen in de uitdrukkingen vereenvoudigd worden tot getallen.

Er is dikwijls een niet-reële gok nodig om een niet-reëel nulpunt te bepalen. Voor convergentie moet een gok mogelijk vrij dicht bij een nulpunt liggen.

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

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

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

D

dbd()

dbd(*datum1*,*datum2*)⇒*waarde*

Geeft het aantal dagen tussen *datum1* en *datum2* met behulp van de actuele-dag-telmethode.

datum1 en *datum2* kunnen getallen of lijsten met getallen zijn binnen het bereik van de datums op de standaard kalender. Als zowel *datum1* als *datum2* lijsten zijn, dan moeten deze dezelfde lengte hebben.

datum1 en *datum2* moeten tussen de jaren 1950 tot en met 2049 liggen.

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

U kunt de datums in twee notaties invoeren. De plaatsing van de decimale punt onderscheidt de datumnotaties.

MM.DDJJ (algemeen gebruikte notatie in de Verenigde Staten)

DDMM.JJ (algemeen gebruikte notatie in Europa)

►DDCatalogus >

Uitdr1 ►DD⇒waarde

In de hoekmodus Graden:

Lijst1 ►DD⇒lijst

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

Matrix1 ►DD⇒matrix

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>DD in te typen.

Geeft het decimale equivalent van het argument, uitgedrukt in graden. Het argument is een getal, lijst of matrix die op basis van de hoekmodus-instelling geïnterpreteerd wordt als decimale graden, radialen of graden.

In de hoekmodus Decimale graden:

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

In de hoekmodus Radialen:

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

►DecimalCatalogus >

Uitdrukking1 ►Decimal⇒uitdrukking

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

Lijst1 ►Decimal⇒uitdrukking

Matrix1 ►Decimal⇒uitdrukking

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>Decimal in te typen.

Geeft het argument in decimale vorm weer. Deze operator kan alleen op het eind van de invoerregel gebruikt worden.

Define *Var* = *Uitdrukking*

Define *Functie*(*Param1*, *Param2*, ...) =
Uitdrukking

Definieert de variabele *Var* of de door de gebruiker gedefinieerde functie *Functie*.

Parameters, zoals *Param1*, vormen de plaats voor het doorgeven van argumenten aan de functie. Bij het oproepen van een door de gebruiker gedefinieerde functie moet u argumenten opgeven (bijvoorbeeld waarden of variabelen) die overeenkomen met de parameters. De functie werkt, wanneer deze wordt aangeroepen, *Uitdrukking* uit met de opgegeven argumenten.

Var en *Functie* kunnen niet de naam van een systeemvariabele of van een ingebouwde functie of commando zijn.

Opmerking: deze vorm van **Define** staat gelijk aan het uitvoeren van de uitdrukking: *uitdrukking* → *Functie*(*Param1*, *Param2*).

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

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

In deze vorm kan de door de gebruiker gedefinieerde functie of programma een blok van meerdere beweringen uitvoeren.

Blok kan zowel een enkele bewering als een serie beweringen op aparte regels zijn. *Blok* kan ook uitdrukkingen en instructies (zoals **If**, **Then**, **Else** en **For**) bevatten.

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

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

Define (Definiëren)

Catalogus > 

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Opmerking: Zie ook **Define LibPriv**, pag. 50 en **Define LibPub**, pag. 50.

```
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

Catalogus > 

Define LibPriv *Var = Uitdrukking*

Define LibPriv *Functie(Param1, Param2, ...) =*
Uitdrukking

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

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

Werkt hetzelfde als **Define**, behalve dat er een persoonlijke bibliotheekvariabele, -functie of -programma wordt gecreëerd. Persoonlijke functies en programma's verschijnen niet in de Catalogus.

Opmerking: Zie ook **Define**, pag. 49 en **Define LibPub**, pag. 50.

Define LibPub

Catalogus > 

Define LibPub *Var = Uitdrukking*

Define LibPub *Functie(Param1, Param2, ...) =*
Uitdrukking

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

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

Werkt hetzelfde als **Define**, behalve dat er een openbare bibliotheekvariabele, -functie of -programma wordt gecreëerd. Openbare functies en programma verschijnen in de Catalogus nadat de bibliotheek is opgeslagen en vernieuwd.

Opmerking: Zie ook **Define**, pag. 49 en **Define LibPriv**, pag. 50.

deltaList()

Zie **ΔList()**, pag. 109.

deltaTmpCnv()

Zie **ΔtmpCnv()**, pag. 202.

DelVar

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

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

DelVar *Var*.

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

Wist de gespecificeerde variabele of variabelegroep uit het geheugen.

DelVar <i>a</i>	Done
$(a+2)^2$	$(a+2)^2$

Als een aantal van de variabelen vergrendeld is, veroorzaakt deze opdracht een foutmelding en worden alleen de niet-vergrendelde variabelen gewist. Zie **unlock**, pag. 210.

DelVar *Var*. wist alle leden van de variabelegroep *Var*. (zoals de statistische *stat.nn*-resultaten of variabelen die gecreëerd zijn met de **LibShortcut()**-functie). De punt (.) in deze vorm van het commando **DelVar** beperkt dit tot het wissen van een variabelegroep; de enkelvoudige variabele *Var* wordt niet gewist.

<i>aa.a:=45</i>	45									
<i>aa.b:=5.67</i>	5.67									
<i>aa.c:=78.9</i>	78.9									
getVarInfo()	<table border="1"> <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()Catalogus > **delVoid(Lijst l) ⇒ lijst** $\text{delVoid}(\{1, \text{void}, 3\}) \quad \{1, 3\}$

Geeft een lijst met de inhoud van *Lijst l* waaruit alle lege elementen verwijderd zijn.

Zie voor meer informatie over lege elementen pag. 267.

derivative()Zie *d()*, pag. 235.**deSolve()**Catalogus > 

deSolve(1steOf2deOrdeGDV, Var, afhankelijkeVar) ⇒ een algemene oplossing

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

Geeft een vergelijking die expliciet of impliciet een algemene oplossing specificeert voor de gewone differentiaalvergelijking van de 1ste of 2de orde (GDV). In de GDV:

- Gebruik een enkel aanhalingsteken (druk op ) om de 1ste afgeleide van de afhankelijke variabele ten opzichte van de onafhankelijke variabele aan te duiden.
- Gebruik twee accenten om de overeenkomstige tweede afgeleide aan te duiden.

Het accent wordt alleen gebruikt voor afgeleiden binnen deSolve(). Gebruik in andere gevallen **d()**.

De algemene oplossing van een vergelijking van de 1ste orde bevat een willekeurige constante met de vorm *ck*, waarbij *k* een geheel getal-suffix van 1 tot en met 255 is. De oplossing van een vergelijking van de 2de orde bevat twee van zulke constanten.

Pas **solve()** toe op een impliciete oplossing als u wilt proberen om deze te converteren naar één of meer equivalente expliciete oplossingen.

$$\begin{aligned} &\text{deSolve}(y' = (\cos(y))^2 \cdot x, x, y) \quad \tan(y) = \frac{x^2}{2} + c4 \end{aligned}$$

Wanneer u uw resultaten vergelijkt met oplossingen uit het boek of met handmatige oplossingen, let er dan op dat verschillende methodes willekeurige constanten op verschillende plaatsen in de berekening invoeren, wat verschillende algemene oplossingen kan opleveren.

deSolve(IsteOrdeGDV and beginVoorw, Var, afhankelijkeVar) $\Rightarrow$ een particuliere oplossing

Geeft een particuliere oplossing die voldoet aan *IsteOrdeGDV* en *beginVoorw*. Dit is gewoonlijk makkelijker dan een algemene oplossing bepalen, beginwaarden substitueren, oplossen naar de arbitraire constante en die waarde vervolgens substitueren in de algemene oplossing.

beginVoorw is een vergelijking van de vorm:

afhankelijkeVar
(*BeginOnafhankelijkeWaarde*) =
BeginAfhankelijkeWaarde

De *OnafhankelijkeBeginWaarde* en *AfhankelijkeBeginWaarde* kunnen variabelen zoals *x0* en *y0* zijn, die geen opgeslagen waarden hebben. Impliciete differentiatie kan helpen bij het verifiëren van impliciete oplossingen.

deSolve

(*2deOrdeGDV*
and*beginVoorw1***and***beginVoorw2, Var,*
afhankelijkeVar) $\Rightarrow$ een particuliere oplossing

Geeft een particuliere oplossing die voldoet aan *2deOrde GDV* en een gespecificeerde waarde heeft van de afhankelijke variabele en zijn eerste afgeleide op één punt.

Gebruik voor *beginVoorw1* de vorm:

afhankelijkeVar
(*OnafhankelijkeBeginWaarde*) =
AfhankelijkeBeginWaarde

$$\text{solve}(\text{Ans}, y) \quad y = \tan^{-1}\left(\frac{x^2 + 2 \cdot c4}{2}\right) + n3 \cdot \pi$$

$$\text{Ans} | c4 = c - 1 \text{ and } n3 = 0 \quad y = \tan^{-1}\left(\frac{x^2 + 2 \cdot (c - 1)}{2}\right)$$

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

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

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

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

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

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

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

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

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

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

$$y = \frac{\frac{1}{3} \cdot \frac{2}{3} \cdot \frac{2}{3} \cdot t^{\frac{3}{2}}}{4} \text{ and } t \geq 0$$

Gebruik voor *beginVoorw2* de vorm:

afhankelijkeVar

(OnafhankelijkeBeginWaarde) =

1steAfgeleideBeginWaarde

deSolve

(
2deOrdeGDV

andgrensVoorw1andgrensVoorw2, Var,

afhankelijkeVar)⇒*een particuliere oplossing*

Geeft een particuliere oplossing die voldoet aan *2deOrdeGDV* en gespecificeerde waarden heeft op twee verschillende punten.

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

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

deSolve(*y''=x* and *y(0)=1* and *y'(2)=3,x,y*)

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

deSolve(*y''=2 \cdot y'* and *y(3)=1* and *y'(4)=2,x,y*)

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

det()

det(*vierkanteMatrix[, Tolerantie]*)

⇒*uitdrukking*

Geeft de determinant van *vierkanteMatrix*.

Optioneel wordt elk matrixelement behandeld als nul als de absolute waarde ervan minder dan *Tolerantie* is. Deze tolerantie wordt alleen gebruikt als de matrix gegevens met een drijvende komma heeft, en geen symbolische variabelen bevat die geen waarde toegekend hebben gekregen. Anders wordt *Tolerantie* genegeerd.

- Als u   gebruikt of de modus **Automatisch of Benaderend** instelt op Benaderend, dan worden berekeningen met behulp van de drijvende komma uitgevoerd.
- Als *Tolerantie* wordt weggelaten of niet

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) \quad a \cdot d - b \cdot c$$

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

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

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

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

$$\det(\text{mat1}, 1) \quad 1. \text{E}20$$

det()Catalogus > 

wordt gebruikt, dan wordt de standaardtolerantie berekend als:

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

diag()Catalogus > 

diag(Lijst) ⇒ matrix

$\text{diag}([2 \ 4 \ 6])$

2	0	0
0	4	0
0	0	6

diag(rijMatrix) ⇒ matrix

diag(kolomMatrix) ⇒ matrix

Geeft een matrix met de waarden in de argumentenlijst of de argumentenmatrix op zijn hoofddiagonaal.

diag(vierkanteMatrix) ⇒ rijMatrix

Geeft een rijmatrix met de elementen uit de hoofddiagonaal van *vierkanteMatrix*.

vierkanteMatrix moet vierkant zijn.

4	6	8
1	2	3
5	7	9

 $\text{diag}(\text{Ans})$

4	6	8
1	2	3
5	7	9

dim()Catalogus > 

dim(Lijst) ⇒ geheel getal

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

Geeft de afmeting van *Lijst*.

dim(Matrix) ⇒ lijst

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

Geeft de afmetingen van matrix als een lijst met twee elementen {rijen, kolommen}.

dim(String) ⇒ geheel getal

$\text{dim}(\text{"Hello"})$ 5

Geeft het aantal tekens in de tekenreeks *String*.

$\text{dim}(\text{"Hello "&"there"})$ 11

Disp *uitdrOfString1* [, *uitdrOfString2*] ...

Geeft de argumenten in de geschiedenis van de *Rekenmachine*. De argumenten worden achter elkaar weergegeven, met smalle spaties als scheiding.

Vooral handig in programma's en functies om de weergave van tussenberekeningen te verzekeren.

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

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

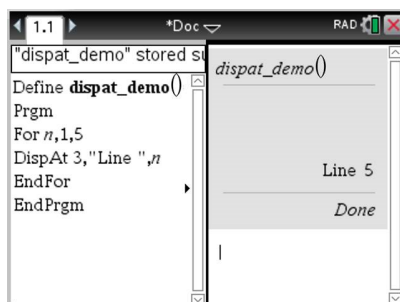
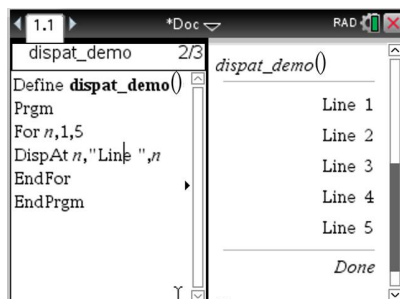
DispAt stelt u in staat de regel op te geven waarin de gespecificeerde uitdrukking of string op het scherm zal worden weergegeven.

Het regelnummer kan worden opgegeven als een uitdrukking.

Let op: het regelnummer geldt niet voor het gehele scherm, maar voor het gedeelte dat direct volgt op de instructie /het programma.

Deze instructie maakt dashboard-achtige uitvoer mogelijk van programma's waarbij de waarde van een uitdrukking of van een sensor-uitlezing op dezelfde regel wordt bijgewerkt.

DispAt en **Disp** kunnen binnen hetzelfde programma worden gebruikt.

DispAt**Voorbeeld**

Opmerking: Het maximale aantal is ingesteld op 8, aangezien dit overeenkomt met een scherm vol met regels op het rekenmachinescherm - zo lang de regels maar geen 2D wiskundige uitdrukkingen bevatten. Het exacte aantal regels is afhankelijk van de inhoud van de weergegeven informatie.

Illustratieve voorbeelden:

Define z()= Prgm For n,1,3 DispAt 1,"N: ",n Disp "Hallo" EndFor EndPrgm	Uitvoer z() Iteratie 1: Regel 1: N:1 Regel 2: Hallo Iteratie 2: Regel 1: N:2 Regel 2: Hallo Regel 3: Hallo Iteratie 3: Regel 1: N:3 Regel 2: Hallo Regel 3: Hallo Regel 4: Hallo
Define z1()= Prgm For n,1,3 DispAt 1,"N: ",n EndFor For n,1,4 Disp "Hallo" EndFor EndPrgm	z1() Regel 1: N:3 Regel 2: Hallo Regel 3: Hallo Regel 4: Hallo Regel 5: Hallo

Foutmeldingen:

Foutbericht	Beschrijving
Het DispAt regelnummer moet tussen de 1 en 8 liggen	Uitdrukking beoordeelt het Regelnummer buiten het bereik van 1-8 (inclusief 8)
Te weinig argumenten	In de functie of instructie ontbreken één of meer argumenten.
Geen argumenten	Hetzelfde als huidige 'syntax error' dialoog
Te veel argumenten	Beperk argument. Dezelfde error als

Foutbericht	Beschrijving
	Disp.
Ongeldig gegevenstype	Eerste argument moet een getal zijn.
Leeg: DispAt leeg	De "Hello World"-gegevenstypefout wordt gegeven voor "void" (indien 'terugbellen' is gedefinieerd)
Conversie-operator: DispAt 2_ft @> _m, "Hallo wereld"	CAS: Gegevenstypefout wordt gegeven (indien 'terugbellen' is gedefinieerd) Numeriek: Conversie zal uitgewerkt worden en indien het resultaat een geldig argument is, zal DispAt de string afdrukken op de resultatenregel.

►DMS Catalogus >

Uitdr ►DMS

In de hoekmodus Graden:

Lijst ►DMS

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

Matrix ►DMS

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>DMS in te typen.

Interpreteert het argument als een hoek en geeft de equivalente DMS (DDDDDD°MM'SS.ss")-waarde weer. Zie °, ', " (pag. 244) voor de DMS (graden, minuten, seconden)-notatie.

Opmerking: ►DMS converteert van radialen naar graden als hij wordt gebruikt in de radialenmodus. Als de invoer gevolgd wordt door een gradensymbool °, treedt er geen conversie op. U kunt ►DMS alleen op het eind van een invoerregel gebruiken.

domain() (domein)Catalogus > **domain(Uitdr1, Var)** ⇒ uitdrukking

Geeft het domein van *Uitdr1* ten opzichte van variabele *Var*.

domain() kan worden gebruikt om domeinen van functies te onderzoeken. Het is beperkt tot reële en eindige domeinen.

Deze functionaliteit heeft beperkingen vanwege tekortkomingen van de computeralgebra-algoritmes voor vereenvoudigen en oplossen

Bepaalde functies kunnen niet worden gebruikt als argumenten voor **domain()** ongeacht of ze expliciet verschijnen of binnen door de gebruiker gedefinieerde variabelen en functies. In het volgende voorbeeld kan de uitdrukking niet worden vereenvoudigd omdat $f()$ een niet-toegestane functie is.

$$\text{domain}\left(\begin{bmatrix} x \\ \frac{1}{t} \, dt, x \\ 1 \end{bmatrix}\right) \rightarrow \text{domain}\left(\begin{bmatrix} x \\ \frac{1}{t} \, dt, x \\ 1 \end{bmatrix}\right)$$

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

dominantTerm()Catalogus > 

dominantTerm(Uitdr1, Var [, Punt])
⇒ uitdrukking

dominantTerm(Uitdr1, Var [, Punt]) | Var > Punt ⇒ uitdrukking

dominantTerm(Uitdr1, Var [, Punt]) | Var < Punt ⇒ uitdrukking

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

Geeft de dominante term van een machtreeks-representatie van *Uitdr1* uitgewerkt rond *Punt*. De dominante term is de term die het snelst groeit in de buurt van $Var = Punt$. De resulterende macht van $(Var - Punt)$ kan een negatieve en/of gebroken exponent hebben. De coëfficiënt van deze macht kan logaritmen bevatten van $(Var - Punt)$ en andere functies van *Var* die gedomineerd worden door alle machten van $(Var - Punt)$ met hetzelfde teken in de exponent.

Punt heeft als standaardwaarde 0. *Punt* kan ∞ of $-\infty$ zijn; in die gevallen is de dominante term de term met de grootste exponent van *Var* in plaats van de kleinste exponent van *Var*.

dominantTerm(...) geeft "**dominantTerm(...)**" als hij niet in staat is om een dergelijke representatie te bepalen, zoals bij essentiële singulariteiten zoals $\sin(1/z)$ bij $z=0$, $e^{-1/z}$ bij $z=0$ of e^z bij $z = \infty$ or $-\infty$.

Als de reeks of één van zijn afgeleiden een sprong-discontinuïteit bij *Punt* heeft, dan bevat het resultaat waarschijnlijk sub-uitdrukkingen van de vorm $\text{sign}(\dots)$ of $\text{abs}(\dots)$ voor een reële uitbreidingsvariabele of $(-1)^{\text{floor}(\dots \text{angle}(\dots) \dots)}$ voor een complexe uitbreidingsvariabele, dit is een variabele die eindigt op $_$ ". Als u de dominante term alleen voor waarden aan één zijde van *Punt* wilt gebruiken, voeg dan aan **dominantTerm(...)** de juiste specificatie toe: " $| Var > Punt$ ", " $| Var < Punt$ ", " $| Var \geq Punt$ " of " $Var \leq Punt$ " om een eenvoudiger resultaat te krijgen.

dominantTerm() is distributief over 1ste-argument-lijsten en matrices.

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

dominantTerm() is handig als u de eenvoudigst mogelijke uitdrukking wilt weten die asymptotisch is naar een andere uitdrukking als $Var \rightarrow Punt$. **dominantTerm()** is ook handig als het niet duidelijk is wat de graad van de eerste niet-nul-term van een reeks zal zijn, en u niet iteratief wilt gokken, hetzij interactief of via een programmalus.

Opmerking: Zie ook **reeksen()**, pag. 171.

dotP()

dotP(Lijst1, Lijst2)⇒uitdrukking

Geeft het inwendige product van twee lijsten.

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

dotP(Vector1, Vector2)⇒uitdrukking

Geeft het inwendige product van twee vectoren.

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

Beide moeten rijvectoren zijn, of beide moeten kolomvectoren zijn.

E

e^()

e^(Uitdr1)⇒uitdrukking

Geeft **e** tot de macht *Uitdr1*.

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

Opmerking: zie ook **e macht-template**, pag. 2.

Opmerking: op  drukken om $e^()$ weer te geven is niet hetzelfde als drukken op het teken  op het toetsenbord.

U kunt een complex getal in $re^{i\theta}$ polaire vorm invoeren. Gebruik deze vorm echter alleen in de hoekmodus Radialen; hij veroorzaakt een domeinfout in de hoekmodi Graden en Decimale graden.

e^(Lijst1)⇒lijst

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

Geeft **e** tot de macht van elk element in *Lijst1*.

e^(*vierkanteMatrix1*)⇒*vierkanteMatrix*

Geeft de 'e tot de macht van *vierkanteMatrix1*'. Dit is niet hetzelfde als het berekenen van e tot de macht van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

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

eff(*nominaalPercentage*, *CpY*)⇒*waarde*

eff(5.75,12)

5.90398

Financiële functie die het nominale rentepercentage *nominaalPercentage* converteert naar een jaarlijks effectief percentage, waarbij *CpY* het aantal samengestelde periodes per jaar is.

nominaalPercentage moet een reëel getal zijn, en *CpY* moet een reëel getal > 0 zijn.

Opmerking: zie ook **nom()**, pag. 130.

eigVc(*vierkanteMatrix*)⇒*matrix*

In rechthoekige complexe opmaak:

Geeft een matrix met de eigenvectoren voor een reële of complexe *vierkanteMatrix*, waarbij elke kolom in het resultaat overeenkomt met een eigenwaarde. Merk op dat een eigenvector niet uniek is; hij kan geschaald worden door een willekeurige constante factor. De eigenvectoren worden genormaliseerd, wat betekent: als $V = [x_1, x_2, \dots, x_n]$, dan:

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

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

eigVc(m1)

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

Om het hele resultaat te zien drukt u op **▲** en gebruikt u vervolgens **◀** en **▶** om de cursor te verplaatsen.

vierkanteMatrix wordt eerst gebalanceerd met gelijkheidstransformaties tot de rij- en kolomnormen zo dicht mogelijk bij dezelfde waarde liggen. *vierkanteMatrix* wordt vervolgens gereduceerd tot de upper-Hessenberg-vorm en de eigenvectoren worden berekend via een Schur-factorisatie.

eigVl()

eigVl(*vierkanteMatrix*)⇒lijst

Geeft een lijst van de eigenwaarden van een reële of complexe *vierkanteMatrix*.

vierkanteMatrix wordt eerst gebalanceerd met gelijkheidstransformaties tot de rij- en kolomnormen zo dicht mogelijk bij dezelfde waarde liggen. *vierkanteMatrix* wordt vervolgens gereduceerd tot de upper-Hessenberg-vorm en de eigenwaarden worden berekend uit de upper-Hessenberg-matrix.

In de rechthoekige complexe opmaak-modus:

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

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

Om het hele resultaat te zien drukt u op  en gebruikt u vervolgens  en  om de cursor te verplaatsen.

Else

Zie If, pag. 92.

ElseIf

Als *BooleaanseUitdr1* **Then**
 Blok1

ElseIf *BooleaanseUitdr2* **Then**
 Blok2

⋮

ElseIf *BooleaanseUitdrN* **Then**
 BlokN

EndIf

⋮

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Define $g(x)$ = Func

If $x \leq -5$ Then

Return 5

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

Return $-x$

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

Return x

ElseIf $x = 10$ Then

Return 3

EndIf

EndFunc

Done

EndFor**Zie For, pag. 77.****EndFunc****Zie Func, pag. 81.****EndIf****Zie If, pag. 92.****EndLoop****Zie Loop, pag. 116.****EndPrgm****Zie Prgm, pag. 145.****EndTry****Zie Try, pag. 203.****EndWhile****Zie While, pag. 214.****euler ()****Catalogus > **

euler(*Uitdr*, *Var*, *afhVar*, {*Var0*, *VarMax*},
afhVar0, *VarStap* [, *eulerStap*]) $\Rightarrow$ *matrix*

euler(*StelselUitdr*, *Var*, *LijstVanAfhVars*,
{*Var0*, *VarMax*}, *LijstVanAfhVars0*,
VarStap [, *eulerStap*]) $\Rightarrow$ *matrix*

euler(*LijstVanUitdr*, *Var*,
LijstVanAfhVars, {*Var0*, *VarMax*},
LijstVanAfhVars0, *VarStap* [, *eulerStap*])
 $\Rightarrow$ *matrix*

Gebruikt de Euler-methode om het stelsel

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

Differentiaalvergelijking:

$$y' = 0,001 \cdot y \cdot (100 - y) \text{ en } y(0) = 10$$

euler(0.001·y·(100-y), t, y, {0, 100}, 10, 1)					
0.	1.	2.	3.	4.	
10.	10.9	11.8712	12.9174	14.042	

Om het volledige resultaat te zien drukt u op  en gebruikt u vervolgens  en  om de cursor te verplaatsen.

op te lossen met $afhVar(Var0)=afhVar0$ op het interval $[Var0, VarMax]$. Geeft een matrix waarvan de eerste rij de Var -uitvoerwaarden definieert, en de tweede rij de waarde van de eerste oplossingscomponent bij de overeenkomstige Var -waarden definieert, enzovoort.

Uitdr is de rechterzijde die de gewone differentiaalvergelijking (GDV) definieert.

StelselUitdr is het stelsel van de rechterleden die het stelsel van GDV's definiëren (komt overeen met de volgorde van afhankelijke variabelen in *LijstVanAfhVars*).

StelselUitdr is een lijst van de rechterleden die het stelsel van GDV's definiëren (komt overeen met de volgorde van afhankelijke variabelen in *LijstVanAfhVars*).

Var is de onafhankelijke variable.

LijstVanAfhVars is een lijst van afhankelijke variabelen.

$\{Var0, VarMax\}$ is een lijst met twee elementen die de functie vertelt om van $Var0$ tot $VarMax$ te integreren.

LijstVanAfhVars0 is een lijst met beginwaarden voor afhankelijke variabelen.

VarStap is een getal dat niet nul is, zodanig dat $\text{sign}(VarStap) = \text{sign}(VarMax-Var0)$ en er oplossingen worden gegeven bij $Var0+i \cdot VarStap$ voor alle $i=0,1,2,\dots$ zodanig dat $Var0+i \cdot VarStap$ binnen $[var0, VarMax]$ valt (mogelijk is er geen oplossingswaarde bij $VarMax$).

eulerStap is een positief geheel getal (standaardwaarde is 1) dat het aantal eulerstappen tussen uitvoerwaarden definieert. De feitelijke stapgrootte die gebruikt wordt door de eulermethode is $VarStap/eulerStap$.

Vergelijk bovengenoemde resultaten met de exacte CAS-oplossing die verkregen is met behulp van $deSolve()$ en $seqGen()$:

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

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

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

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

Stelsel vergelijkingen:

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

met $y1(0)=2$ en $y2(0)=5$

$$euler \left(\begin{cases} y1 + 0.1 \cdot y1 \cdot y2 \\ 3 \cdot y2 - y1 \cdot y2 \end{cases}, t, \{y1, y2\}, \{0.5\}, \{2.5\}, 1 \right)$$

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

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

eval() is alleen geldig in het TI-Innovator™ Hub argument van de programmeeropdrachten **Get**, **GetStr**, en **Send**. De software werkt de uitdrukking *Expr* uit en vervangt **eval()** door het resultaat daarvan, in de vorm van een tekenreeks (string).

Het argument *Expr* moet vereenvoudigd kunnen worden tot een reëel getal.

Stel het blauwe deel van de RGB LED in op halve intensiteit.

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

Zet het blauwe element terug op OFF (UIT).

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

Het argument van **eval()** moet vereenvoudigd kunnen worden tot een reëel getal.

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

Programma om het rode element in te faden

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

Voer het programma uit.

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

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

Hoewel **eval()** het resultaat niet weergeeft, kunt u de Hub opdracht string die het resultaat is na het uitvoeren van de opdracht bekijken door een van de volgende speciale variabelen te onderzoeken.

iostr.SendAns
iostr.GetAns
iostr.GetStrAns

Opmerking: Zie ook **Get** (pag. 82), **GetStr** (pag. 89), en **Send** (pag. 168).

exact()		Catalogus > 
exact (<i>Uitdr1</i> [, <i>Tolerantie</i>])⇒ <i>uitdrukking</i>	$\text{exact}(0.25)$	$\frac{1}{4}$
exact (<i>Lijst1</i> [, <i>Tolerantie</i>])⇒ <i>lijst</i>	$\text{exact}(0.333333)$	$\frac{333333}{1000000}$
exact (<i>Matrix1</i> [, <i>Tolerantie</i>])⇒ <i>matrix</i>	$\text{exact}(0.333333, 0.001)$	$\frac{1}{3}$
Gebruik de Exact-modus om, waar mogelijk, het rationaal-getal-equivalent van het argument te geven.		
<i>Tolerantie</i> specificeert de tolerantie voor de conversie; de standaardwaarde is 0 (nul).		
	$\text{exact}(3.5 \cdot x + y)$	$\frac{7 \cdot x}{2} + y$
	$\text{exact}(\{0.2, 0.33, 4.125\})$	$\left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$

Exit		Catalogus > 
Exit	Functielijst:	
Sluit het huidige For , While of Loop -blok af.	Define $g()$ = Func <i>Done</i>	
Exit is niet toegestaan buiten de drie lusstructuren (For , While of Loop).	Local <i>temp</i> , <i>i</i>	
	$0 \rightarrow \text{temp}$	
	For <i>i</i> , 1, 100, 1	
	$\text{temp} + i \rightarrow \text{temp}$	
	If $\text{temp} > 20$ Then	
	Exit	
	EndIf	
	EndFor	
	EndFunc	
	$g()$	21

►exp		Catalogus > 
Uitdr ►exp	$\frac{d}{dx} (e^x + e^{-x})$	$2 \cdot \sinh(x)$
Geeft <i>Uitdr</i> weer uitgedrukt in termen van het getal <i>e</i> . Dit is een operator voor weergaveconversie. Deze kan alleen op het eind van de invoerregel gebruikt worden.	$2 \cdot \sinh(x)$ ►exp	$e^x - e^{-x}$
Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @►exp in te typen.		

exp()**e^x-toets****exp(Uitdr1)**⇒uitdrukking

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

Geeft **e** tot de macht *Uitdr1*.

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

Opmerking: zie ook **e** exponent-template, pag. 2.

e^{3^2}	e^9
-----------	-------

U kunt een complex getal in $re^{i\theta}$ polaire vorm invoeren. Gebruik deze vorm echter alleen in de hoekmodus Radialen; hij veroorzaakt een domeinfout in de hoekmodi Graden en Decimale graden.

exp(Lijst1)⇒lijst

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

Geeft **e** tot de macht van elk element in *Lijst1*.**exp(vierkanteMatrix1)**⇒vierkanteMatrix

Geeft de 'e tot de macht van *vierkanteMatrix1*'. Dit is niet hetzelfde als het berekenen van **e** tot de macht van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

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

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

exp►list()**Catalogus > ****exp►list(Uitdr,Var)**⇒lijst

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

Onderzoekt *Uitdr* voor vergelijkingen die gescheiden worden door het woord "or," en geeft een lijst met elk rechterlid van de vergelijkingen van de vorm $Var=Uitdr$. Dit geeft u een makkelijke manier om bepaalde oplossingswaarden die ingebed zijn in de resultaten van de functies **solve()**, **cSolve()**, **fMin()** en **fMax()** te extraheren.

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

Opmerking: **exp►list()** is niet noodzakelijk bij de functies **zeros** en **cZeros()**, omdat deze rechtstreeks een lijst met oplossingswaarden teruggeven.

u kunt deze operator vanaf het toetsenbord van de computer invoeren door

exp@>list(...) in te typen.

expand(Uitdr1 [, Var]) ⇒ uitdrukking

expand(Lijst1 [, Var]) ⇒ lijst

expand(Matrix1 [, Var]) ⇒ matrix

expand(Uitdr1) geeft *Uitdr1* uitgewerkt ten opzichte van al zijn variabelen. De uitwerking is een veeltermuitwerking bij veeltermen en een partiële breukuitwerking bij rationale uitdrukkingen.

Het doel van **expand()** is het transformeren van *Uitdr1* in een som en/of verschil van eenvoudige termen. Daarentegen is het doel van **factor()** het transformeren van *Uitdr1* in een product en/of quotiënt van eenvoudige factoren.

expand(Uitdr1, Var) geeft *Uitdr1* uitgewerkt ten opzichte van *Var*. Gelijke machten van *Var* worden samengenomen. De termen en hun factoren worden gesorteerd met *Var* als de hoofdvariabele. Er kan wat onbedoelde ontbinding of uitwerking ten opzichte van de verzamelde coëfficiënten optreden. Vergeleken met het weglaten van *Var* scheelt dit vaak tijd, geheugen en schermruimte, terwijl de uitdrukking begrijpelijker wordt.

Ook als er maar één variabele is, kan het gebruik van *Var* de noemerontbinding die gebruikt wordt bij partiële breukuitwerking completer maken.

Tip: bij rationale uitdrukkingen is **propFrac()** een sneller maar minder extreem alternatief voor **expand()**.

Opmerking: zie ook **comDenom()** voor een uitgewerkte teller boven een uitgewerkte noemer.

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

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

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

expand()

Catalogus >

expand(Uitdr1, [Var]) past ook distributiviteit toe bij logaritmes en machten met gebroken exponenten, ongeacht *Var*. Voor een verdere toepassing van distributiviteit bij logaritmes en machten met gebroken exponenten kunnen ongelijkheidsbeperkingen noodzakelijk zijn om te garanderen dat sommige factoren niet-negatief zijn.

expand(Uitdr1, [Var]) past ook distributiviteit toe bij absolute waarden, **sign()** en exponenten, ongeacht *Var*.

Opmerking: zie ook **tExpand()** voor goniometrische hoek-som-uitwerking en uitwerking van meerdere hoeken.

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

expr()

Catalogus >

expr(String)⇒uitdrukking

Geeft de tekenreeks in *String* als een uitdrukking en voert deze onmiddellijk uit.

$\text{expr}("1+2+x^2+x")$	x^2+x+3
$\text{expr}("\text{expand}((1+x)^2)")$	$x^2+2 \cdot x+1$
"Define cube(x)=x^3" → <i>funcstr</i>	"Define cube(x)=x^3"
$\text{expr}(\text{funcstr})$	<i>Done</i>
$\text{cube}(2)$	8

ExpReg

Catalogus >

ExpReg *X*, *Y* [, [*Freq*][, [*Categorie*, *Opnemen*]]

Berekent de exponentiële regressie $y = a \cdot (b)^x$ op de lijsten *X* en *Y* met frequentie *Freq*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en *Y* zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig *X*- en *Y*-gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige *X*- en *Y*-gegevens.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a \cdot (b)^x$
stat.a, stat.b	Regressiecoëfficiënten
stat.r ²	Coëfficiënt van lineaire verband voor getransformeerde gegevens
stat.r	Correlatiecoëfficiënt voor getransformeerde gegevens (x , $\ln(y)$)
stat.Resid	Residuen die geassocieerd zijn met het exponentiële model
stat.ResidTrans	Residuen die geassocieerd zijn met de lineaire regressie van getransformeerde gegevens
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

F

factor()

factor(Uitdr1[, Var]) ⇒ uitdrukking

factor(Lijst1[, Var]) ⇒ lijst

factor(Matrix1[, Var]) ⇒ matrix

factor(Uitdr1) geeft *Uitdr1* ontbonden ten opzichte van al zijn variabelen over een gemeenschappelijke noemer.

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

Uitdr1 wordt zoveel mogelijk ontbonden naar lineaire rationale factoren, zonder nieuwe niet-reële uitdrukkingen te introduceren. Dit alternatief is geschikt als u ontbinding in factoren ten opzichte van meer dan één variabele wilt.

`factor(Uitdr1,Var)` geeft *Uitdr1* ontbonden ten opzichte van de variabele *Var*.

Uitdr1 wordt zoveel mogelijk ontbonden in factoren die lineair zijn in *Var*, zelfs als dit irrationale constanten of subuitdrukkingen die irrationaal zijn in andere variabelen oplevert.

De factoren en hun termen worden gesorteerd met *Var* als de hoofdvariabele. In elke factor worden gelijksoortige machten van *Var* samengenomen. Neem *Var* op als argument als u een ontbinding ten opzichte van alleen die variabele nodig heeft, en u bereid bent om irrationale uitdrukkingen in andere variabelen te accepteren om een ontbinding ten opzichte van *Var* te vergroten. Er kan wat onbedoelde ontbinding ten opzichte van andere variabelen optreden.

Bij de automatische instelling van de **Automatische of Benaderende** modus kunt u door *Var* op te nemen benadering met drijvende komma-coëfficiënten mogelijk maken, waar irrationale coëfficiënten niet expliciet beknopt uitgedrukt kunnen worden in termen van de ingebouwde functies. Ook als er maar één variabele is, kan het opnemen van *Var* een vollediger ontbinding opleveren.

Opmerking: zie ook `comDenom()` voor een snelle manier om partiële ontbinding te bereiken als `factor()` niet snel genoeg is, of het geheugen te veel belast.

Opmerking: zie ook `cFactor()` om volledig te ontbinden naar complexe coëfficiënten op zoek naar lineaire factoren.

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

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

factor(*rationaalGetal*) geeft het rationale getal ontbonden in priemfactoren. Bij samengestelde getallen neemt de berekeningstijd exponentieel toe met het aantal cijfers in de op één na grootste factor. Het ontbinden van een geheel getal van 30 cijfers kan bijvoorbeeld langer dan een dag duren, en het ontbinden van een geheel getal van 100 cijfers kan meer dan een eeuw duren.

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

Een berekening handmatig stoppen:

- **Rekenmachine:** Houd de toets  ingedrukt en druk enkele malen op .
- **Windows®:** Houd **F12** ingedrukt en druk enkele malen op **Enter**.
- **Macintosh®:** Houd **F5** ingedrukt en druk enkele malen op **Enter**.
- **iPad®:** De app toont een instructie. U kunt blijven wachten of annuleren.

Als u alleen wilt bepalen of een getal een priemgetal is, gebruik dan liever **isPrime()**. Dat is veel sneller, vooral als *rationaalGetal* geen priemgetal is en als de op één na grootste factor meer dan vijf cijfers heeft.

FCdf(*ondergrens*,*bovengrens*,*dfTeller*,*dfNoemer*)
 $\Rightarrow$ *getal* als *ondergrens* en *bovengrens* getallen zijn,
lijst als *ondergrens* en *bovengrens* lijsten zijn

FCdf(*ondergrens*,*bovengrens*,*dfTeller*,*dfNoemer*)
 $\Rightarrow$ *getal* als *ondergrens* en *bovengrens* getallen zijn,
lijst als *ondergrens* en *bovengrens* lijsten zijn

Berekent de F-kansverdeling tussen *ondergrens* en *bovengrens* voor de gespecificeerde *dfTeller* (aantal vrijheidsgraden) en *dfNoemer*.

Bij $P(X \leq \textit{bovengrens})$ is de ingestelde *ondergrens* = 0.

Fill *Uitdr*, *matrixVar* ⇒ *matrix*

Vervangt ieder element in variabele *matrixVar* door *Uitdr*.

matrixVar moet al bestaan.

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

Fill *Uitdr*, *lijstVar* ⇒ *lijst*

Vervangt ieder element in variabele *lijstVar* door *Uitdr*.

lijstVar moet al bestaan.

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

FiveNumSummary**FiveNumSummary** *X*[,*Freq*][,*Categorie*,*Opnemen*]

Levert een verkorte versie van de statistieken voor 1 variabele van lijst *X*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

X representeert een lijst met de gegevens.

Freq is een optionele lijst met frequentiewaarden.

Elk element in *Freq* specificeert de frequentie waarmee elke overeenkomstige *X*-waarde voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met numerieke categoriecodes voor de overeenkomstige *X*-waarden.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Een leeg element in een van de lijsten *X*, *Freq* of *Categorie* resulteert in een lege plaats voor het overeenkomstige element in al deze lijsten. Voor meer informatie over lege elementen, zie pag. 267.

Uitvoervariabele	Beschrijving
stat.MinX	Minimum van de x-waarden
stat.Q ₁ X	1ste kwartiel van x
stat.MedianX	Mediaan van x

Uitvoervariabele	Beschrijving
stat.Q ₃ X	3de kwartiel van x
stat.MaxX	Maximum van de x-waarden

floor()

Catalogus > 

floor(Uitdr1)⇒geheel getal

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

Geeft het grootste gehele getal dat $\leq$ dan het argument. Deze functie is hetzelfde als **int()**.

Het argument kan een reëel of complex getal zijn.

floor(Lijst1)⇒lijst

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

floor(Matrix1)⇒matrix

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

Geeft een lijst of matrix van de floor-waarde van elk element.

Opmerking: zie ook **ceiling()** en **int()**.

fMax()

Catalogus > 

fMax(Uitdr, Var)⇒Booleaanse uitdrukking

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

fMax(Uitdr, Var, ondergrens)

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

fMax(Uitdr, Var, ondergrens, bovengrens)

fMax(Uitdr, Var) | ondergrens ≤ Var ≤ bovengrens

Geeft een Booleaanse uitdrukking die mogelijke waarden specificeert van *Var* die *Uitdr* maximaliseren of zijn kleinste bovengrens lokaliseren.

U kunt de ("|")-operator gebruiken om het oplossingsinterval te beperken en/of andere beperkingen te specificeren.

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

Bij de instelling Benaderend van de **Automatische of Benaderende** modus zoekt **fMax()** iteratief naar één benaderend lokaal maximum. Dit is vaak sneller, vooral als u de “|”-operator gebruikt om de zoekactie te beperken tot een relatief klein interval dat exact één lokaal maximum bevat.

Opmerking: zie ook **fMin()** en **max()**.

fMin(Uitdr, Var)⇒Booleaanse uitdrukking

fMin(Uitdr, Var, ondergrens)

fMin(Uitdr, Var, ondergrens, bovengrens)

fMin(Uitdr, Var) | ondergrens ≤ Var ≤ bovengrens

Geeft een Booleaanse uitdrukking die mogelijke waarden specificeert van *Var* die *Uitdr* minimaliseren of zijn grootste ondergrens lokaliseren.

U kunt de (“|”)-operator gebruiken om het oplossingsinterval te beperken en/of andere beperkingen te specificeren.

Bij de instelling Benaderend van de **Automatische of Benaderende** modus zoekt **fMin()** iteratief naar één benaderend lokaal minimum. Dit is vaak sneller, vooral als u de “|”-operator gebruikt om de zoekactie te beperken tot een relatief klein interval dat exact één lokaal minimum bevat.

Opmerking: zie ook **fMax()** en **min()**.

$fMin(1 - (x-a)^2 - (x-b)^2, x)$	$x = -\infty$ of $x = \infty$
$fMin(0.5 \cdot x^3 - x - 2, x) x \geq 1$	$x = 1$

For <i>Var</i> , <i>Laag</i> , <i>Hoog</i> [, <i>Stap</i>] <i>Blok</i> EndFor	Define $g()$ =Func Local <i>tempsum</i> , <i>step</i> , <i>i</i> 0 $\rightarrow$ <i>tempsum</i> 1 $\rightarrow$ <i>step</i> For <i>i</i> ,1,100, <i>step</i> <i>tempsum</i> + <i>i</i> $\rightarrow$ <i>tempsum</i> EndFor EndFunc	Done
Voert de beweringen in <i>Blok</i> iteratief uit voor elke waarde van <i>Var</i> , van <i>Laag</i> naar <i>Hoog</i> , in stappen van <i>Stap</i> .		
<i>Var</i> mag geen systeemvariabele zijn.		
<i>Stap</i> kan positief of negatief zijn. De standaardwaarde is 1.	$g()$	5050

Blok kan een enkele bewering of een serie beweringen zijn die gescheiden worden door het teken “:”.

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

format (<i>Uitdr</i> [, <i>opmaakString</i>]) $\Rightarrow$ <i>string</i>	format(1.234567,"f3")	"1.235"
Geeft <i>Uitdr</i> als een tekenreeks op basis van de opmaaktemplate.	format(1.234567,"s2")	"1.23E0"
<i>Uitdr</i> moet vereenvoudigd worden naar een getal.	format(1.234567,"e3")	"1.235E0"
<i>opmaakString</i> is een string die de volgende vorm moet hebben: “F[n]”, “S[n]”, “E[n]”, “G[n][c]”, waarbij [] optionele gedeeltes aangeeft.	format(1.234567,"g3")	"1.235"
F[n]: Vaste opmaak. n is het aantal cijfers dat weergegeven moet worden achter de decimale punt.	format(1234.567,"g3")	"1,234.567"
S[n]: Wetenschappelijke opmaak. n is het aantal cijfers dat weergegeven moet worden achter de decimale punt.	format(1.234567,"g3,r:")	"1:235"

E[n]: Ingenieursopmaak. *n* is het aantal cijfers na het eerste significante cijfer. De exponent wordt aangepast naar een veelvoud van drie, en de decimale punt wordt met nul, één of twee cijfers naar rechts verplaatst.

G[n][c]: Zelfde als de vaste opmaak, maar scheidt de cijfers links van de radix (decimale scheidingstekens) tevens in groepen van drie. *c* specificeert het groepscheidingstekens; de standaardinstelling is een komma. Als *c* een punt is, wordt de radix weergegeven als een komma.

[Rc]: Elk van bovengenoemde specificatietekens kan als suffix de *Rc* radix-vlag krijgen, waarbij *c* een enkel teken is dat specificeert wat er gesubstitueerd moet worden voor het radixpunt.

fPart()

fPart(Uitdr1) ⇒ uitdrukking

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

fPart(Lijst1) ⇒ lijst

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

fPart(Matrix1) ⇒ matrix

Geeft de breuk van het argument.

Geeft bij een lijst of matrix de breuk van de elementen.

Het argument kan een reëel of complex getal zijn.

FPdf()

Fpdf(XWaarde, dfTeller, dfNoemer) ⇒ getal als *XWaarde* een getal is, lijst als *XWaarde* een lijst is

Berekent de kans voor de *F*-verdeling bij *XWaarde* voor de gespecificeerde *dfTeller* (vrijheidsgraden) en *dfNoemer*.

freqTable►list

(Lijst1,freqGeheelGetalLijst)⇒lijst

Geeft een lijst met de elementen uit *Lijst1* uitgebreid volgens de frequenties in *freqGeheelGetalLijst*. Deze functie kan gebruikt worden om een frequentietabel voor de Gegevensverwerking & Statistiek-toepassing samen te stellen.

Lijst1 kan elke geldige lijst zijn.

freqGeheelGetalLijst moet dezelfde afmeting als *Lijst1* hebben en mag alleen niet-negatieve gehele getallen bevatten. Elk element specificeert het aantal keer dat het overeenkomstige element uit *Lijst1* wordt herhaald in de resulterende lijst. Een waarde van nul sluit het overeenkomstige *Lijst1*-element uit.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **freqTable@>list(...)** in te typen.

Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

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

frequency()

frequency(Lijst1,klassenLijst)⇒lijst

Geeft een lijst met de aantallen elementen in *Lijst1*. De aantallen zijn gebaseerd op klassen die u definieert in *klassenLijst*.

Als *klassenLijst* {b(1), b(2), ..., b(n)} is, dan zijn de gespecificeerde klassen { $? \leq b(1)$, $b(1) < ? \leq b(2)$, ..., $b(n-1) < ? \leq b(n)$, $b(n) > ?$ }. De resulterende lijst is één element langer dan *klassenLijst*.

Elk element van het resultaat komt overeen met het aantal elementen in *Lijst1* die binnen die klasse liggen. Uitgedrukt in termen van de **countIf()**-functie is het resultaat {countIf(list, $? \leq b(1)$), countIf(list, $b(1) < ? \leq b(2)$), ..., countIf(list, $b(n-1) < ? \leq b(n)$), countIf(list, $b(n) > ?$)}.

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

Uitleg van het resultaat:

2 elementen van *Datalist* zijn $\leq 2,5$

4 elementen van *Datalist* zijn $> 2,5$ en $\leq 4,5$

3 elementen van *Datalist* zijn $> 4,5$

Het element "hello" is een string en kan niet in een van de gedefinieerde klassen geplaatst worden.

Elementen van *Lijst1* die niet “in een klasse geplaatst kunnen worden” worden genegeerd. Lege elementen worden eveneens genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

In de toepassing Lijsten & Spreadsheet kunt u een reeks cellen op de plaats van beide argumenten gebruiken.

Opmerking: zie ook `countif()`, pag. 37.

FTest_2Samp

FTest_2Samp *Lijst1, Lijst2[, Freq1[, Freq2[, Hypoth]]]*

FTest_2Samp *Lijst1, Lijst2[, Freq1[, Freq2[, Hypoth]]]*

(Invoer van een gegevenslijst)

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

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

(Invoer van samenvattingsstatistieken)

Voert een F-toets met twee steekproeven uit. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Voor $H_1: \sigma_1 > \sigma_2$ stelt u *Hypoth* > 0 in

Voor $H_1: \sigma_1 \neq \sigma_2$ (standaardinstelling) stelt u *Hypoth* = 0 in

Voor $H_1: \sigma_1 < \sigma_2$ stelt u *Hypoth* < 0 in

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.F	Berekende $\hat{U}$ -statistiek voor de gegevensverzameling
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.dfNumer	teller vrijheidsgraden = $n_1 - 1$
stat.dfDenom	noemer vrijheidsgraden = $n_2 - 1$

Uitvoervariabele	Beschrijving
stat.sx1, stat.sx2	Steekproefstandaarddeviatie van de gegevensverzamelingen in <i>Lijst 1</i> en <i>Lijst 2</i>
stat.x1_bar stat.x2_bar	Steekproefgemiddelde van de gegevensverzamelingen in <i>Lijst 1</i> en <i>Lijst 2</i>
stat.n1, stat.n2	Grootte van de steekproeven

Func

Catalogus >

Func

Blok

EndFunc

Template voor het creëren van een door de gebruiker gedefinieerde functie.

Blok kan een enkele bewerking of een serie bewerkingen zijn die gescheiden worden door het teken ":", of een serie bewerkingen op aparte regels. De functie kan de instructie **Return** gebruiken om een specifiek resultaat te retourneren.

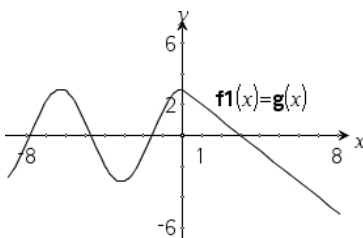
Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Een stuksgewijs gedefinieerde functie definiëren:

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

Resultaat grafiek $g(x)$



G

gcd()

Catalogus >

gcd(Waarde1, Waarde2) ⇒ uitdrukking

gcd(18,33)	3
------------	---

Geeft de grootste gemene deler van de twee argumenten. De **gcd** van twee breuken is de **gcd** van hun tellers gedeeld door de **lcm** van hun noemers.

In de Automatische of Benaderende modus is de **gcd** van breuken met een drijvende komma 1,0.

gcd()Catalogus > **gcd**(*Lijst1*, *Lijst2*)⇒*lijst* $\text{gcd}(\{12, 14, 16\}, \{9, 7, 5\}) \quad \{3, 7, 1\}$

Geeft de grootste gemene delers van de overeenkomstige elementen in *Lijst1* en *Lijst2*.

gcd(*Matrix1*, *Matrix2*)⇒*matrix* $\text{gcd}\left(\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}, \begin{bmatrix} 4 & 8 \\ 12 & 16 \end{bmatrix}\right) \quad \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$

Geeft de grootste gemene delers van de overeenkomstige elementen in *Matrix1* en *Matrix2*.

geomCdf()Catalogus > 

geomCdf(*p*, *ondergrens*, *bovengrens*)⇒*getal* als *ondergrens* en *bovengrens* getallen zijn, *lijst* als *ondergrens* en *bovengrens* lijsten zijn

geomCdf(*p*, *bovengrens*) voor $P(1 \leq X \leq \text{bovengrens})$ ⇒*getal* als *bovengrens* een getal is, *lijst* als *bovengrens* een lijst is

Berekent een cumulatieve geometrische kans van *ondergrens* naar *bovengrens* met de gespecificeerde succeskans *p*.

Voor $P(X \leq \text{bovengrens})$ stelt u *ondergrens* = 1 in.

geomPdf()Catalogus > 

geomPdf(*p*, *XWaarde*)⇒*getal* als *XWaarde* een getal is, *lijst* als *XWaarde* een lijst is

Berekent de kans op *XWaarde*, het nummer van de poging waarbij het eerste succes optreedt, voor de discrete geometrische verdeling met de gespecificeerde succeskans *p*.

Get

Hub Menu

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

Programmeeropdracht: Haalt een waarde op van een aangesloten TI-Innovator™ Hub en wijst de waarde toe aan de variabele *var*.

Voorbeeld: Vraag de huidige waarde van de ingebouwde lichtniveau-sensor van de hub aan. Gebruik **Get** om de waarde op te halen en toe te wijzen aan de variabele *lichtwaarde*.

De waarde moet zijn aangevraagd:

- Vooraf, door een **Send "READ ..."** opdracht.
— of —
- Door het inbedden van een **"READ ..."** verzoek als het optionele argument van *promptString*. Met deze methode kunt u één enkele opdracht gebruiken om de waarde aan te vragen en op te halen.

Er vindt vereenvoudiging van een impliciete vermenigvuldiging plaats. Een ontvangen tekenreeks als "123" wordt bijvoorbeeld geïnterpreteerd als een numerieke waarde. Gebruik **GetStr** in plaats van **Get** om de tekenreeks te behouden.

Als u het optionele argument *statusVar* erin opneemt, wordt er een waarde aan toegekend op basis van het succes van de bewerking. De waarde nul betekent dat er geen gegevens werden ontvangen.

In de tweede syntax, stelt het argument *func()* het programma in staat om de ontvangen tekenreeks als een functiedefinitie op te slaan. Deze syntax werkt alsof het programma de volgende opdracht heeft uitgevoerd:

Definieer *func(arg1, ...argn) =
ontvangen tekenreeks*

Het programma kan vervolgens de gedefinieerde functie *func()* gebruiken.

Opmerking: U kunt de opdracht **Get** binnen een door de gebruiker gedefinieerd programma gebruiken, maar niet binnen een functie.

Opmerking: Zie ook **GetStr**, pag. 89 en **Send**, pag. 168.

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

Neem het READ verzoek op in de opdracht **Get**.

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

getDenom(Uitdr1) ⇒ uitdrukking

Transformeert het argument in een uitdrukking met een vereenvoudigde gemeenschappelijke noemer, en geeft vervolgens de noemer ervan.

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

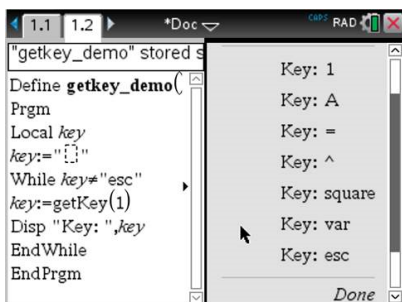
getKey()**getKey([01]) ⇒ returnString**

Beschrijving: getKey() - stelt een TI-Basic programma in staat om toetsenbord invoer op te halen - van de rekenmachine, computer en emulator op de computer.

Voorbeeld:

- keypressed := **getKey()** zal een toets teruggeven of een lege string indien er geen toets is ingedrukt. Deze opdracht zal direct resultaat hebben.
- keypressed := **getKey(1)** zal wachten totdat er een toets is ingedrukt. Deze opdracht zal de uitvoering van het programma onderbreken totdat er een toets is ingedrukt.

getKey()

Voorbeeld:**De behandeling van toetsaanslagen:**

rekenmachine/emulator toets	Desktop	Teruggeven waarde
Esc	Esc	"esc"
Touchpad - Bovenaan klikken	N.v.t.	"omhoog"
Aan	N.v.t.	"hoofdscherm"
Scratch-apps	N.v.t.	"kladblok"
Touchpad - Links klikken	N.v.t.	"links"
Touchpad - In het midden klikken	N.v.t.	"in het midden"
Touchpad - rechts klikken	N.v.t.	"rechts"

rekenmachine/emulatortoets	Desktop	Teruggeven waarde
Doc	N.v.t.	"doc"
Tabblad	Tabblad	"tabblad"
Touchpad - onderaan klikken	Pijl omlaag	"omlaag"
Menu	N.v.t.	"menu"
Ctrl	Ctrl	geen terugkeer
Shift	Shift	geen terugkeer
Var	N.v.t.	"var"
Del	N.v.t.	"del"
=	=	"="
Goniometrie	N.v.t.	"Gonio"
0 t/m 9	0-9	"0" ... "9"
Templates	N.v.t.	"sjabloon"
Catalogus	N.v.t.	"cat"
^	^	"^"
x^2	N.v.t.	"kwadraat"
/ (deeltoets)	/	"/"
* (vermenigvuldigingstoets)	*	"*"
e^x	N.v.t.	"uitdr"
10^x	N.v.t.	"10macht"
+	+	"+"
-	-	"_"
(	(	"("
)	)	")"
.	.	"."
(-)	N.v.t.	"-" (negatie- teken)
Invoeren	Invoeren	"invoeren"
ee	N.v.t.	"E" (wetenschappelijke notatie E)

rekenmachine/emulatortoets	Desktop	Teruggeven waarde
a - z	a-z	alpha = ingedrukte letter (kleine letter) ("a" - "z")
shift a-z	shift a-z	alpha = ingedrukte letter "A" - "Z"
		Let op: met ctrl-shift kunt u caps vergrendelen
?!	N.v.t.	"?!"
pi	N.v.t.	"pi"
Vlag	N.v.t.	geen terugkeer
,	,	","
Return	N.v.t.	"return"
spatie	spatie	" " (spatie)
Ontoegankelijk	Speciale tekentoetsen zoals @,!,^, etc.	Het teken wordt geretourneerd
N.v.t.	Functietoetsen	Geen teken geretourneerd
N.v.t.	Speciale desktop besturingstoetsen	Geen teken geretourneerd
Ontoegankelijk	Andere desktoptoetsen die niet beschikbaar zijn op de rekenmachine terwijl getKey () wacht op de volgende druk op een toets. {[,],,;, ;, ...)	Hetzelfde teken dat u krijgt in Notities (niet in een wiskundekader)

Opmerking: Het is belangrijk om op te merken dat de aanwezigheid van **getKey()** in een programma verandert hoe bepaalde gebeurtenissen door het systeem worden afgehandeld. Een aantal hiervan worden hieronder beschreven.

Beëindig programma en handel gebeurtenis af - Precies alsof de gebruiker het programma zou verlaten door op de **AAN** toets te drukken

"**Ondersteuning**" hieronder betekent - Systeem werkt zoals verwacht - programma blijft doorgaan.

Gebeurtenis	Rekenmachine	Desktop - TI-Nspire™-leerlingensoftware
Snelle peiling	Beëindig programma, handel gebeurtenis af	Hetzelfde als de rekenmachine (TI-Nspire™)

Gebeurtenis	Rekenmachine	Desktop - TI-Nspire™- leerlingensoftware
		Student Software, TI-Nspire™ Navigator™ NC Teacher Software-alleen)
Remote file mgmt (Incl. het versturen van het 'Exit Press 2 Test' bestand vanaf een andere rekenmachine of desktop-rekenmachine)	Beëindig programma, handel gebeurtenis af	Hetzelfde als de rekenmachine. (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-alleen)
Klas beëindigen	Beëindig programma, handel gebeurtenis af	Ondersteuning (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-alleen)

Gebeurtenis	Rekenmachine	Desktop - TI-Nspire™ Alle versies
TI-Innovator™ Hub verbinden/loskoppelen	Ondersteuning - kan succesvol instructies geven aan de TI-Innovator™ Hub. Nadat u het programma verlaten hebt, werkt de TI-Innovator™ Hub nog steeds met de rekenmachine.	Hetzelfde als de rekenmachine

getLangInfo()

Catalogus > 

getLangInfo() ⇒ *string*

getLangInfo()

"en"

Geeft een string die overeenkomt met de korte naam van de actieve taal. U kunt deze functie bijvoorbeeld gebruiken in een programma of functie om de huidige taal te bepalen.

Engels = "en"

Deens = "da"

Duits = "de"

Fins = "fi"

Frans = "fr"
Italiaans = "it"
Nederlands = "nl"
Vlaams = "nl_BE"
Noors = "no"
Portugees = "pt"
Spaans = "es"
Zweeds = "sv"

getLockInfo()

getLockInfo(*Var*)⇒*waarde*

Geeft de huidige status "vergrendeld" of "ontgrendeld" van variabele *Var*.

waarde =0: *Var* is ontgrendeld of bestaat niet.

waarde =1: *Var* is vergrendeld en kan niet worden gewijzigd of gewist.

Zie **Lock**, pag. 112 en **unLock**, pag. 210.

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

getMode()

getMode(*ModeNaamGeheel getal*)
⇒*waarde*

getMode(0)⇒*lijst*

getMode(*ModeNaamGeheel getal*) geeft een waarde die de huidige instelling van de modus *ModeNaamGeheel getal* representeert.

getMode(0) geeft een lijst met getallenparen. Elk paar bestaat uit een modusnummer en een instellingsnummer.

Zie onderstaande tabel voor een lijst met de modi en de bijbehorende instellingen.

getMode(0)	{ 1,7,2,1,3,1,4,1,5,1,6,1,7,1,8,1 }
getMode(1)	7
getMode(8)	1

Als u de instellingen opslaat met **getMode** (0) → *var*, dan kunt u **setMode**(*var*) gebruiken in een functie of programma om de instellingen tijdelijk te herstellen, alleen binnen de uitvoering van de functie of het programma. Zie **setMode()**, pag. 172.

Modus-naam	Modus nummer	Instellingsnummers
Cijfers weergeven	1	1=Drijvend, 2=Drijvend1, 3=Drijvend2, 4=Drijvend3, 5=Drijvend4, 6=Drijvend5, 7=Drijvend6, 8=Drijvend7, 9=Drijvend8, 10=Drijvend9, 11=Drijvend10, 12=Drijvend11, 13=Drijvend12, 14=Vast0, 15=Vast1, 16=Vast2, 17=Vast3, 18=Vast4, 19=Vast5, 20=Vast6, 21=Vast7, 22=Vast8, 23=Vast9, 24=Vast10, 25=Vast11, 26=Vast12
Hoek	2	1=Radialen, 2=Graden, 3=Decimale graden
Exponentiële opmaak	3	1=Normaal, 2=Wetenschappelijk, 3=Ingenieursnotatie
Reëel of complex	4	1=Reëel, 2=Rechthoekig, 3=Polair
Automatisch of benaderend	5	1=Automatisch, 2=Benaderend, 3=Exact
Vectoropmaak	6	1=Rechthoekig, 2=Cilindrisch, 3=Bolvormig
Grondtal	7	1=Decimaal, 2=Hexadecimaal, 3=Binair
Eenhedenstelsel	8	1=SI, 2=Eng/US

getNum()

getNum(*Uitdr1*) ⇒ *uitdrukking*

Transformeert het argument naar een uitdrukking met een vereenvoudigde gemeenschappelijke noemer, en geeft vervolgens de teller ervan.

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

GetStr

GetStr[*promptString*,] *var*[, *statusVar*]

Zie **Get** voor voorbeelden.

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

[, *statusVar*]

Programmeeropdracht: Werkt hetzelfde als de opdracht **Get**, met de uitzondering dat de ontvangen waarde altijd wordt geïnterpreteerd als een tekenreeks. De opdracht **Get** daarentegen interpreteert het antwoord als een uitdrukking, tenzij deze tussen aanhalingstekens (") is geplaatst.

Opmerking: Zie ook **Get**, pag. 82 en **Send**, pag. 168.

getType()

Catalogus >

getType(var)⇒*string*

Geeft een tekenreeks die het gegevenstype aangeeft van variabele var.

Als *var* niet gedefinieerd is, geeft dit de tekenreeks "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>	Done
<code>getType(temp)</code>	"NONE"

getVarInfo()

Catalogus >

getVarInfo()⇒*matrix of string*

getVarInfo(BibliotheekNaamString)⇒*matrix of string*

getVarInfo() geeft een matrix met informatie (variabelenaam, type, bibliotheektoegankelijkheid en de status vergrendeld of ontgrendeld) voor alle variabelen en bibliotheekobjecten die gedefinieerd zijn in de huidige opgave.

Als er geen variabelen gedefinieerd zijn, geeft **getVarInfo()** de string "NONE".

getVarInfo(BibliotheekNaamString) geeft een matrix met informatie voor alle bibliotheekobjecten die gedefinieerd zijn in bibliotheek *BibliotheekNaamString*. *BibliotheekNaamString* moet een string zijn (tekst tussen aanhalingstekens) of een stringvariabele.

Als de bibliotheek *BibliotheekNaamString* niet bestaat, treedt er een fout op.

getVarInfo()	"NONE"															
Define x=5	Done															
Lock x	Done															
Define LibPriv y={ 1,2,3 }	Done															
Define LibPub z(x)=3·x ² -x	Done															
getVarInfo()	<table><tr><td>x</td><td>"NUM"</td><td>"{"</td><td>"</td><td>1</td></tr><tr><td>y</td><td>"LIST"</td><td>"LibPriv"</td><td>"</td><td>0</td></tr><tr><td>z</td><td>"FUNC"</td><td>"LibPub"</td><td>"</td><td>0</td></tr></table>	x	"NUM"	"{"	"	1	y	"LIST"	"LibPriv"	"	0	z	"FUNC"	"LibPub"	"	0
x	"NUM"	"{"	"	1												
y	"LIST"	"LibPriv"	"	0												
z	"FUNC"	"LibPub"	"	0												
getVarInfo(tmp3)	"Error: Argument must be a string"															
getVarInfo("tmp3")	<table><tr><td>volcyl2</td><td>"NONE"</td><td>"LibPub"</td><td>"</td><td>0</td></tr></table>	volcyl2	"NONE"	"LibPub"	"	0										
volcyl2	"NONE"	"LibPub"	"	0												

getVarInfo()

Catalogus > 

Zie het voorbeeld links, waarin het resultaat van **getVarInfo()** wordt toegekend aan variabele *vs*. Als u probeert rij 2 of rij 3 van *vs* weer te geven, krijgt u de foutmelding "Ongeldige lijst of matrix" omdat minimaal één van de elementen in deze rijen (variabele *b* bijvoorbeeld) opnieuw wordt uitgewerkt naar een matrix.

Deze fout kan ook optreden wanneer u *Ans* gebruikt om een **getVarInfo()**-resultaat opnieuw uit te werken.

Het systeem geeft bovengenoemde foutmelding omdat de huidige versie van de software geen gegeneraliseerde matrixstructuur ondersteunt waarbij een element van een matrix een matrix of een lijst kan zijn.

$a:=1$	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

Catalogus > 

Goto *labelNaam*

Brengt de besturing over naar het label *labelNaam*.

labelNaam moet in dezelfde functie gedefinieerd worden met behulp van een **Lbl**-instructie.

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Define $g()$ =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	
$g()$	55

►Grad

Catalogus > 

Uitdr1 ► Grad $\Rightarrow$ uitdrukking

Converteert *Uitdr1* naar een hoek in decimale graden.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **@>Grad** in te typen.

In de hoekmodus Graden:

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

In de hoekmodus Radialen:

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

identity()Catalogus > **identity**(*Geheel getal*) $\Rightarrow$ *matrix*

Geeft de eenheidsmatrix met de afmeting *Geheel getal*.

Geheel getal moet een positief geheel getal zijn.

identity (4)	1	0	0	0
	0	1	0	0
	0	0	1	0
	0	0	0	1

IfCatalogus > 

If *BooleaanseUitdr*
Bewering

If *BooleaanseUitdr* **Then**
Blok

EndIf

```
Define g(x)=Func
    If x<0 Then
        Return x2
    EndIf
EndFunc
```

Done

g(-2) 4

Voert, als *BooleaanseUitdr* wordt uitgewerkt naar waar, de enkele bewering *Bewering* of het blok beweringen *Blok* uit alvorens door te gaan met de uitvoering.

Gaat, als *BooleaanseUitdr* wordt uitgewerkt naar onwaar, door met de uitvoering zonder de bewering of het blok beweringen uit te voeren.

Blok kan ofwel bestaan uit een enkele bewering of uit een serie beweringen die worden gescheiden door het teken ":" .

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

If BooleaanseUitdr Then*Blok1***Else***Blok2***EndIf**

Als *BooleaanseUitdr* wordt uitgewerkt naar waar, wordt *Blok1* uitgevoerd en wordt *Blok2* vervolgens overgeslagen.

Als *BooleaanseUitdr* wordt uitgewerkt naar onwaar, wordt *Blok1* overgeslagen maar wordt *Blok2* wel uitgevoerd.

Blok1 en *Blok2* kunnen bestaan uit één enkele bewering.

If BooleaanseUitdr1 Then*Blok1***ElseIf BooleaanseUitdr2 Then***Blok2*

:

ElseIf BooleaanseUitdrN Then*BlokN***EndIf**

Hiermee kunnen vertakkingen worden gemaakt. Als *BooleaanseUitdr1* wordt uitgewerkt naar waar, wordt *Blok1* uitgevoerd. Als *BooleaanseUitdr1* wordt uitgewerkt naar onwaar, wordt *BooleaanseUitdr2* uitgevoerd enz.

Define $g(x)$ = Func

Done

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

Else

Return x

EndIf

EndFunc

 $g(12)$ 12 $g(-12)$ 12Define $g(x)$ = FuncIf $x < -5$ 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()**Catalogus >

ifFn(BooleaanseUitdr, Waarde_Indien_waar [, Waarde_Indien_onwaar [, Waarde_Indien_onbekend]]) $\Rightarrow$ uitdrukking, lijst of matrix

Werkt de Booleaanse uitdrukking *BooleaanseUitdr* (of elk element uit *BooleaanseUitdr*) uit en geeft een resultaat op basis van de volgende regels:

- *BooleaanseUitdr* kan een enkele waarde, een lijst of een matrix toetsen.
- Als een element van *BooleaanseUitdr* wordt uitgewerkt naar waar, wordt het

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

De testwaarde **1** is kleiner dan 2,5, dus het overeenkomstige

Waarde_Indien_Waar-element **5** wordt gekopieerd naar de resultatenlijst.

De testwaarde **2** is kleiner dan 2,5, dus het overeenkomstige

overeenkomstige element uit *Waarde_Indien_waar* gegeven.

- Als een element van *BooleaanseUitdr* wordt uitgewerkt naar onwaar, wordt het overeenkomstige element uit *Waarde_Indien_onwaar* gegeven. Als u *Waarde_Indien_onwaar* weglaat, wordt 'ongedef' gegeven.
- Als een element van *BooleaanseUitdr* noch waar noch onwaar is, wordt het overeenkomstige element *Waarde_Indien_onbekend* gegeven. Als u *Waarde_Indien_onbekend* weglaat, wordt ongedef gegeven.
- Als het tweede, derde of vierde argument van de **ifFn()**-functie één enkele uitdrukking is, dan wordt de Booleaanse toets toegepast op elke positie in *BooleaanseUitdr*.

Opmerking: als de vereenvoudigde bewering *BooleaanseUitdr* een lijst of een matrix bevat, dan moeten alle andere lijst- of matrixargumenten dezelfde afmeting(en) hebben, en heeft het resultaat dezelfde afmeting(en).

Waarde_Indien_Waar-element 6 wordt gekopieerd naar de resultatenlijst.

De testwaarde 3 is niet kleiner dan 2,5, dus het overeenkomstige *Waarde_Indien_Onwaar*-element 10 wordt gekopieerd naar de resultatenlijst.

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

Waarde_Indien_waar is één enkele waarde en komt overeen met elke willekeurige geselecteerde positie.

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

Waarde_Indien_onwaar is niet gespecificeerd. Ongedef (ongedefinieerd) wordt gebruikt.

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

Eén element geselecteerd uit *Waarde_Indien_waar*. Eén element geselecteerd uit *Waarde_Indien_onbekend*.

imag()

imag(Uitdr1) $\Rightarrow$ uitdrukking

Geeft het imaginaire deel van het argument.

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

$$\text{imag}(z) \quad 0$$

$$\text{imag}(x+i \cdot y) \quad y$$

Opmerking: alle ongedefinieerde variabelen worden behandeld als reële variabelen. Zie ook **real()**, page 155

imag(Lijst1) $\Rightarrow$ lijst

Geeft een lijst met de imaginaire delen van de elementen.

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

imag(Matrix1) $\Rightarrow$ matrix

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

Geeft een matrix met de imaginaire delen van de elementen.

impDif()

impDif(*Vergelijking*, *Var*, *afhankelijkeVar*[*Orde*]) ⇒ *uitdrukking*

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

waarbij de *Orde* standaard 1 is.

Berekent de impliciete afgeleide voor vergelijkingen waarin één variabele gedefinieerd is in termen van een andere.

Indirectie

inString()

inString(*bronString*, *subString*[*Start*]) ⇒ *geheel getal*

$\text{inString}(\text{"Hello there"}, \text{"the"})$	7
$\text{inString}(\text{"ABCEFG"}, \text{"D"})$	0

Geeft de tekenpositie in string *bronString* waarop string *subString* voor de eerste keer begint.

Start specificeert, indien opgenomen, de tekenpositie binnen *bronString* waarop de zoekactie begint. Standaardinstelling = 1 (het eerste teken van *bronString*).

Als *bronString* niet *subString* bevat, of als geldt: *Start* > lengte van *bronString*, dan wordt er een nul gegeven als resultaat.

int()

int (*Uitdr*) ⇒ *geheel getal*

$\text{int}(-2.5)$	-3.
--------------------	-----

int(*LijstI*) ⇒ *lijst*

int(*MatrixI*) ⇒ *matrix*

$\text{int}([-1.234 \ 0 \ 0.37])$	$[-2. \ 0 \ 0.]$
-----------------------------------	------------------

Geeft het grootste gehele getal dat kleiner of gelijk is aan het argument. Deze functie is hetzelfde als **floor**().

Het argument kan een reëel of complex getal zijn.

Geeft bij een lijst of matrix het grootste gehele getal van elk van de elementen.

intDiv()

intDiv (*Getal1*, *getal2*) $\Rightarrow$ *geheel getal*

intDiv (*Lijst1*, *Lijst2*) $\Rightarrow$ *lijst*

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

Geeft het grootste gehele getal met een plus- of minteken dat kleiner of gelijk is aan (*Getal1* $\div$ *Getal2*).

Geeft bij lijsten en matrices het grootste gehele getal met een plus- of minteken dat kleiner of gelijk is aan (*argument1* $\div$ *argument2*) voor elk paar elementen.

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

integraal**interpoleren ()**

interpoleren (*xWaarde*, *xLijst*, *yLijst*, *yAccentLijst*) $\Rightarrow$ *lijst*

Deze functie doet het volgende:

Gegeven *xLijst*, *yLijst*=**f**(*xLijst*), en *yAccentLijst*=**f'**(*xLijst*) voor een onbekende functie **f**, wordt er een derdemachtsinterpolatie gebruikt om de functie **f** voor *xWaarde* te benaderen. Aangenomen wordt dat *xLijst* een lijst met monotoon stijgende of dalende getallen is, maar deze functie kan een waarde opleveren zelfs wanneer dit niet het geval is. Deze functie loopt door *xLijst* en zoekt naar een interval [*xLijst*[*i*], *xLijst*[*i*+1]] dat *xWaarde* bevat. Als de functie een dergelijk interval vindt, geeft deze een geïnterpoleerde waarde voor **f**(*xWaarde*); anders retourneert de functie **ongedef.**

Differentiaalvergelijking:

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

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

Om het hele resultaat te zien drukt u op $\blacktriangle$ en gebruikt u vervolgens $\blacktriangleleft$ en $\blacktriangleright$ om de cursor te verplaatsen.

interpoleren ()

Catalogus > 

xLijst, *yLijst* en *yAccentLijst* moeten dezelfde dimensie ≥ 2 hebben en uitdrukkingen bevatten die vereenvoudigd worden tot getallen.

xWaarde kan een ongedefinieerde variabele, een getal of een lijst met getallen zijn.

Gebruik de functie `interpolate()` om de functiewaarden voor de xwaardenlijst te berekenen:

```
xvaluelist:=seq(i,i,0,10,0.5)
{0,0.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,
xlist:=mat►list(rk[1])
{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.}
ylist:=mat►list(rk[2])
{5.,3.19499,5.00394,6.99957,9.00593,10.9979
yprimelist:=-3*y+6*t+5|y=ylist and t=xlist
{-10.,1.41503,1.98819,2.00129,1.98221,2.006
interpolate(xvaluelist,xlist,ylist,yprimelist)
{5.,2.67062,3.19499,4.02782,5.00394,6.00011}
```

inv χ^2 ()

Catalogus > 

inv χ^2 (Oppervlakte,df)

invChi2(Oppervlakte,df)

Berekent de inverse cumulatieve kansfunctie χ^2 (chi-kwadraat) die wordt gespecificeerd door de vrijheidsgraad, *df* voor een gegeven *Oppervlakte* onder de kromme.

invF()

Catalogus > 

invF(Oppervlakte,dfTeller,dfNoemer)

invF(Oppervlakte,dfTeller,dfNoemer)

berekent de inverse cumulatieve verdelingsfunctie F die wordt gespecificeerd door *dfTeller* en *dfNoemer* voor een gegeven *Oppervlakte* onder de kromme.

invBinom()

Catalogus > 

invBinom

(CumulatieveKans, AantalPogingen, Kans, UitvoerVorm) $\Rightarrow$ scalair of matrix

Gegeven het aantal pogingen (AantalPogingen) en de kans op succes van elke poging (Kans), geeft deze functie het minimum aantal successen, k , zodanig dat de cumulatieve kans op k successen groter of gelijk is aan de gegeven cumulatieve kans (CumulatieveKans)..

UitvoerVorm=0, resultaat wordt weergegeven als een scalair (standaard).

UitvoerVorm=1, resultaat wordt weergegeven als een matrix.

Voorbeeld: Mary en Kevin spelen een dobbelspel. Mary moet het maximum aantal keer raden dat 6 verschijnt in 30 worpen. Als het getal 6 precies zo veel keer of minder verschijnt, dan wint Mary. Bovendien Geldt: hoe kleiner het aantal dat ze raadt, des te groter haar winst is Wat is het kleinste getal dat Mary kan raden als ze wil dat de kans om te winnen groter is dan 77%?

$\text{invBinom}\left(0.77, 30, \frac{1}{6}\right)$	6				
$\text{invBinom}\left(0.77, 30, \frac{1}{6}, 1\right)$	<table><tr><td>5</td><td>0.616447</td></tr><tr><td>6</td><td>0.776537</td></tr></table>	5	0.616447	6	0.776537
5	0.616447				
6	0.776537				

invBinomN()

Catalogus > 

invBinomN(CumulatieveKans, Kans, AantalSucces, UitvoerVorm) $\Rightarrow$ scalair of matrix

Gegeven de kans op succes van elke poging (Kans) en het aantal successen (AantalSucces), geeft deze functie het minimum aantal pogingen, N , zodanig dat de cumulatieve kans op x successen kleiner of gelijk is aan de gegeven cumulatieve kans (CumulatieveKans).

UitvoerVorm=0, resultaat wordt weergegeven als een scalair (standaard).

UitvoerVorm=1, resultaat wordt weergegeven als een matrix.

Voorbeeld: Monique oefent doelschoten voor netbal. Ze weet uit ervaring dat haar kans op scoren bij elk schot 70% is. Ze is van plan om te oefenen totdat ze 50 doelpunten scoort. Hoeveel schoten moet ze nemen om ervoor te zorgen dat de kans dat ze ten minste 50 doelpunten zal maken meer dan 0,99 is?

$\text{invBinomN}(0.01, 0.7, 49)$	86				
$\text{invBinomN}(0.01, 0.7, 49, 1)$	<table><tr><td>85</td><td>0.010451</td></tr><tr><td>86</td><td>0.00709</td></tr></table>	85	0.010451	86	0.00709
85	0.010451				
86	0.00709				

invNorm()

Catalogus > 

invNorm(Oppervlakte[, μ [, σ]])

Berekent de inverse van de cumulatieve normale kansverdelingsfunctie voor een opgegeven Oppervlakte onder de kromme van de normale verdeling die wordt gedefinieerd door μ en σ .

invt(*Oppervlakte,df*)

Berekent de inverse van de cumulatieve student-t kansverdelingsfunctie die wordt gespecificeerd door het aantal vrijheidsgraden *df*, voor een gegeven *Oppervlakte* onder de curve.

iPart()

iPart(*Getal*) $\Rightarrow$ *geheel getal*

iPart(*Lijst1*) $\Rightarrow$ *lijst*

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

iPart(-1.234) -1.

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

Geeft het gehele deel van het argument.

Geeft bij lijsten en matrices het gehele deel van elk element.

Het argument kan een reëel of complex getal zijn.

irr()

irr(*CF0,CFLijst* [,*CFFreq*]) $\Rightarrow$ *waarde*

Financiële functie die de interne rentabiliteit van een investering berekent.

CF0 is de initiële cashflow op tijdstip 0; dit moet een reëel getal zijn.

CFLijst is een lijst met cashflow-bedragen na de initiële-cashflow *CF0*.

CFFreq is een optionele lijst waarin elk element de frequentie waarmee een gegroepeerd (opeenvolgend) cashflow-bedrag voorkomt specificeert; dit is het overeenkomstige element van *CFLijst*. De standaardwaarde is 1; als u waarden invoert, dan moeten dit positieve gehele getallen < 10.000 zijn.

list1:={6000,-8000,2000,-3000}
{6000,-8000,2000,-3000}

list2:={2,2,2,1} {2,2,2,1}

irr(5000,list1,list2) -4.64484

Opmerking: Zie ook **mirr()**, pag. 122.

isPrime(*Getal*) $\Rightarrow$ *Booleaanse constante uitdrukking*

Geeft waar of onwaar om aan te geven of *getal* een geheel getal ≥ 2 is, dat alleen deelbaar is door zichzelf en door 1.

Als *Getal* langer is dan 306 cijfers en geen factoren ≤ 1021 heeft, dan geeft **isPrime(*Getal*)** een foutmelding weer.

Als u alleen maar wilt bepalen of *Getal* een priemgetal is, gebruik dan **isPrime()** in plaats van **factor()**. Dat is veel sneller, vooral als *Getal* geen priemgetal is en een op één na grootste factor heeft van meer dan ongeveer vijf cijfers.

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

isPrime(5)	true
isPrime(6)	false

Functie om het volgende priemgetal groter dan een gespecificeerd getal te vinden:

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

isVoid(*Var*) $\Rightarrow$ *Booleaanse constante uitdrukking*

isVoid(*Uitdr*) $\Rightarrow$ *Booleaanse constante uitdrukking*

isVoid(*Lijst*) $\Rightarrow$ *lijst van Booleaanse constante uitdrukkingen*

Geeft waar of onwaar om aan te geven of het argument een leeg gegevenstype is.

Zie voor meer informatie over lege elementen pag. 267.

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

Lbl	Catalogus > 
Lbl <i>labelNaam</i> Definieert een label met de naam <i>labelNaam</i> binnen een functie. U kunt een Goto <i>labelNaam</i>-instructie gebruiken om de besturing naar de instructie onmiddellijk na het label te brengen. <i>labelNaam</i> moet aan dezelfde naamgevingsvereisten voldoen als een variabelenaam. Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.	Define $g()$=Func Local <i>temp,i</i> $0 \rightarrow temp$ $1 \rightarrow i$ Lbl <i>top</i> $temp+i \rightarrow temp$ If $i < 10$ Then $i+1 \rightarrow i$ Goto <i>top</i> EndIf Return <i>temp</i> EndFunc Done $g()$ 55

lcm()	Catalogus > 
lcm(<i>Getal1, Getal2</i>)⇒<i>uitdrukking</i> lcm(<i>Lijst1, Lijst2</i>)⇒<i>lijst</i> lcm(<i>Matrix1, Matrix2</i>)⇒<i>matrix</i> Geeft het kleinste gemene veelvoud van de twee argumenten. De lcm van twee breuken is de lcm van hun tellers gedeeld door de gcd (grootste gemene veelvoud) van hun noemers. De lcm van breukgetallen met een drijvende komma is hun product. Geeft bij twee lijsten of matrices de kleinste gemene veelvoud van de overeenkomstige elementen.	lcm(6,9) 18 lcm($\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}$) $\left\{\frac{2}{3}, 14, 80\right\}$

left()	Catalogus > 
left(<i>bronString[, Aantal]</i>)⇒<i>string</i> Geeft het meest linkse <i>Aantal</i> tekens in tekenreeks <i>bronString</i>.	left("Hello",2) "He"

Als u *Aantal* weglaat, wordt de hele *bronString* gegeven.

left(*Lijst1* [, *Aantal*]) ⇒ *lijst*

Geeft het meest linkse *Aantal* elementen in *Lijst1*.

Als u *Aantal* weglaat, wordt de hele *Lijst1* gegeven.

left(*Vergelijken*) ⇒ *uitdrukking*

Geeft het linkerlid van een vergelijking of ongelijkheid.

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

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

libShortcut()

libShortcut(*BibliotheekNaamString*, *SnelNaamString* [, *LibPrivVlag*]) ⇒ *lijst met variabelen*

Creëert een variabelegroep in de huidige opgave die verwijzingen naar alle objecten in het gespecificeerde bibliotheekdocument *BibliotheekNaamString* bevat. Voegt de groepsleden tevens toe aan het Variabelen-menu. U kunt vervolgens naar elk object verwijzen met behulp van zijn *SnelNaamString*.

Stel *LibPrivVlag*=0 om persoonlijke bibliotheekobjecten uit te sluiten (standaardinstelling)

Stel *LibPrivVlag*=1 om persoonlijke bibliotheekobjecten op te nemen

Zie voor het kopiëren van een variabelegroep **CopyVar** (pag. 30).

Zie voor het wissen van een variabelegroep **DelVar** (pag. 51).

In dit voorbeeld wordt uitgegaan van een op de juiste manier opgeslagen en vernieuwd bibliotheekdocument met de naam **linalg2** dat de gedefinieerde objecten *clearmat*, *gauss1* en *gauss2* bevat.

```
getVarInfo("linalg2")
┌ clearmat  "FUNC"  "LibPub "
│ gauss1    "PRGM"  "LibPriv "
│ gauss2    "FUNC"  "LibPub "
└──────────┘
```

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

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

limit(Uitdr1, Var, Punt [,Richting])
 $\Rightarrow$ uitdrukking

limit(Lijst1, Var, Punt [, Richting]) $\Rightarrow$ lijst

limit(Matrix1, Var, Punt [, Richting])
 $\Rightarrow$ matrix

Geeft de gevraagde limiet.

Opmerking: zie ook **Limiet-template**, pag. 6.

Richting: negatief=van links, positief=van rechts, anders=beide. (Als *Richting* wordt weggelaten is de standaardinstelling "beide".)

Limieten naar positief ∞ en negatief ∞ worden altijd geconverteerd naar eenzijdige limieten vanaf de eindige zijde.

Afhankelijk van de omstandigheden geeft **limit()** zichzelf of undef wanneer hij geen unieke limiet kan bepalen. Dit betekent niet noodzakelijk dat er geen unieke limiet bestaat. undef betekent dat het resultaat ofwel een onbekend getal met een eindige of oneindige grootte is, of dat het de hele verzameling van dergelijke getallen is.

limit() maakt gebruik van methodes als de regel van L'Hopital, dus er zijn unieke limieten die deze functie niet kan bepalen. Als *Uitdr1* andere onbepaalde variabelen bevat dan *Var*, dan moet u deze mogelijk beperken om een beknopter resultaat te verkrijgen.

Limieten kunnen zeer gevoelig zijn voor afrondingsfouten. Vermijd indien mogelijk de instelling Benaderend van de modus **Automatisch of Benaderend** en benaderende getallen bij het berekenen van limieten. Anders kan het gebeuren dat limieten die nul zouden moeten zijn, of een oneindige grootte zouden moeten hebben dit niet hebben, en limieten die een eindige grootte van niet nul zouden moeten hebben, dit niet hebben.

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

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

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

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

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

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

$$\lim_{x \rightarrow \infty} (a^x) \Big|_{a > 1} \quad \infty$$

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

LinRegBx $X, Y, [Freq], [Categorie, Opnemen]$

Berekent de lineaire regressie $y = a + b \cdot x$ op de lijsten X en Y met frequentie $Freq$. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en Y zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig X - en Y -gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige X - en Y -gegevens.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a + b \cdot x$
stat.a, stat.b	Regressiecoëfficiënten
stat. r^2	Determinatiecoëfficiënt
stat.r	Correlatiecoëfficiënt
stat.Resid	Residuen uit de regressie
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

LinRegMx $X, Y, [Freq], [Categorie, Opnemen]$

Berekent de lineaire regressie $y = m \cdot x + b$ op de lijsten X en Y met frequentie $Freq$. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en Y zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig X - en Y -gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige X - en Y -gegevens.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $m \cdot x + b$
stat.m, stat.b	Regressiecoëfficiënten
stat.r ²	Determinatiecoëfficiënt
stat.r	Correlatiecoëfficiënt
stat.Resid	Residuen uit de regressie
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

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

Voor helling. Berekent een niveau C betrouwbaarheidsinterval voor de helling.

LinRegtIntervals $X, Y[, F[, 1, Xwaarde[, CNiv]]]$

Voor respons. Berekent een voorspelde y-waarde, een niveau C voorspellingsinterval voor één observatie en een niveau C betrouwbaarheidsinterval voor de gemiddelde respons.

Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben.

X en Y zijn lijsten met onafhankelijke en afhankelijke variabelen.

F is een optionele lijst met frequentiewaarden. Elk element in F specificeert de frequentie waarmee elk overeenkomstig X - en Y -gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a + b \cdot x$
stat.a, stat.b	Regressiecoëfficiënten
stat.df	Vrijheidsgraden
stat.r ²	Determinatiecoëfficiënt
stat.r	Correlatiecoëfficiënt
stat.Resid	Residuen uit de regressie

Alleen voor het type Helling

Uitvoervariabele	Beschrijving
[stat.CLower, stat.CUpper]	Betrouwbaarheidsinterval voor de helling
stat.ME	Foutmarge betrouwbaarheidsinterval
stat.SESlope	Standaardfout van helling

Uitvoervariabele	Beschrijving
stat.s	Standaardfout van de lijn

Alleen voor het type Respons

Uitvoervariabele	Beschrijving
[stat.CLower, stat.CUpper]	Betrouwbaarheidsinterval voor de gemiddelde respons
stat.ME	Foutmarge betrouwbaarheidsinterval
stat.SE	Standaardfout van de gemiddelde respons
[stat.LowerPred, stat.UpperPred]	Voorspellingsinterval voor één observatie
stat.MEPred	Foutmarge voor voorspellingsinterval
stat.SEPred	Standaardfout voor voorspelling
stat. $\hat{y}$	$a + b \cdot X$ Waarde

LinRegtTest

Catalogus > 

LinRegtTest $X, Y[, Freq[, Hypoth]]$

Berekent een lineaire regressie op de X - en Y -lijsten en een t -toets op de waarde van helling β en de correlatiecoëfficiënt ρ voor de vergelijking $y = \alpha + \beta x$. Hij toetst de nulhypothese $H_0: \beta = 0$ (equivalent $\rho = 0$) tegen één van de drie alternatieve hypothesen.

Alle lijsten moeten gelijke afmetingen hebben.

X en Y zijn lijsten met onafhankelijke en afhankelijke variabelen.

$Freq$ is een optionele lijst met frequentiewaarden. Elk element in $Freq$ specificeert de frequentie waarmee elk overeenkomstig X - en Y -gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

$Hypoth$ is een optionele waarde die één van de drie alternatieve hypothesen specificeert, waartegen de nulhypothese ($H_0: \beta = \rho = 0$) wordt getoetst.

Voor $H_1: \beta \neq 0$ en $\rho \neq 0$ (standaard) stelt u $Hypoth = 0$ in

Voor $H_1: \beta < 0$ en $\rho < 0$ stelt u $Hypoth < 0$ in

Voor $H_1: \beta > 0$ en $\rho > 0$ stelt u $Hypoth > 0$ in

Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a + b \cdot x$
stat.t	<i>t</i> -statistiek voor significantietoets
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.df	Vrijheidsgraden
stat.a, stat.b	Regressiecoëfficiënten
stat.s	Standaardfout van de lijn
stat.SESlope	Standaardfout van helling
stat.r ²	Determinatiecoëfficiënt
stat.r	Correlatiecoëfficiënt
stat.Resid	Residuen uit de regressie

linSolve()

linSolve(*StelselLineaireVgl*, *Var1*, *Var2*, ...) ⇒ *lijst*

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

linSolve(*LineaireVgl1* en *LineaireVgl2* en ..., *Var1*, *Var2*, ...) ⇒ *lijst*

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

linSolve({*LineaireVgl1*, *LineaireVgl2*, ...}, *Var1*, *Var2*, ...) ⇒ *lijst*

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

linSolve(*StelselLineaireVgl*, {*Var1*, *Var2*, ...}) ⇒ *lijst*

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

linSolve(*LineaireVgl1* en *LineaireVgl2* en ..., {*Var1*, *Var2*, ...}) ⇒ *lijst*

linSolve({*LineaireVgl1*, *LineaireVgl2*, ...}, {*Var1*, *Var2*, ...}) ⇒ *lijst*

Geeft een lijst met oplossingen voor de variabelen *Var1*, *Var2*, ...

Het eerste argument moet uitgewerkt worden tot een stelsel lineaire vergelijkingen of tot één lineaire vergelijking. Anders treedt er een argumentfout op.

Bijvoorbeeld: het uitwerken van **linSolve(x=1 en x=2,x)** levert een "Argument Error" op.

ΔList()

ΔList(Lijst1)⇒lijst

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

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **deltaList(...)** in te typen.

Geeft een lijst met de verschillen tussen opeenvolgende elementen in *Lijst1*. Ieder element van *Lijst1* wordt afgetrokken van het volgende element van *Lijst1*. De resulterende lijst is altijd één element korter dan de oorspronkelijke *Lijst1*.

list►mat()

list►mat(Lijst [, elementenPerRij])
⇒matrix

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

Geeft een matrix die rij voor rij gevuld wordt met de elementen uit *Lijst*.

elementenPerRij specificeert, indien opgenomen, het aantal elementen per rij. De standaardwaarde is het aantal element in *Lijst* (één rij).

Als *Lijst* de resulterende matrix niet vult, dan worden er nullen toegevoegd.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **list@>mat(...)** in te typen.

Uitdr ►ln⇒*uitdrukking*

Zorgt ervoor dat de invoer *Uitdr* wordt geconverteerd in een uitdrukking met alleen natuurlijke logaritmes (ln).

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door $\text{e} > \ln$ in te typen.

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

ln()

  -toetsen

ln(*Uitdr1*)⇒*uitdrukking*

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

ln(*Lijst1*)⇒*lijst*

Geeft de natuurlijke logaritme van het argument.

Als de complexe opmaak-modus Reëel is:

Geeft bij een lijst de natuurlijke logaritme van de elementen.

$$\ln(\{-3, 1.2, 5\}) \quad \text{"Error: Non-real calculation"}$$

ln(*vierkanteMatrix1*)⇒*vierkanteMatrix*

Geeft de natuurlijke logaritme van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de natuurlijke logaritme van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

Als de complexe opmaak-modus Rechthoekig is:

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

In de hoekmodus Radialen en rechthoekige complexe opmaak:

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

Om het hele resultaat te zien drukt u op $\blacktriangle$ en gebruikt u vervolgens $\blacktriangleleft$ en $\blacktriangleright$ om de cursor te verplaatsen.

LnReg

Catalogus > 

LnReg *X*, *Y* [, [*Freq*] [, [*Categorie*, *Opnemen*]]

Berekent de logaritmische regressie $y = a + b \cdot \ln(x)$ op de lijsten X en Y met frequentie *Freq*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en Y zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig X - en Y -gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriëcodes voor de overeenkomstige X - en Y -gegevens.

Opnemen is een lijst met één of meer van de categoriëcodes. Alleen de gegevens waarvan de categoriëcode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a + b \cdot \ln(x)$
stat.a, stat.b	Regressiecoëfficiënten
stat.r ²	Coëfficiënt van lineaire determinatie voor getransformeerde gegevens
stat.r	Correlatiecoëfficiënt voor getransformeerde gegevens ($\ln(x)$, y)
stat.Resid	Residuen die geassocieerd zijn met het logaritmische model
stat.ResidTrans	Residuen die geassocieerd zijn met de lineaire regressie van getransformeerde gegevens
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categoriëen opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categoriëen opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

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

Maakt de gespecificeerde *vars* bekend als lokale variabelen. Die variabelen bestaan alleen tijdens de uitwerking van een functie, en worden gewist wanneer de functie uitgevoerd is.

Opmerking: lokale variabelen besparen geheugen omdat ze slechts tijdelijk bestaan. Bovendien storen ze eventuele bestaande algemene variabelen niet. Lokale variabelen moeten gebruikt worden voor **For**-lussen en voor het tijdelijk opslaan van waarden in een functie van meerdere regels, aangezien wijzigingen van algemene variabelen niet zijn toegestaan in een functie.

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

```
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

<i>rollcount()</i>	16
--------------------	----

<i>rollcount()</i>	3
--------------------	---

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

Vergrendelt de gespecificeerde variabelen of variabelegroep. Vergrendelde variabelen kunnen niet worden gewijzigd of gewist.

U kunt de systeemvariabele *Ans* niet vergrendelen of ontgrendelen, en u kunt de systeemvariabelegroepen *stat.* en *tvm.* niet vergrendelen.

Opmerking: Het commando **Vergrendelen (Lock)** wist de Ongedaan maken/Overdoen-geschiedenis als het wordt toegepast op niet-vergrendelde variabelen.

Zie **unLock**, pag. 210 en **getLockInfo()**, pag. 88.

<i>a:=65</i>	65
--------------	----

Lock <i>a</i>	<i>Done</i>
---------------	-------------

getLockInfo(<i>a</i>)	1
-------------------------	---

<i>a:=75</i>	"Error: Variable is locked."
--------------	------------------------------

DelVar <i>a</i>	"Error: Variable is locked."
-----------------	------------------------------

Unlock <i>a</i>	<i>Done</i>
-----------------	-------------

<i>a:=75</i>	75
--------------	----

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

log()

ctrl 10^x -toetsen

log(Uitdr1[,Uitdr2])⇒*uitdrukking*

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

log(Lijst1[,Uitdr2])⇒*lijst*

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

Geeft de logaritme met grondtal-*Uitdr2*- van het eerste argument.

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

Opmerking: zie ook **Log-template**, pag. 2.

Geeft bij een lijst de logaritme met grondtal-*Uitdr2*- van de elementen.

Als de complexe opmaak-modus Reëel is:

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

Als het tweede argument wordt weggelaten, dan wordt 10 als grondtal gebruikt.

Als de complexe opmaak-modus
Rechthoekig is:

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

log(vierkanteMatrix1[,Uitdr])
⇒*vierkanteMatrix*

In de hoekmodus Radialen en rechthoekige
complexe opmaak:

$$\log_{10} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \\ \begin{bmatrix} 0.795387 + 0.753438 \cdot i & 0.003993 - 0.64747 \cdot i \\ 0.194895 - 0.315095 \cdot i & 0.462485 + 0.27077 \cdot i \\ -0.115909 - 0.904706 \cdot i & 0.488304 + 0.77746 \cdot i \end{bmatrix}$$

Geeft de logaritme met grondtal *Uitdr* van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de logaritme met grondtal-*Uitdr* van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

Om het hele resultaat te zien drukt u op **▲** en gebruikt u vervolgens **◀** en **▶** om de cursor te verplaatsen.

Als het grondtal-argument wordt weggelaten, dan wordt 10 als grondtal gebruikt.

►logbase

Catalogus >

Uitdr ►**logbase(Uitdr1)**⇒*uitdrukking*

Zorgt ervoor dat de invoer *Uitdrukking* vereenvoudigd wordt tot een uitdrukking met het grondtal *Uitdr1*.

$$\log_3 (10) - \log_5 (5) \text{ ► } \log_{\text{base}(5)} (5) \quad \log_5 \left(\frac{10}{3} \right) \\ \log_5 (3)$$

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>logbase (...) in te typen.

Logistic

Logistic *X*, *Y* [, [*Freq*] [, *Categorie*, *Opnemen*]]

Berekent de logistische regressie $y = c/(1+a \cdot e^{-bx})$ op de lijsten *X* en *Y* met frequentie *Freq*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en *Y* zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig *X*- en *Y*-gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige *X*- en *Y*-gegevens.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriencode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $c/(1+a \cdot e^{-bx})$
stat.a, stat.b, stat.c	Regressiecoëfficiënten
stat.Resid	Residuen uit de regressie
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>

Uitvoervariabele	Beschrijving
stat.YReg	Lijst van gegevens in de gemodificeerde <i>Y</i> Lijst die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

LogisticD

Catalogus > 

LogisticD *X*, *Y* [, [*Iteraties*], [*Freq*] [, *Categorie*, *Opnemen*]]

Berekent de logistische regressie $y = (c/(1+a \cdot e^{-bx})+d)$ op de lijsten *X* en *Y* met frequentie *Freq*, met behulp van een gespecificeerd aantal *Iteraties*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en *Y* zijn lijsten met onafhankelijke en afhankelijke variabelen.

Iteraties is een optionele waarde die het maximaal aantal keer specificeert dat een oplossing wordt geprobeerd. Als deze wordt weggelaten, wordt 64 gebruikt. Doorgaans leiden grotere waarden tot een hogere nauwkeurigheid maar een langere berekeningstijd, en andersom.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig *X*- en *Y*-gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige *X*- en *Y*-gegevens.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $c/(1+a \cdot e^{-bx})+d$

Uitvoervariabele	Beschrijving
stat.a, stat.b, stat.c, stat.d	Regressiecoëfficiënten
stat.Resid	Residuen uit de regressie
stat.XReg	Lijst van gegevens in de gemoedificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemoedificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

Loop

Catalogus >


Loop
Blok
EndLoop

Voert de beweringen in *Blok* herhaaldelijk uit. Merk op dat de lus eindeloos wordt uitgevoerd, tenzij er een **Goto**- of **Exit**-instructie wordt uitgevoerd binnen *Blok*.

Blok is een reeks beweringen die gescheiden worden door het teken “:”.

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

```

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

```

	<i>Done</i>
<i>rollcount()</i>	16
<i>rollcount()</i>	3

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

Berekent de Doolittle LU (beneden-boven)-decompositie van een reële of complexe matrix. De benedendriehoeksmatrix wordt opgeslagen in *lMatrix*, de bovendriehoeksmatrix in *uMatrix* en de permutatiematrix (die de rijverwisselingen tijdens de berekening beschrijft) in *pMatrix*.

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

Optioneel wordt elk matrixelement behandeld als nul als de absolute waarde ervan minder dan *Tol* is. Deze tolerantie wordt alleen gebruikt als de matrix gegevens met een drijvende komma heeft, en geen symbolische variabelen bevat die geen waarde toegekend hebben gekregen. Anders wordt *Tol* genegeerd.

- Als u   gebruikt of de modus **Automatisch of Benaderend** instelt op Benaderend, dan worden berekeningen met behulp van de drijvende komma uitgevoerd.
- Als *Tol* wordt weggelaten of niet wordt gebruikt, dan wordt de standaardtolerantie berekend als: $5E-14 \cdot \max(\dim(Matrix)) \cdot \text{rowNorm}(Matrix)$

Het LU ontbindingsalgoritme gebruikt gedeeltelijke pivoting met rijverwisselingen.

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

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

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

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

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

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

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

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

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

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

M

mat►list()

mat►list(*Matrix*) ⇒ *lijst*

Geeft een lijst die gevuld is met de elementen in *Matrix*. De elementen worden rij voor rij gekopieerd uit *Matrix*.

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

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

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

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door `mat@>list(...)` in te typen.

max()

Catalogus >

max(Uitdr1, Uitdr2) ⇒ uitdrukking

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

max(Lijst1, Lijst2) ⇒ lijst

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

max(Matrix1, Matrix2) ⇒ matrix

Geeft het maximum van de twee argumenten. Als de argumenten twee lijsten of matrices zijn, dan wordt een lijst of matrix met de maximumwaarde van elk paar corresponderende gegevens gegeven.

max(Lijst) ⇒ uitdrukking

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

Geeft het maximalelement in *lijst*.

max(Matrix1) ⇒ matrix

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

Geeft een rijvector met het maximalelement van elke kolom in *Matrix1*.

Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

Opmerking: zie ook `fMax()` en `min()`.

mean()

Catalogus >

mean(Lijst[, freqLijst]) ⇒ uitdrukking

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

Geeft het gemiddelde van de elementen in *Lijst*.

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

Elk element uit *freqLijst* telt het aantal malen dat het overeenkomstige element in *Lijst* achter elkaar voorkomt.

mean(Matrix1[, freqMatrix]) ⇒ matrix

In de rechthoekige vectoropmaak:

Geeft een rijvector van de gemiddelden van alle kolommen in *Matrix1*.

mean()

Catalogus > 

Elk element uit *freqMatrix* telt het aantal malen dat het overeenkomstige element in *Matrix1* achter elkaar voorkomt.

Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

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

median()

Catalogus > 

median(Lijst[,freqLijst])⇒*uitdrukking*

Geeft de mediaan van de elementen in *Lijst*.

Elk element uit *freqLijst* telt het aantal malen dat het overeenkomstige element in *Lijst* voorkomt.

median(Matrix1[,freqMatrix])⇒*matrix*

Geeft een rijvector met de medianen van de kolommen in *Matrix1*.

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

Elk element uit *freqMatrix* telt het aantal malen dat het overeenkomstige element in *Matrix1* achter elkaar voorkomt.

Opmerkingen:

- alle gegevens in de lijst of matrix moeten vereenvoudigen tot getallen.
- Lege elementen in de lijst of matrix worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

MedMed

Catalogus > 

MedMed X,Y[,Freq][,Categorie,Opnemen]

Berekent de mediaan-mediaan-lijn $y = (m \cdot x + b)$ op de lijsten *X* en *Y* met frequentie *Freq*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en *Y* zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden.

Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig *X*- en *Y*-gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige *X*- en *Y*-gegevens..

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Mediaan-mediaan-lijnvergelijking: $m \cdot x + b$
stat.m, stat.b	Modelcoëfficiënten
stat.Resid	Residuen uit de mediaan-mediaan-lijn
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

mid()

mid(bronString, Start[, Aantal]) $\Rightarrow$ string

Geeft *Aantal* tekens uit de tekenreeks *bronString*, beginnend met teken 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)	" "

Als *Aantal* wordt weggelaten of groter is dan de afmeting van *bronString*, dan worden alle tekens van *bronString* gegeven, beginnend met het teken nummer *Start*.

Aantal moet ≥ 0 zijn. Als *Aantal* = 0, dan wordt een lege string gegeven.

mid(*bronLijst*, *Start* [, *Aantal*]) $\Rightarrow$ *lijst*

Geeft *Aantal* elementen uit *bronLijst*, beginnend met element nummer *Start*.

Als *Aantal* wordt weggelaten of groter is dan de afmeting van *bronLijst*, dan worden alle elementen uit *bronLijst* gegeven, beginnend met element nummer *Start*.

Aantal moet ≥ 0 zijn. Als *Aantal* = 0, dan wordt er een lege lijst gegeven.

mid(*bronStringLijst*, *Start* [, *Aantal*]) $\Rightarrow$ *lijst*

Geeft *Aantal* strings uit de lijst met strings *bronStringLijst*, beginnend met element nummer *Start*.

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

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

min(*Uitdr1*, *Uitdr2*) $\Rightarrow$ *uitdrukking*

min(*Lijst1*, *Lijst2*) $\Rightarrow$ *lijst*

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

Geeft het minimum van de twee argumenten. Als de argumenten twee lijsten of matrices zijn, dan wordt een lijst of matrix met de minimumwaarden van elk paar corresponderende gegevens gegeven.

min(*Lijst*) $\Rightarrow$ *uitdrukking*

Geeft het minimelement van *Lijst*.

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

Geeft een rijvector met het minimelement van elke kolom in *Matrix1*.

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

$\text{min}(\{0,1,-7,1,3,0,5\})$	-7
----------------------------------	----

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

Opmerking: zie ook **fMin()** en **max()**.

mirr()

mirr

(
financPercentage
,herinvestPercentage,CF0,CFLijst
[,CFFreq])

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

Financiële functie die de gewijzigde interne rentabiliteit van een investering geeft.

financPercentage is het rentepercentage dat u betaalt over de cashflow-bedragen.

herinvestPercentage is het rentepercentage waarop de cashflows opnieuw geïnvesteerd worden.

CF0 is de begin-cashflow op tijdstip 0; dit moet een reëel getal zijn.

CFLijst is een lijst met cashflow-bedragen na de begin-cashflow *CF0*.

CFFreq is een optionele lijst waarin elk element de frequentie waarmee een gegroepeerde (opeenvolgend) cashflow-bedrag voorkomt specificeert; dit is het overeenkomstige element van *CFLijst*. De standaardwaarde is 1; als u waarden invoert, dan moeten dit positieve gehele getallen < 10.000 zijn.

Opmerking: zie ook **irr()**, pag. 99.

mod()

mod(*Uitdr1*, *Uitdr2*) ⇒ *uitdrukking*

<i>mod</i> (7,0)	7
------------------	---

mod(*Lijst1*, *Lijst2*) ⇒ *lijst*

<i>mod</i> (7,3)	1
------------------	---

mod(*Matrix1*, *Matrix2*) ⇒ *matrix*

<i>mod</i> (-7,3)	2
-------------------	---

Geeft het eerste argument modulus het tweede argument zoals gedefinieerd wordt door de identiteiten:

<i>mod</i> (7,-3)	-2
-------------------	----

<i>mod</i> (-7,-3)	-1
--------------------	----

<i>mod</i> ({ 12, -14, 16 }, { 9, 7, -5 })	{ 3, 0, -4 }
--	--------------

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

mod()Catalogus > 

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

Wanneer het tweede argument niet-nul is, dan is het resultaat periodiek in dat argument. Het resultaat is nul of heeft hetzelfde teken als het tweede argument.

Als de argumenten twee lijsten of twee matrices zijn, dan wordt een lijst of matrix met de modulus van elk paar corresponderende elementen gegeven.

Opmerking: zie ook **remain()**, pag. 158

mRow()Catalogus > 

$$\text{mRow}(\text{Uitdr}, \text{Matrix1}, \text{Index}) \Rightarrow \text{matrix}$$

Geeft een kopie van *Matrix1* met elk element in rij *Index* van *Matrix1* vermenigvuldigd met *Uitdr*.

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

mRowAdd()Catalogus > 

$$\text{mRowAdd}(\text{Uitdr}, \text{Matrix1}, \text{Index1}, \text{Index2}) \Rightarrow \text{matrix}$$

Geeft een kopie van *Matrix1* met elk element in rij *Index2* van *Matrix1* vervangen door:

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

$$\text{mRowAdd}\left(n, \begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right) \quad \begin{bmatrix} a & b \\ a \cdot n + c & b \cdot n + d \end{bmatrix}$$

Uitdr · rij *Index1* + rij *Index2*

MultRegCatalogus > 

$$\text{MultReg } Y, X1[, X2[, X3[, \dots [, X10]]]$$

Berekent een meervoudige lineaire regressie van lijst *Y* op de lijsten *X1*, *X2*, ..., *X10*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.b0, stat.b1, ...	Regressiecoëfficiënten
stat.R ²	Coëfficiënt van meervoudige determinatie
stat.ŷ Lijst	$\hat{y}$ Lijst = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residuen uit de regressie

MultRegIntervals

Catalogus > 

MultRegIntervals *Y, X1[,X2[,X3,...*
[,X10]]], XWaardeLijst[,CNiveau]

Berekent een voorspelde y-waarde, een niveau C voorspellingsinterval voor één observatie en een niveau C betrouwbaarheidsinterval voor de gemiddelde respons.

Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.ŷ	Een puntschatting: $\hat{y} = b_0 + b_1 \cdot x_1 + \dots$ voor <i>XWaardeLijst</i>
stat.dfError	Vrijheidsgraden van de fouten
stat.CLower, stat.CUpper	Betrouwbaarheidsinterval voor een gemiddelde respons
stat.ME	Foutmarge betrouwbaarheidsinterval
stat.SE	Standaardfout van de gemiddelde respons
stat.LowerPred, stat.UpperPred	Voorspellingsinterval voor één observatie
stat.MEPred	Foutmarge voor voorspellingsinterval
stat.SEPred	Standaardfout voor voorspelling
stat.bList	Lijst van regressiecoëfficiënten, {b0,b1,b2,...}

Uitvoervariabele	Beschrijving
stat.Resid	Residuen uit de regressie

MultRegTests

Catalogus > 

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

Meervoudige lineaire regressietoets berekent een meervoudige lineaire regressie op de gegevens, en biedt de globale F -toets-statistiek en t -toets-statistieken voor de coëfficiënten.

Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoer

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots$
stat.F	Globale F -toets-statistiek
stat.PVal	P-waarde geassocieerd met de globale F -statistiek
stat.R ²	Coëfficiënt van meervoudige determinatie
stat.AdjR ²	Aangepaste coëfficiënt van meervoudige determinatie
stat.s	Standaarddeviatie van de fout
stat.DW	Durbin-Watson-statistiek; wordt gebruikt om te bepalen of er automatische correlatie van de eerste orde aanwezig is in het model
stat.dfReg	Vrijheidsgraden van de regressie
stat.SSReg	Som van de kwadraten van de regressies
stat.MSReg	Gemiddelde kwadraat van de regressies
stat.dfError	Vrijheidsgraden van de fouten
stat.SSError	Som van de kwadraten van de fouten
stat.MSError	Gemiddelde kwadraat van de fouten
stat.bList	{ $b_0, b_1, \dots$ } Lijst van coëfficiënten
stat.tList	Lijst van t -statistieken, één voor elke coëfficiënt in de bList

Uitvoervariabele	Beschrijving
stat.PList	Lijst van P-waarden voor elke t-statistiek
stat.SEList	Lijst van standaardfouten voor coëfficiënten in bLijst
stat.ŷ Lijst	$\hat{y}$ Lijst = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residuen uit de regressie
stat.sResid	Gestandaardiseerde residuen; verkregen door een residu te delen door zijn standaarddeviatie
stat.CookDist	Afstand van Cook; maat voor de invloed van een observatie op basis van het residu en de invloed
stat.Leverage	Maat voor hoever de waarden van de onafhankelijke variabelen van hun gemiddelde waarden af liggen

N

nand (niet en)

toetsen

BooleaanseUitdr1 **nand** *BooleaanseUitdr2*
 levert *Booleaanse uitdrukking*

$x \geq 3$ and $x \geq 4$	$x \geq 4$
$x \geq 3$ nand $x \geq 4$	$x < 4$

BooleaanseLijst1 **nand** *BooleaanseLijst2*
 levert *Booleaanse lijst*

BooleaanseMatrix1 **nand**
BooleaanseMatrix2 levert *Booleaanse matrix*

Geeft de ontkenning (negatie) van een logische **and** bewerking op de twee argumenten. Geeft waar, onwaar of een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

Geheel getal1 **nand** *Geheel getal2* $\Rightarrow$ *geheel getal*

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

Vergelijkt twee reële gehele getallen bit voor bit met behulp van een **nand**-bewerking. Intern worden beide gehele getallen geconverteerd naar 64-bits binaire getallen met een teken (positief of negatief). Wanneer overeenkomstige bits vergeleken worden, is het resultaat 0 als beide bits 1 zijn; anders is het resultaat 1. De geretourneerde waarde vertegenwoordigt de bitresultaten, en wordt weergegeven volgens de ingestelde grondtal-modus.

U kunt de gehele getallen invoeren in elk talstelsel. Voor een binaire of hexadecimale invoer moet u respectievelijk het voorvoegsel 0b of 0h gebruiken. Zonder voorvoegsel worden gehele getallen behandeld als decimaal (grondtal 10).

nCr()

Catalogus >

nCr(Uitdr1, Uitdr2)⇒uitdrukking

$nCr(z, 3)$	$\frac{z \cdot (z-2) \cdot (z-1)}{6}$
$Ans z=5$	10
$nCr(z, c)$	$\frac{z!}{c! \cdot (z-c)!}$
$\frac{Ans}{nPr(z, c)}$	$\frac{1}{c!}$

Voor geheel getal *Uitdr1* en *Uitdr2* met $Uitdr1 \geq Uitdr2 \geq 0$, is **nCr()** het aantal combinaties van *Uitdr1* dingen die met *Uitdr2* keer tegelijk zijn genomen. (Dit is ook bekend als een binomiale coëfficiënt.) Beide argumenten kunnen gehele getallen of symbolische uitdrukkingen zijn.

nCr(Uitdr, 0)⇒1

nCr(Uitdr, negGeheel getal)⇒0

nCr(Uitdr, posGeheel getal)⇒ Uitdr · (Uitdr-1) ... (Uitdr-posGeheel getal+1)/ posGeheel getal!

nCr(Expr, nietGeheel getal)⇒uitdrukking!/ ((Uitdr-nietGeheel getal)! ·nietGeheel getal!)

nCr(Lijst1, Lijst2)⇒lijst

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

Geeft een lijst met combinaties op basis van de overeenkomstige elementparen in de twee lijsten. De argumenten moeten lijsten van dezelfde afmeting zijn.

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

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

Geeft een matrix met combinaties op basis van de overeenkomstige elementparen in de twee matrices. De argumenten moeten matrices van dezelfde afmeting zijn.

nDerivative()Catalogus > **nDerivative**(*Uitdr1*, *Var*=*Waarde*[,*Orde*])
 $\Rightarrow$ *waarde*

$nDerivative(x , x=1)$	1
-------------------------	---

$nDerivative(x , x) _{x=0}$	undef
------------------------------	-------

nDerivative(*Uitdr1*, *Var*[,*Orde*]) |
Var=*Waarde* $\Rightarrow$ *waarde*

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

Geeft de numerieke afgeleide die berekend is met automatische differentiatiemethodes.

Wanneer *Waarde* gespecificeerd is, wordt elke eerdere variabeletoekenning of elke huidige "|" -substitutie voor de variabele onderdrukt.

Orde van de afgeleide moet 1 of 2 zijn.

newList()Catalogus > **newList**(*aantalElementen*) $\Rightarrow$ *lijst*

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

Geeft een lijst met de afmeting *aantalElementen*. Elk element is nul.

newMat()Catalogus > **newMat**(*aantalRijen*, *aantalKolommen*)
 $\Rightarrow$ *matrix*

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

Geeft een matrix met nullen met de afmeting *aantalRijen* bij *aantalKolommen*.

nfMax()Catalogus > **nfMax**(*Uitdr*, *Var*) $\Rightarrow$ waarde

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

nfMax(*Uitdr*, *Var*, *ondergrens*) $\Rightarrow$ waarde

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

nfMax(*Uitdr*, *Var*, *ondergrens*, *bovengrens*) $\Rightarrow$ waarde**nfMax**(*Uitdr*, *Var*) | *ondergrens* $\leq$ *Var* $\leq$ *bovengrens* $\Rightarrow$ waarde

Geeft een mogelijke numerieke waarde van variabele *Var* waarvoor het lokale maximum van *Uitdr* optreedt.

Als u *ondergrens* en *bovengrens* opgeeft, zoekt de functie binnen het gesloten interval [*ondergrens*, *bovengrens*] naar het lokale maximum.

Opmerking: Zie ook **fMax()** en **d()**.**nfMin()**Catalogus > **nfMin**(*Uitdr*, *Var*) $\Rightarrow$ waarde

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

nfMin(*Uitdr*, *Var*, *ondergrens*) $\Rightarrow$ waarde

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

nfMin(*Uitdr*, *Var*, *ondergrens*, *bovengrens*) $\Rightarrow$ waarde**nfMin**(*Uitdr*, *Var*) | *ondergrens* $\leq$ *Var* $\leq$ *bovengrens* $\Rightarrow$ waarde

Geeft een mogelijke numerieke waarde van variabele *Var* waarvoor het lokale minimum van *Uitdr* optreedt.

Als u *ondergrens* en *bovengrens* opgeeft, zoekt de functie binnen het gesloten interval [*ondergrens*, *bovengrens*] naar het lokale minimum.

Opmerking: zie ook **fMin()** en **d()**.**nlnt()**Catalogus > **nlnt**(*Uitdr1*, *Var*, *Onder*, *Boven*)
 $\Rightarrow$ uitdrukking

$\text{nlnt}\left(e^{-x^2}, x, -1, 1\right)$	1.49365
--	---------

Als de integrand *Uitdr1* geen andere variabele dan *Var* bevat, en als *Onder* en *Boven* constanten, positief ∞ of negatief ∞ zijn, dan geeft **nInt()** een benadering van $\int$ (*Uitdr1*, *Var*, *Onder*, *Boven*). Deze benadering is een gewogen gemiddelde van enkele steekproefwaarden van de integrand in het interval *Onder*<*Var*<*Boven*.

Het doel is zes significante cijfers. Het adaptieve algoritme eindigt wanneer het waarschijnlijk lijkt dat het doel is bereikt, of wanneer het onwaarschijnlijk lijkt dat extra steekproeven een lonende verbetering zullen opleveren.

Er wordt een waarschuwing weergegeven ("Twijfelachtige nauwkeurigheid") wanneer het erop lijkt dat het doel niet is bereikt.

Nest **nInt()** om meervoudige numerieke integratie uit te voeren. Integratiegrenzen kunnen afhangen van integratievariabelen erbuiten.

Opmerking: zie ook $\int()$, pag. 224.

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

$$\int_{-\pi}^{\pi+10^{-12}} \cos(x) dx \quad -\sin\left(\frac{1}{1000000000000}\right)$$

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

nom()

nom(effectiefPercentage, CpY) $\Rightarrow$ waarde

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

Financiële functie die het jaarlijkse effectieve rentepercentage *effectiefPercentage* naar een nominaal percentage converteert, waarbij *CpY* het aantal rentetermijnen per jaar is.

effectiefPercentage moet een reëel getal zijn en *CpY* moet een reëel getal > 0 zijn.

Opmerking: zie ook **eff()**, pag. 62.

nor (noch)

BooleaanseUitdr1 **nor** *BooleaanseUitdr2*
levert *Booleaanse uitdrukking*

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

BooleaanseLijst1 **nor** *BooleaanseLijst2*
levert *Booleaanse lijst*

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

BooleaanseMatrix1norBooleaanseMatrix2
 levert *Booleaanse matrix*

Geeft de ontkenning (negatie) van een logische **or** bewerking op de twee argumenten. Geeft waar, onwaar of een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

Geheel getal1norGeheel getal2⇒geheel getal

Vergelijkt twee reële gehele getallen bit voor bit met behulp van een **nor**-bewerking. Intern worden beide gehele getallen geconverteerd naar 64-bits binaire getallen met een teken (positief of negatief). Wanneer overeenkomstige bits vergeleken worden, is het resultaat 1 als beide bits 1 zijn; anders is het resultaat 0. De geretourneerde waarde representeert de bitresultaten, en wordt weergegeven volgens de ingestelde grondtal-modus.

U kunt de gehele getallen invoeren in elk talstelsel. Voor een binaire of hexadecimale invoer moet u respectievelijk het voorvoegsel 0b of 0h gebruiken. Zonder voorvoegsel worden gehele getallen behandeld als decimaal (grondtal 10).

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

norm()

Catalogus > 

norm(Matrix)⇒*uitdrukking*

norm(Vector)⇒*uitdrukking*

Geeft de Frobenius-norm.

$\text{norm}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$\sqrt{a^2+b^2+c^2+d^2}$
$\text{norm}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)$	$\sqrt{30}$
$\text{norm}\left(\begin{bmatrix} 1 & 2 \end{bmatrix}\right)$	$\sqrt{5}$
$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	$\sqrt{5}$

normalLine()Catalogus > 

normalLine(*Uitdr1*,*Var*,*Punt*)
 $\Rightarrow$ *uitdrukking*

$\text{normalLine}(x^2, x, 1)$	$\frac{3}{2} \cdot \frac{x}{2}$
--------------------------------	---------------------------------

normalLine(*Uitdr1*,*Var*=*Punt*)
 $\Rightarrow$ *uitdrukking*

$\text{normalLine}((x-3)^2 - 4, x, 3)$	$x=3$
--	-------

Geeft de normaal (loodrecht op de raaklijn) aan de kromme die gespecificeerd wordt door *Uitdr1* op het punt dat gespecificeerd is in *Var*=*Punt*.

$\text{normalLine}\left(\frac{1}{x^3}, x=0\right)$	0
--	---

$\text{normalLine}(\sqrt{ x }, x=0)$	undef
--------------------------------------	-------

Zorg ervoor dat de onafhankelijke variabele niet gedefinieerd is. Bijvoorbeeld: als *f1*(*x*):=5 en *x*: =3, dan geeft **normalLine**(*f1*(*x*),*x*,2) "false."

normCdf()Catalogus > 

normCdf(*ondergrens*,*bovengrens*[,*μ*[,*σ*]]) $\Rightarrow$ *getal* als *ondergrens* en *bovengrens* getallen zijn, *lijst* als *ondergrens* en *bovengrens* lijsten zijn

Berekent de normale verdelingskans tussen *ondergrens* en *bovengrens* voor de gespecificeerde *μ* (standaard=0) en *σ* (standaard=1).

Voor $P(X \leq \text{bovengrens})$ stelt u *ondergrens* = $-\infty$.

normPdf()Catalogus > 

normPdf(*XWaarde*[,*μ*[,*σ*]]) $\Rightarrow$ *getal* als *XWaarde* een getal is, *lijst* als *XWaarde* een lijst is

Berekent de kansdichtheidsfunctie voor de normale verdeling bij een gespecificeerde *XWaarde* voor de gespecificeerde *μ* en *σ*.

not (niet)Catalogus > 

not *BooleaanseUitdr* $\Rightarrow$ *Booleaanse uitdrukking*

$\text{not}(2 \geq 3)$	true
$\text{not}(x < 2)$	$x \geq 2$

Geeft waar, onwaar of een vereenvoudigde vorm van het argument.

not not innocent	<i>innocent</i>
---------------------------	-----------------

not *Geheel getal* $\Rightarrow$ *geheel getal*

In de Hex-grondtalmodus:

Belangrijk: nul, niet de letter O.

Geeft het één-complement van een reëel geheel getal. Intern wordt *Geheel getal* geconverteerd naar een 64-bits binair getal met een plus- of min-teken. De waarde van elke bit wordt omgewisseld (0 wordt 1 en andersom) voor het één-complement. Resultaten worden weergegeven volgens de grondtal-modus.

U kunt het gehele getal in elk grondtal invoeren. Voor een binaire of hexadecimale invoer moet u respectievelijk het prefix 0b of 0h gebruiken. Zonder prefix wordt het gehele getal behandeld als decimaal (grondtal 10).

Als u een decimaal geheel getal invoert dat te groot is voor een 64-bits binaire vorm met een teken (positief of negatief), dan wordt er een symmetrische modulo-bewerking gebruikt om de waarde binnen het betreffende bereik te brengen. Zie voor meer informatie ►Base2, pag. 18.

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

In de Bin-grondtalmodus:

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

Om het hele resultaat te zien drukt u op ► en gebruikt u vervolgens ◀ en ► om de cursor te verplaatsen.

Opmerking: een binaire invoer kan maximaal 64 cijfers hebben (het prefix 0b niet meegeteld). Een hexadecimale invoer kan maximaal 16 cijfers hebben.

nPr(Uitdr1, Uitdr2)⇒uitdrukking

Voor geheel getal *Uitdr1* en *Uitdr2* met $Uitdr1 \geq Uitdr2 \geq 0$, is **nPr()** het aantal permutaties van *Uitdr1* dingen die met *Uitdr2* keer tegelijk zijn genomen. Beide argumenten kunnen gehele getallen of symbolische uitdrukkingen zijn.

nPr(Uitdr, 0)⇒1

nPr(Uitdr, negGeheel getal)⇒ 1/ ((Uitdr+1) · (Uitdr+2) ... (uitdrukking-negGeheel getal))

nPr(Uitdr, posGeheel getal)⇒ Uitdr · (Uitdr-1) ... (Uitdr-posGeheel getal+1)

nPr(Uitdr, nietGeheel getal)⇒ Uitdr! / (Uitdr-nietGeheel getal)!

nPr(Lijst1, Lijst2)⇒lijst

$nPr(z, 3)$	$z \cdot (z-2) \cdot (z-1)$
$Ans z=5$	60
$nPr(z, -3)$	$\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$
$nPr(z, c)$	$\frac{z!}{(z-c)!}$
$Ans \cdot nPr(z-c, -c)$	1

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

Geeft een lijst met permutaties op basis van de overeenkomstige elementparen in de twee lijsten. De argumenten moeten lijsten van dezelfde afmeting zijn.

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

Geeft een matrix met permutaties op basis van de overeenkomstige elementparen in de twee matrices. De argumenten moeten matrices van dezelfde afmeting zijn.

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

npv(*RentePercentage*, *CFO*, *CFLijst* [, *CFFreq*])

Financiële functie die de netto contante waarde berekent; de som van de contante waarden voor de kasinstromen en -uitstromen. Een positief resultaat voor npv duidt op een winstgevende investering.

RentePercentage is de rente waarmee de cashflows verdisconteerd moeten worden (de kosten van het geld) over één periode.

CFO is de begin-cashflow op tijdstip 0; dit moet een reëel getal zijn.

CFLijst is een lijst met cashflow-bedragen na de begin-cashflow *CFO*.

CFFreq is een lijst waarin elk element de frequentie waarmee een gegroepeerde (opeenvolgend) cashflow-bedrag voorkomt specificeert; dit is het overeenkomstige element van *CFLijst*. De standaardwaarde is 1; als u waarden invoert, dan moeten dit positieve gehele getallen < 10.000 zijn.

$$\begin{aligned} \text{list1} &:= \{6000, -8000, 2000, -3000\} \\ &\quad \{6000, -8000, 2000, -3000\} \\ \text{list2} &:= \{2, 2, 2, 1\} \quad \{2, 2, 2, 1\} \\ \text{npv}(10, 5000, \text{list1}, \text{list2}) &\quad 4769.91 \end{aligned}$$

nSolve(*Vergelijking*, *Var*[=*Gok*]) $\Rightarrow$ *getal of fout_string*

nSolve(*Vergelijking*, *Var* [=*Gok*], *ondergrens*) $\Rightarrow$ *getal of fout_string*

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

nSolve(*Vergelijking*, *Var*
[*=Gok*], *ondergrens*, *bovengrens*) $\Rightarrow$ *getal* of
fout_string

nSolve(*Vergelijking*, *Var*[*=Gok*]) |
ondergrens $\leq$ *Var* $\leq$ *bovengrens* $\Rightarrow$ *getal* of
fout_string

Zoekt iteratief naar één benaderende numerieke oplossing van *Vergelijking*, voor de ene variabele ervan. Specificeer de variabele als:

variabele

– of –

variabele = *reëel getal*

Bijvoorbeeld: x is geldig en x=3 ook.

nSolve() is vaak veel sneller dan solve() of zeros(), vooral als de “|”-operator gebruikt wordt om de zoekactie te beperken tot een klein interval met exact één eenvoudige oplossing.

nSolve() probeert één punt te bepalen waarop het residu nul is, of twee relatief dicht bij elkaar liggende punten waarop het residu tegenovergestelde tekens heeft, en de grootte van het residu niet overdreven is. Als dit niet bereikt kan worden met behulp van een bescheiden aantal steekproefpunten, dan wordt de string “geen oplossing gevonden” gegeven.

Opmerking: zie ook cSolve(), cZeros(), solve() en zeros().

Opmerking: als er meerdere oplossingen zijn, dan kunt u een gok gebruiken om een bepaalde oplossing te helpen vinden.

$\text{nSolve}(x^2 + 5 \cdot x - 25 = 9, x) x < 0$	-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*], *X1*, [*Freq*], [*Categorie*, *Opnemen*]]

OneVar [*n*], *X1*, *X2* [*X3* [...], *X20*]]

Berekent statistieken voor één variabele op maximaal 20 lijsten. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

De *X*-argumenten zijn gegevenslijsten.

Freq is een optionele lijst met frequentiewaarden.

Elk element in *Freq* specificeert de frequentie waarmee elke overeenkomstige *X*-waarde voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met numerieke categoriecodes voor de overeenkomstige *X*-waarden.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Een leeg element in een van de lijsten *X*, *Freq* of *Categorie* resulteert in een lege plaats voor het overeenkomstige element in al deze lijsten. Een leeg element in een van de lijsten *X1* tot en met *X20* resulteert in een lege plaats voor het overeenkomstige element in al deze lijsten. Voor meer informatie over lege elementen, zie pag. 267.

Uitvoervariabele	Beschrijving
stat. $\bar{x}$	Gemiddelde van de <i>x</i> -waarden
stat. Σx	Som van de <i>x</i> -waarden
stat. Σx^2	Som van de x^2 -waarden
stat. <i>sx</i>	Steekproef-standaarddeviatie van <i>x</i>
stat. <i>x</i>	Populatie-standaarddeviatie van <i>x</i>
stat. <i>n</i>	Aantal gegevens
stat.Min <i>X</i>	Minimum van de <i>x</i> -waarden
stat. $Q_1 X$	1ste kwartiel van <i>x</i>
stat.Median <i>X</i>	Mediaan van <i>x</i>
stat. $Q_3 X$	3de kwartiel van <i>x</i>

Uitvoervariabele	Beschrijving
stat.MaxX	Maximum van de x-waarden
stat.SSX	Som van de kwadraten van de afwijkingen ten opzichte van het gemiddelde van x

or (of)

Catalogus > 

BooleaanseUitdr1 **or** *BooleaanseUitdr2*
levert *Booleaanse uitdrukking*

$x \geq 3$ or $x \geq 4$ $x \geq 3$

BooleaanseLijst1 **or** *BooleaanseLijst2*
levert *Booleaanse lijst*

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

If $x \leq 0$ or $x \geq 5$

Goto *end*

Return $x \cdot 3$

Lbl *end*

EndFunc

BooleaanseMatrix1 **or** *BooleaanseMatrix2*
levert *Booleaanse matrix*

Geeft waar of onwaar of een
vereenvoudigde vorm van de
oorspronkelijke invoer.

$g(3)$ 9

$g(0)$ *A function did not return a value*

Geeft waar als een van beide of beide
uitdrukkingen uitgewerkt worden tot waar.
Geeft alleen onwaar als beide
uitdrukkingen uitgewerkt worden tot
onwaar.

Opmerking: zie *xor*.

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Geheel getal1 **or** *Geheel getal2* $\Rightarrow$ *geheel getal*

In de Hex-grondtalmodus:

0h7AC36 or 0h3D5F 0h7BD7F

Vergelijkt twee reële gehele getallen bit voor bit met behulp van een or-bewerking. Intern worden beide gehele getallen geconverteerd naar 64-bits binaire getallen met een plus- of min-teken. Wanneer overeenkomstige bits vergeleken worden, is het resultaat 1 als een van beide of beide bits 1 zijn; het resultaat is alleen 0 als beide bits 0 zijn. De geretourneerde waarde geeft de bitresultaten, en wordt weergegeven volgens de grondtal-modus.

Belangrijk: nul, niet de letter O.

In de Bin-grondtalmodus:

0b100101 or 0b100 0b100101

Opmerking: een binaire invoer kan maximaal 64 cijfers hebben (het prefix 0b niet meegeteld). Een hexadecimale invoer kan maximaal 16 cijfers hebben.

U kunt de gehele getallen invoeren in elk grondtal. Voor een binaire of hexadecimale invoer moet u respectievelijk het prefix 0b of 0h gebruiken. Zonder prefix worden gehele getallen behandeld als decimaal (grondtal 10).

Als u een decimaal geheel getal invoert dat te groot is voor een 64-bits binaire vorm met een teken (positief of negatief), dan wordt er een symmetrische modulo-bewerking gebruikt om de waarde binnen het betreffende bereik te brengen. Zie voor meer informatie ►Base2, pag. 18.

Opmerking: zie xor.

ord()

ord(String) ⇒ *geheel getal*

ord(Lijst l) ⇒ *lijst*

Geeft de numerieke code van het eerste teken in tekenreeks *String*, of een lijst van de eerste tekens van elk lijstelement.

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

P**P►Rx()**

P►Rx(rUitdr, θUitdr) ⇒ *uitdrukking*

P►Rx(rLijst, θLijst) ⇒ *lijst*

P►Rx(rMatrix, θMatrix) ⇒ *matrix*

Geeft de equivalente x-coördinaat van het (r, θ)-paar.

Opmerking: het θ-argument wordt geïnterpreteerd als een hoek in graden, decimale graden of radialen, volgens de ingestelde hoekmodus. Als het argument een uitdrukking is, dan kunt u °, °G of °r gebruiken om de hoekmodusinstelling tijdelijk te onderdrukken.

In de hoekmodus Radialen:

P►Rx(r, θ)	$\cos(\theta) \cdot r$
P►Rx(4, 60°)	2
P►Rx($\left\{-3, 10, 1.3\right\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}\right)$	$\left\{\frac{-3}{2}, 5\sqrt{2}, 1.3\right\}$

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **P@>Rx (...)** in te typen.

P►Ry(*rUitdr*, *θUitdr*)⇒*uitdrukking*

In de hoekmodus Radialen:

P►Ry(*rLijst*, *θLijst*)⇒*lijst*

P►Ry(*rMatrix*, *θMatrix*)⇒*matrix*

Geeft het equivalente y-coördinaat van het (r, θ)-paar.

Opmerking: het θ-argument wordt geïnterpreteerd als een hoek in graden, decimale graden of radialen, volgens de ingestelde hoekmodus. Als het argument een uitdrukking is, dan kunt u °, °, of r gebruiken om de hoekmodusinstelling tijdelijk te onderdrukken.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **P@>Ry (...)** in te typen.

$P \blacktriangleright Ry(r, \theta)$	$\sin(\theta) \cdot r$
$P \blacktriangleright Ry(4, 60^\circ)$	$2 \cdot \sqrt{3}$
$P \blacktriangleright Ry\left(\{-3, 10, 1.3\}, \left\{\frac{\pi}{3}, \frac{\pi}{4}, 0\right\}\right)$	$\left\{\frac{-3 \cdot \sqrt{3}}{2}, -5 \cdot \sqrt{2}, 0\right\}$

PassErr

Zie voor een voorbeeld van **PassErr** Voorbeeld 2 onder het commando **Try** (pag. 203).

Brengt een fout naar het volgende niveau.

Als systeemvariabele *errCode* nul is, dan doet **PassErr** niets.

De **Else**-zin van het **Try...Else...EndTry**-blok moet **ClrErr** of **PassErr** gebruiken. Als de fout verwerkt of genegeerd moet worden, gebruik dan **ClrErr**. Als onbekend is wat er met de fout gedaan moet worden, gebruik dan **PassErr** om hem te verzenden naar de volgende foutenafhandelaar. Als er geen onbesliste **Try...Else...EndTry**-foutenafhandelaars meer zijn, wordt het foutendialoogvenster weergegeven zoals normaal is.

Opmerking: zie ook **ClrErr**, pag. 27 en **Try**, pag. 203.

Opmerking bij het invoeren van het voorbeeld: in de toepassing Rekenmachine op de rekenmachine kunt u definities van meerdere regels invoeren door op  in plaats van op  te drukken aan het eind van iedere regel. Op het toetsenbord van de computer houdt u **Alt** ingedrukt en drukt u op **Enter**.

piecewise()

Catalogus > 

piecewise(*Uitdr1* [, *Condl* [, *Uitdr2* [, *Cond2* [, ...]]]])

Geeft definities van een stuksgewijs gedefinieerde functie in de vorm van een lijst. U kunt ook stuksgewijs gedefinieerde functies creëren met behulp van een template.

Define $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$	<i>Done</i>
$p(1)$	1
$p(-1)$	undef

Opmerking: zie ook **Stuksgewijs gedefinieerde functie-template**, pag. 3.

poissCdf()

Catalogus > 

poissCdf(λ , *ondergrens*, *bovengrens*) $\Rightarrow$ *getal* als *ondergrens* en *bovengrens* getallen zijn, *lijst* als *ondergrens* en *bovengrens* lijsten zijn

poissCdf(λ , *bovengrens*) (voor $P(0 \leq X \leq \text{bovengrens}) \Rightarrow$ *getal* als *bovengrens* een getal is, *lijst* als *bovengrens* een lijst is

Berekent een cumulatieve kans voor de discrete Poisson-verdeling met het gespecificeerde gemiddelde λ .

Voor $P(X \leq \text{bovengrens})$ stelt u *ondergrens*=0 in

poissPdf()

Catalogus > 

poissPdf(λ , *XWaarde*) $\Rightarrow$ *getal* als *XWaarde* een getal is, *lijst* als *XWaarde* een lijst is

Berekent een kans voor de discrete Poisson-verdeling met het gespecificeerde gemiddelde λ .

► Polar

Vector ► Polar

Geeft *vector* weer in polaire vorm $[r \angle \theta]$. De vector moet de afmeting 2 hebben en kan een rij of een kolom zijn.

Opmerking: ►Polar is een weergave-opmaakinstructie, geen conversiefunctie. U kunt dit commando alleen gebruiken op het eind van een invoerregel, en *ans* wordt niet bijgewerkt.

Opmerking: zie ook ►Rect, pag. 155.

complexeWaarde ►Polar

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **e>Polar** in te typen.

Geeft *complexeWaarde* in polaire vorm weer.

- De hoekmodus Graden geeft $(r \angle \theta)$.
- De hoekmodus Radialen geeft $re^{i\theta}$.

complexeWaarde kan elke complexe vorm hebben. Een $re^{i\theta}$ -invoer veroorzaakt echter een fout in de hoekmodus Graden.

Opmerking: u moet haakjes gebruiken voor een $(r \angle \theta)$ polaire invoer.

$$\begin{array}{l} [1 \ 3.] \text{►Polar} \quad [3.16228 \ \angle 1.24905] \\ [x \ y] \text{►Polar} \quad \left[\sqrt{x^2+y^2} \ \angle \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1} \left(\frac{x}{y} \right) \right] \end{array}$$

In de hoekmodus Radialen:

$$\begin{array}{l} (3+4i) \text{►Polar} \quad e^{i \cdot \left(\frac{\pi}{2} - \tan^{-1} \left(\frac{3}{4} \right) \right)} .5 \\ \left(4 \angle \frac{\pi}{3} \right) \text{►Polar} \quad e^{\frac{i \cdot \pi}{3}} .4 \end{array}$$

In de hoekmodus Decimale graden:

$$(4i) \text{►Polar} \quad (4 \angle 100.)$$

In de hoekmodus Graden

$$(3+4i) \text{►Polar} \quad \left(5 \angle 90 - \tan^{-1} \left(\frac{3}{4} \right) \right)$$

polyCoeffs()

polyCoeffs(Poly [,Var])⇒lijst

Geeft een lijst van de coëfficiënten van veelterm (polynoom) *Poly* ten opzichte van variabele *Var*.

$$\begin{array}{l} \text{polyCoeffs}(4x^2-3x+2,x) \quad \{4,-3,2\} \\ \text{polyCoeffs}((x-1)^2 \cdot (x+2)^3) \quad \{1,4,1,-10,-4,8\} \end{array}$$

Poly moet een veeltermuitdrukking in *Var* zijn. Wij adviseren om *Var* niet weg te laten, tenzij *Poly* een uitdrukking in een enkele variabele is.

Werkt de veelterm uit en selecteert *x* voor de weggelaten *Var*.

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

polyDegree()

polyDegree(*Poly* [, *Var*]) $\Rightarrow$ waarde

Geeft de graad van de veeltermuitdrukking *Poly* ten opzichte van variabele *Var*. Als u *Var* weglaat, dan selecteert de functie **polyDegree()** een standaardwaarde uit de variabelen in de veelterm *Poly*.

Poly moet een veeltermuitdrukking in *Var* zijn. Wij adviseren om *Var* niet weg te laten, tenzij *Poly* een uitdrukking in een enkele variabele is.

$\text{polyDegree}(5)$	0
$\text{polyDegree}(\ln(2) + \pi, x)$	0

Constante veeltermen

$\text{polyDegree}(4 \cdot x^2 - 3 \cdot x + 2, x)$	2
$\text{polyDegree}((x-1)^2 \cdot (x+2)^3)$	5

$\text{polyDegree}\left(\left(x+y^2+z^3\right)^2, x\right)$	2
$\text{polyDegree}\left(\left(x+y^2+z^3\right)^2, y\right)$	4
$\text{polyDegree}\left((x-1)^{10000}, x\right)$	10000

De graad kan geëxtraheerd worden, ook al kunnen de coëfficiënten niet geëxtraheerd worden. Dit komt doordat de graad geëxtraheerd kan worden zonder de veelterm uit te breiden.

polyEval()Catalogus > **polyEval(Lijst1, Uitdr1)**⇒uitdrukking

$\text{polyEval}(\{a, b, c\}, x)$	$a \cdot x^2 + b \cdot x + c$
-----------------------------------	-------------------------------

polyEval(Lijst1, Lijst2)⇒uitdrukking

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

Interpreteert het eerste argument als de coëfficiënt van een veelterm met aflopende machten, en geeft de veelterm uitgewerkt voor de waarde van het tweede argument.

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

polyGcd()Catalogus > **polyGcd(Uitdr1, Uitdr2)**⇒uitdrukking

Geeft de grootste gemene deler van de twee argumenten.

Uitdr1 en *Uitdr2* moeten veeltermuitdrukkingen zijn.

Lijst-, matrix- en Booleaanse argumenten zijn niet toegestaan.

$\text{polyGcd}(100, 30)$	10
---------------------------	----

$\text{polyGcd}(x^2 - 1, x - 1)$	$x - 1$
----------------------------------	---------

$\text{polyGcd}(x^3 - 6 \cdot x^2 + 11 \cdot x - 6, x^2 - 6 \cdot x + 8)$	$x - 2$
---	---------

polyQuotient()Catalogus > **polyQuotient(Poly1, Poly2 [, Var])**
⇒uitdrukking

Geeft het quotiënt van veelterm *Poly1* gedeeld door veelterm *Poly2* ten opzichte van de gespecificeerde variabele *Var*.

Poly1 en *Poly2* moeten veeltermuitdrukkingen in *Var* zijn. Wij adviseren om *Var* niet weg te laten, tenzij *Poly1* en *Poly2* uitdrukkingen in dezelfde enkele variabele zijn.

$\text{polyQuotient}(x - 1, x - 3)$	1
-------------------------------------	---

$\text{polyQuotient}(x - 1, x^2 - 1)$	0
---------------------------------------	---

$\text{polyQuotient}(x^2 - 1, x - 1)$	$x + 1$
---------------------------------------	---------

$\text{polyQuotient}(x^3 - 6 \cdot x^2 + 11 \cdot x - 6, x^2 - 6 \cdot x + 8)$	x
--	-----

$\text{polyQuotient}((x - y) \cdot (y - z), x + y + z, x)$	$y - z$
--	---------

$\text{polyQuotient}((x - y) \cdot (y - z), x + y + z, y)$	$2 \cdot x - y + 2 \cdot z$
--	-----------------------------

$\text{polyQuotient}((x - y) \cdot (y - z), x + y + z, z)$	$-(x - y)$
--	------------

polyRemainder()Catalogus > **polyRemainder**(*Poly1*, *Poly2* [, *Var*]) $\Rightarrow$ *uitdrukking*

Geeft de rest van veelterm *Poly1* gedeeld door veelterm *Poly2* ten opzichte van de gespecificeerde variabele *Var*.

Poly1 en *Poly2* moeten veeltermuitdrukkingen in *Var* zijn. Wij adviseren om *Var* niet weg te laten, tenzij *Poly1* en *Poly2* uitdrukkingen in dezelfde enkele variabele zijn.

$\text{polyRemainder}(x-1, x-3)$	2
$\text{polyRemainder}(x-1, x^2-1)$	$x-1$
$\text{polyRemainder}(x^2-1, x-1)$	0
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, x)$	$-(y-z) \cdot (2 \cdot y+z)$
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, y)$	$-2 \cdot x^2 - 5 \cdot x \cdot z - 2 \cdot z^2$
$\text{polyRemainder}((x-y) \cdot (y-z), x+y+z, z)$	$(x-y) \cdot (x+2 \cdot y)$

polyRoots()Catalogus > **polyRoots**(*Poly*, *Var*) $\Rightarrow$ *lijst***polyRoots**(*LijstVanCoëff*) $\Rightarrow$ *lijst*

De eerste syntax, **polyRoots**(*Poly*, *Var*), geeft een lijst met reële oplossingen van de veelterm *Poly* voor de variabele *Var*. Geeft een lege lijst als er geen reële oplossingen bestaan: { }.

Poly moet een veelterm met één variabele zijn.

De tweede syntax, **polyRoots**(*LijstVanCoëff*), geeft een lijst met reële oplossingen voor de coëfficiënten in *LijstVanCoëff*.

Opmerking: zie ook **cPolyRoots()**, pag. 38.

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

PowerRegCatalogus > **PowerReg** *X*, *Y* [, *Freq*] [, *Categorie*, *Opnemen*]

Berekent de machtsregressie $y = (a \cdot x)^b$ op de lijsten *X* en *Y* met frequentie *Freq*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en *Y* zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden.

Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig *X*- en *Y*-gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige *X*- en *Y*-gegevens.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a \cdot (x)^b$
stat.a, stat.b	Regressiecoëfficiënten
stat.r ²	Coëfficiënt van lineaire determinatie voor getransformeerde gegevens
stat.r	Correlatiecoëfficiënt voor getransformeerde gegevens ($\ln(x)$, $\ln(y)$)
stat.Resid	Residuen die geassocieerd zijn met het machtsmodel
stat.ResidTrans	Residuen die geassocieerd zijn met de lineaire regressie van getransformeerde gegevens
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

Prgm
Blok

Bereken GCD (grootste gemene deler) en geef tussenresultaten weer.

EndPrgm

Template voor het creëren van een door de gebruiker gedefinieerd programma. Moet gebruikt worden met het commando **Define**, **Define LibPub** of **Define LibPriv**.

Blok kan een enkele bewering of een serie beweringen zijn die gescheiden worden door het teken *“:”*, of een serie beweringen op aparte regels.

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

```

Define proggcd(a,b)=Prgm
  Local d
  While b≠0
    d:=mod(a,b)
    a:=b
    b:=d
  Disp a," ",b
  EndWhile
  Disp "GCD=",a
  EndPrgm
  
```

Done

proggcd(4560,450)	
	450 60
	60 30
	30 0
	GCD=30
	Done

product(Lijst[, Start[, Eind]])
 ⇒*uitdrukking*

Geeft het product van de elementen in *Lijst*. *Start* en *Eind* zijn optioneel. Ze specificeren een bereik van elementen.

product(MatrixI[, Start[, Eind]])⇒*matrix*

Geeft een rijvector met de producten van de elementen in de kolommen van *MatrixI*. *Start* en *Eind* zijn optioneel. Ze specificeren een bereik van rijen.

product({1,2,3,4})	24
product({2,x,y})	2·x·y
product({4,5,8,9},2,3)	40

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

Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

propFrac()

propFrac(*Uitdr1*[, *Var*]) $\Rightarrow$ *uitdrukking*

propFrac(*rationaal_getal*) geeft *rationaal_getal* als de som van een geheel getal en een breuk die hetzelfde teken hebben, en waarbij de noemer groter is dan de teller.

propFrac(*rationale_uitdrukking*, *Var*) geeft de som van echte breuken en een veelterm ten opzichte van *Var*. De graad van *Var* in de noemer is groter dan de graad van *Var* in de teller in elke echte breuk. Gelijke machten van *Var* worden samengenomen. De termen en hun factoren worden gesorteerd met *Var* als de hoofdvariabele.

Als *Var* wordt weggelaten, dan wordt een uitbreiding naar een echte breuk uitgevoerd ten opzichte van de belangrijkste hoofdvariabele. De coëfficiënten van het veeltermdeel worden vervolgens eerst echt gemaakt ten opzichte van hun belangrijkste hoofdvariabele, en zo verder.

Bij rationale uitdrukkingen is **propFrac()** een sneller maar minder extreem alternatief voor **expand()**.

U kunt de functie **propFrac()** gebruiken om gemengde breuken te representeren en om het optellen en aftrekken van gemengde breuken te demonstreren.

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

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

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

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

$$\text{propFrac}\left(\frac{11}{7}\right) \quad 1 + \frac{4}{7}$$

$$\text{propFrac}\left(3 + \frac{1}{11} + 5 + \frac{3}{4}\right) \quad 8 + \frac{37}{44}$$

$$\text{propFrac}\left(3 + \frac{1}{11} - \left(5 + \frac{3}{4}\right)\right) \quad -2 - \frac{29}{44}$$

QR

QR *Matrix*, *qMatrix*, *rMatrix*[, *Tol*]

Berekent de Householder QR-ontbinding van een reële of complexe matrix. De resulterende Q- en R-matrices worden opgeslagen in de gespecificeerde *Matrix*. De Q-matrix is unitair. De R-matrix is bovendreiehoeks.

Optioneel wordt elk matrixelement behandeld als nul als de absolute waarde ervan minder dan *Tol* is. Deze tolerantie wordt alleen gebruikt als de matrix gegevens met een drijvende komma heeft, en geen symbolische variabelen bevat die geen waarde toegekend hebben gekregen. Anders wordt *Tol* genegeerd.

- Als u   gebruikt of de modus **Automatisch of Benaderend** instelt op Benaderend, dan worden berekeningen met behulp van de drijvende komma uitgevoerd.
- Als *Tol* wordt weggelaten of niet wordt gebruikt, dan wordt de standaardtolerantie berekend als:
 $5E-14 \cdot \max(\dim(\text{Matrix})) \cdot \text{rowNorm}(\text{Matrix})$

De QR-ontbinding wordt numeriek berekend met behulp van Householder-transformaties. De symbolische oplossing wordt berekend met behulp van Gram-Schmidt. De kolommen in *qMatNaam* zijn de orthonormale basisvectoren die de ruimte die gedefinieerd wordt door *matrix* omspannen.

Het getal met drijvende komma (9.) in *m1* zorgt ervoor dat de resultaten worden berekend in drijvende-kommavorm.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix}$
QR <i>m1,qm,rm</i> Done	
<i>qm</i>	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$
<i>rm</i>	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

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

QuadReg $X, Y [, Freq] [, Categorie, Opnemen]$

Berekent de kwadratische veeltermregressie $y = a \cdot x^2 + b \cdot x + c$ op de lijsten X en Y met frequentie $Freq$. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en Y zijn lijsten met onafhankelijke en afhankelijke variabelen.

$Freq$ is een optionele lijst met frequentiewaarden. Elk element in $Freq$ specificeert de frequentie waarmee elk overeenkomstig X - en Y -gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige X - en Y -gegevens.

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a \cdot x^2 + b \cdot x + c$
stat.a, stat.b, stat.c	Regressiecoëfficiënten
stat.R ²	Determinatiecoëfficiënt
stat.Resid	Residuen uit de regressie
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

QuartReg $X, Y [, Freq] [, Categorie, Opnemen]$

Berekent de vierdegraads veeltermregressie $y = a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$ op de lijsten X en Y met frequentie $Freq$. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en Y zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig X - en Y -gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige X - en Y -gegevens..

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a \cdot x^4 + b \cdot x^3 + c \cdot x^2 + d \cdot x + e$
stat.a, stat.b, stat.c, stat.d, stat.e	Regressiecoëfficiënten
stat.R ²	Determinatiecoëfficiënt
stat.Resid	Residuen uit de regressie
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

R ► Pθ()**Catalogus >****R ► Pθ** (*xUitdr*, *yUitdr*) ⇒ *uitdrukking*

In de hoekmodus Graden:

R ► Pθ (*xLijst*, *yLijst*) ⇒ *lijst***R ► Pθ** (*xMatrix*, *yMatrix*) ⇒ *matrix*

$$\text{R ► P}\theta(x,y) \quad 90 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Geeft de equivalente θ-coördinaat van het (x,y)-paar argumenten.

In de hoekmodus Decimale graden:

Opmerking: de uitkomst wordt in graden, decimale graden of radialen gegeven, volgens de ingestelde hoekmodus.

$$\text{R ► P}\theta(x,y) \quad 100 \cdot \text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Opmerking: u kunt deze operator invoeren vanaf het toetsenbord van de computer door **R@>Ptheta (...)** in te typen.

In de hoekmodus Radialen:

$$\begin{aligned} \text{R ► P}\theta(3,2) & \quad \tan^{-1}\left(\frac{2}{3}\right) \\ \text{R ► P}\theta\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right) & \quad \begin{bmatrix} 0 & \tan^{-1}\left(\frac{16}{\pi}\right) + \frac{\pi}{2} & 0.643501 \end{bmatrix} \end{aligned}$$

R ► Pr()**Catalogus >****R ► Pr** (*xUitdr*, *yUitdr*) ⇒ *uitdrukking*

In de hoekmodus Radialen:

R ► Pr (*xLijst*, *yLijst*) ⇒ *lijst***R ► Pr** (*xMatrix*, *yMatrix*) ⇒ *matrix*

$$\begin{aligned} \text{R ► Pr}(3,2) & \quad \sqrt{13} \\ \text{R ► Pr}(x,y) & \quad \sqrt{x^2 + y^2} \end{aligned}$$

Geeft de equivalente r-coördinaat van het (x,y)-paar argumenten.

$$\begin{aligned} \text{R ► Pr}\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right) & \quad \begin{bmatrix} 3 & \frac{\sqrt{\pi^2 + 256}}{4} & 2.5 \end{bmatrix} \end{aligned}$$

Opmerking: u kunt deze operator invoeren vanaf het toetsenbord van de computer door **R@>Pr (...)** in te typen.

► Rad**Catalogus >****Uitdr1 ► Rad** ⇒ *uitdrukking*

In de hoekmodus Graden:

Converteert het argument naar radialen.

$$(1.5) \text{ ► Rad} \quad (0.02618)^r$$

Opmerking: u kunt deze operator invoeren vanaf het toetsenbord van de computer door **@>Rad** in te typen.

In de hoekmodus Decimale graden:

$(1.5)^r$ ► Rad $(0.023562)^r$ **rand()**Catalogus > **rand()** ⇒ *uitdrukking***rand(*AantalPogingen*)** ⇒ *lijst***rand()** geeft een willekeurige waarde tussen 0 en 1.**rand(*AantalPogingen*)** geeft een lijst met *AantalPogingen* willekeurige waarden tussen 0 en 1.

Stelt de seed van het willekeurige getal in.

RandSeed 1147

Done

rand(2)

{ 0.158206, 0.717917 }

randBin()Catalogus > **randBin(*n*, *p*)** ⇒ *uitdrukking***randBin(*n*, *p*, *AantalPogingen*)** ⇒ *lijst***randBin(*n*, *p*)** geeft een willekeurig reëel getal uit een opgegeven binomiale verdeling.**randBin(*n*, *p*, *AantalPogingen*)** geeft een lijst met *AantalPogingen* willekeurige reële getallen uit een opgegeven binomiale verdeling.

randBin(80,0.5)

42

randBin(80,0.5,3)

{ 41, 32, 39 }

randInt()Catalogus > **randInt****(
ondergrens
,bovgrens)** ⇒
*uitdrukking***randInt
(
ondergrens
,bovgrens
,*AantalPogingen*)**
⇒ *lijst*

randInt(3,10)

5

randInt(3,10,4)

{ 9, 7, 5, 8 }

randInt

(
ondergrens
,bovengrens) geeft
 een willekeurig
 geheel getal binnen
 het bereik dat wordt
 bepaald door de
 gehele
 getallen *ondergrens*
 en *bovengrens*.

randInt

(
ondergrens
,bovengrens
,AantalPogingen)
 geeft een lijst met
AantalPogingen van
 willekeurige gehele
 getallen binnen het
 opgegeven bereik.

randMat()

randMat(aantalRijen, aantalKolommen)
 $\Rightarrow$ matrix

Geeft een matrix met gehele getallen
 tussen -9 en 9 met de gespecificeerde
 afmeting.

Beide argumenten moeten vereenvoudigen
 tot gehele getallen.

RandSeed 1147	Done		
randMat(3,3)	8	-3	6
	-2	3	-6
	0	4	-6

Opmerking: de waarden in deze matrix
 veranderen elke keer dat u op **enter** drukt.

randNorm()

randNorm(μ , σ) $\Rightarrow$ uitdrukking
randNorm(μ , σ , AantalPogingen) $\Rightarrow$ lijst

randNorm(μ , σ) geeft een decimaal getal
 uit de gespecificeerde normale verdeling.
 Dit kan elk reëel getal zijn, maar het zal
 sterk geconcentreerd zijn in het interval
 $[\mu-3\cdot\sigma, \mu+3\cdot\sigma]$.

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

randNorm()

randNorm(μ , σ , *AantalPogingen*) geeft een lijst met *AantalPogingen* decimale getallen uit de gespecificeerde normale verdeling.

randPoly()

randPoly(*Var*, *Orde*) $\Rightarrow$ uitdrukking

Geeft een veelterm in *Var* van de gespecificeerde *Orde*. De coëfficiënten zijn willekeurige gehele getallen in het bereik $-9 \text{ t/m } 9$. De leidende coëfficiënt is niet nul.

Orde moet 0–99 zijn.

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

randSamp()

randSamp(*Lijst*, *AantalPogingen* [*geenTerugl*]) $\Rightarrow$ lijst

Geeft een lijst met een willekeurige steekproef van *AantalPogingen* uit *Lijst* met een optie voor met teruglegging (*geenTerugl*=0), of zonder teruglegging (*geenTerugl*=1). De standaardinstelling is met teruglegging.

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

RandSeed

RandSeed *Getal*

Als *Getal* = 0, dan worden de seedwaarden ingesteld op de fabrieksinstellingen voor de generator van toevalsgetallen. Als *Getal* $\neq 0$, dan wordt dit commando gebruikt om twee seedwaarden te genereren, die worden opgeslagen in systeemvariabelen seed1 en seed2.

RandSeed 1147	Done
rand()	0.158206

real()

Catalogus > 

real(Uitdr1) ⇒ uitdrukking

Geeft het reële deel van het argument.

Opmerking: alle ongedefinieerde variabelen worden behandeld als reële variabelen. Zie ook **imag()**, page 94.

real(Lijst1) ⇒ lijst

Geeft de reële delen van alle elementen.

real(Matrix1) ⇒ matrix

Geeft de reële delen van alle elementen.

$\text{real}(2+3 \cdot i)$	2
$\text{real}(z)$	z
$\text{real}(x+i \cdot y)$	x

$$\text{real}(\{a+i \cdot b, 3, i\}) \quad \{a, 3, 0\}$$

$$\text{real}\left(\begin{bmatrix} a+i \cdot b & 3 \\ c & i \end{bmatrix}\right) \quad \begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$$

► Rect

Catalogus > 

Vector ► Rect

Opmerking: u kunt deze operator invoeren vanaf het toetsenbord van de computer door **@>Rect** in te typen.

Geeft **Vector** in rechthoekige vorm $[x, y, z]$. De vector moet de afmeting 2 of 3 hebben en kan een rij of een kolom zijn.

Opmerking: **► Rect** is een instructie voor het weergave-format, geen conversiefunctie. U kunt dit commando alleen gebruiken aan het eind van een invoerregel, en *ans* wordt niet bijgewerkt.

Opmerking: zie ook **► Polair**, pag. 141.

complexeWaarde ► Rect

Geeft **complexeWaarde** weer in rechthoekige vorm $a+bi$. De **complexeWaarde** kan elke complexe vorm hebben. Een $\text{re}^{i\theta}$ -invoer veroorzaakt echter een fout in de hoekmodus Graden.

Opmerking: u moet haakjes gebruiken voor een invoer $(r \angle \theta)$ in poolcoördinaten.

$$\left(\begin{bmatrix} 3 & \angle \frac{\pi}{4} & \angle \frac{\pi}{6} \end{bmatrix} \right) \text{► Rect} \quad \begin{bmatrix} \frac{3 \cdot \sqrt{2}}{4} & \frac{3 \cdot \sqrt{2}}{4} & \frac{3 \cdot \sqrt{3}}{2} \end{bmatrix}$$

$$\left[\begin{array}{ccc} a & \angle b & \angle c \end{array} \right] \text{► Rect} \quad \begin{bmatrix} a \cdot \cos(b) \cdot \sin(c) & a \cdot \sin(b) \cdot \sin(c) & a \cdot \cos(c) \end{bmatrix}$$

In de hoekmodus Radialen:

$$\left(4 \cdot e^{\frac{\pi}{3}} \right) \text{► Rect} \quad 4 \cdot e^{\frac{\pi}{3}}$$

$$\left(\left(4 \angle \frac{\pi}{3} \right) \right) \text{► Rect} \quad 2 + 2 \cdot \sqrt{3} \cdot i$$

In de hoekmodus Decimale graden:

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

In de hoekmodus Graden:

$((4 \angle 60)) \blacktriangleright$ Rect

$2+2\sqrt{3}\cdot i$

Opmerking: om $\angle$ te typen selecteert u dit uit de symbolenlijst in de Catalogus.

ref()

Catalogus >

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

Geeft de rij-echelonvorm van *MatrixI*.

Optioneel wordt elk matricelement behandeld als nul als de absolute waarde ervan minder is dan *Tol*. Deze tolerantie wordt alleen gebruikt als de matrix gegevens met een drijvende komma heeft, en geen symbolische variabelen bevat die geen waarde toegekend hebben gekregen. Anders wordt *Tol* genegeerd.

- Als u   gebruikt, of de modus **Automatisch of Benaderend** instelt op Benaderend, dan worden berekeningen uitgevoerd met behulp van de drijvende komma.
- Als *Tol* wordt weggelaten of niet wordt gebruikt, dan wordt de standaardtolerantie berekend als:
 $5E-14 \cdot \max(\dim(\text{MatrixI})) \cdot \text{rowNorm}(\text{MatrixI})$

Vermijd ongedefinieerde elementen in *MatrixI*. Deze kunnen leiden tot onverwachte resultaten.

Bijvoorbeeld: als in de volgende uitdrukking *aniet* gedefinieerd is, dan verschijnt er een waarschuwingsbericht en wordt de uitkomst weergegeven als:

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

De waarschuwing verschijnt omdat het gegeneraliseerde element $1/a$ niet geldig zou zijn voor $a=0$.

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

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow mI \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

ref(*mI*) 

$$\begin{bmatrix} 1 & \frac{d}{c} \\ 0 & 1 \end{bmatrix}$$

U kunt dit vermijden door van tevoren een waarde in *a* op te slaan of door de beperkende operator ("|") te gebruiken om een waarde te vervangen, zoals te zien is in het volgende voorbeeld.

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

Opmerking: zie ook `rref()`, page 166.

RefreshProbeVars

RefreshProbeVars

Geeft u toegang tot sensorgegevens van alle aangesloten sensorsondes in uw TI-Basic-programma.

StatusVar
waarde

Status

statusVar =0 Normaal (ga verder met het programma)

De Vernier DataQuest™-toepassing staat in de modus gegevensverzameling.

statusVar =1 **Opmerking:** Dit commando werkt alleen als de toepassing Vernier DataQuest™ in de meetmodus



staat.

statusVar =2 De toepassing Vernier DataQuest™ is niet gestart.

statusVar =3 De toepassing Vernier DataQuest™ is gestart, maar u hebt geen sondes aangesloten.

Voorbeeld

```
Define temp()=
```

```
Prgm
```

```
© Controleer of het systeem  
klaar is
```

```
RefreshProbeVars status
```

```
If status=0 Then
```

```
Disp "klaar"
```

```
Voor n,1,50
```

```
RefreshProbeVars status
```

```
temperatuur:=meter.temperatuur
```

```
Disp "Temperatuur:  
",temperatuur
```

```
If temperatuur>30 Then
```

```
Disp "Te heet"
```

```
EndIf
```

```
© Wacht 1 seconde tussen  
metingen
```

```
Wait 1
```

```
EndFor
```

```
Else
```

Disp "Niet klaar. Probeer het later opnieuw"

EndIf

EndPrgm

Opmerking: dit kan ook worden gebruikt met TI-Innovator™ Hub.

remain()

remain(Uitdr1, Uitdr2) ⇒ uitdrukking

remain(Lijst1, Lijst2) ⇒ lijst

remain(Matrix1, Matrix2) ⇒ matrix

Geeft de rest van het eerste argument ten opzichte van het tweede argument zoals wordt gedefinieerd door de identiteiten:

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

$\text{remain}(x,y) = x - y \cdot \text{Part}(x/y)$

Merk op dat daarom geldt: **remain(-x,y) = -remain(x,y)**. Het resultaat is ofwel nul, of heeft hetzelfde teken als het eerste argument.

Opmerking: zie ook **mod()**, pag. 122.

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

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

Request

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

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

Programmeeropdracht: Pauzeert het programma en toont een dialoogvenster met het bericht *promptString* en een invoervak voor het antwoord van de gebruiker.

Als de gebruiker een antwoord typt en op **OK** klikt, wordt de inhoud van het invoervak toegekend aan variabele *var*.

Definieer een programma:

```
Definieer request_demo()=Prgm
  Request "Straal: ",r
  Disp "Oppervlakte = ",pi*r^2
EndPrgm
```

Voer het programma uit en typ een antwoord:

`request_demo()`

Als de gebruiker klikt op **Annuleren**, gaat het programma verder zonder invoer te accepteren. Het programma gebruikt de vorige waarde van *var* als *var* al gedefinieerd was.

Het optionele argument *ToonVlag* kan een willekeurige uitdrukking zijn.

- Als *ToonVlag* wordt weggelaten of wordt uitgewerkt tot **1**, dan worden het promptbericht en het antwoord van de gebruiker weergegeven in de geschiedenis van de rekenmachine.
- Als *ToonVlag* wordt uitgewerkt tot **0** worden de prompt en het antwoord niet weergegeven in de geschiedenis.

Het optionele argument *statusVar* geeft het programma een manier om te vast te stellen hoe de gebruiker het dialoogvenster heeft afgesloten. Merk op dat voor *statusVar* het argument *ToonVlag* vereist is.

- Als de gebruiker op **OK** heeft geklikt of op **Enter** of **Ctrl+Enter** heeft gedrukt, dan wordt de variabele *statusVar* ingesteld op waarde **1**.
- Anders wordt de variabele *statusVar* ingesteld op waarde **0**.

Met het argument *func()* kan een programma het antwoord van een gebruiker opslaan als een functiedefinitie. Deze syntax werkt alsof de gebruiker het volgende commando heeft uitgevoerd:

Definieer *func(arg1, ...argn) = antwoord van gebruiker*

Het programma kan vervolgens de gedefinieerde functie *func()* gebruiken. De *promptString* moet de gebruiker helpen om een passend *antwoord van de gebruiker* in te voeren, dat de functiedefinitie voltooit.



Resultaat na selectie van **OK**:

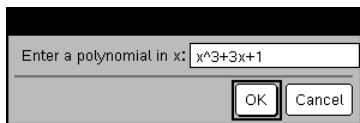
Straal: 6/2
Area= 28.2743

Definieer een programma:

```
Definieer polynomial()=Prgm
  Request "Voer een veelterm in
x:",p(x)
  Disp "Echte oplossingen
zijn:",polyRoots(p(x),x)
EndPrgm
```

Voer het programma uit en typ een antwoord:

polynomial()



Resultaat na het invoeren van x^3+3x+1 en selecteren van **OK**:

Echte oplossingen zijn: $\{-0.322185\}$

Opmerking: u kunt het commando **Request** gebruiken binnen een door de gebruiker gedefinieerd programma, maar niet binnen een functie.

Om een programma te stoppen dat een **Request**-commando binnen een oneindige lus bevat:

- **Rekenmachine:** Houd de toets  ingedrukt en druk enkele malen op .
- **Windows®:** Houd **F12** ingedrukt en druk enkele malen op **Enter**.
- **Macintosh®:** Houd **F5** ingedrukt en druk enkele malen op **Enter**.
- **iPad®:** De app toont een instructie. U kunt blijven wachten of annuleren.

Opmerking: zie ook **RequestStr**, page 160.

RequestStr

RequestStr *promptString, var[, ToonVlag]*

Programmeeropdracht: Werkt hetzelfde als de eerste syntax van het commando **Request**, behalve dat het antwoord van de gebruiker altijd wordt geïnterpreteerd als een string. Het commando **Request** daarentegen interpreteert het antwoord als een uitdrukking, tenzij de gebruiker deze tussen aanhalingstekens zet ("").

Opmerking: u kunt het commando **RequestStr** binnen een door de gebruiker gedefinieerd programma gebruiken, maar niet binnen een functie.

Om een programma te stoppen dat een **RequestStr**-commando bevat binnen een oneindige lus:

- **Rekenmachine:** Houd de toets  ingedrukt en druk enkele malen op .
- **Windows®:** Houd **F12** ingedrukt en druk enkele malen op **Enter**.
- **Macintosh®:** Houd **F5** ingedrukt en druk

Definieer een programma:

```
Definieer requestStr_demo()=Prgm
    RequestStr "Uw naam:",naam,0
    Disp "Antwoord heeft ",dim
    (naam)," tekens."
EndPrgm
```

Voer het programma uit en typ een respons:

`requestStr_demo()`



Resultaat na het selecteren van **OK** (merk op dat het *ToonVlag* argument van **0** de prompt en het antwoord weglaat uit de geschiedenis):

`requestStr_demo()`

enkele malen op **Enter**.

Antwoord heeft 5 tekens.

- **iPad®**: De app toont een instructie. U kunt blijven wachten of annuleren.

Opmerking: zie ook **Request**, page 158.

Return

Return [*Uitdr*]

Geeft *Uitdr* als het resultaat van de functie. Gebruik dit commando binnen een blok **Func...EndFunc**.

Opmerking: Gebruik **Return** zonder een argument binnen een blok **Prgm...EndPrgm** om een programma af te sluiten.

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

```
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(*LijstI*[, *Aantal*]) ⇒ *lijst*

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

Geeft het meest rechtse *Aantal* elementen in *LijstI*.

Als u *Aantal* weglaat, wordt de hele *LijstI* gegeven.

right(*bronString*[, *Aantal*]) ⇒ *string*

$\text{right}(\text{"Hello"}, 2)$ "lo"

Geeft het meest rechtse *Aantal* tekens in de tekenreeks *bronString*.

Als u *Aantal* weglaat, wordt de hele *bronString* gegeven.

right(*Vergelijken*) ⇒ *uitdrukking*

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

Geeft het rechterlid van een vergelijking of ongelijkheid.

rk23(*Uitdr*, *Var*, *afhVar*, {*Var0*, *VarMax*}, *afhVar0*, *VarStap* [, *fouttol*]) $\Rightarrow$ *matrix*

rk23(*StelselUitdr*, *Var*, *LijstVanAfhVars*, {*Var0*, *VarMax*}, *LijstVanAfhVars0*, *VarStap* [, *fouttol*]) $\Rightarrow$ *matrix*

rk23(*LijstVanUitdr*, *Var*, *LijstVanAfhVars*, {*Var0*, *VarMax*}, *LijstVanAfhVars0*, *VarStap* [, *fouttol*]) $\Rightarrow$ *matrix*

Gebruikt de Runge-Kutta-methode om het stelsel op te lossen

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

met *afhVar*(*Var0*)=*afhVar0* in het interval [*Var0*,*VarMax*]. Geeft een matrix waarvan de eerste rij de *Var*-uitvoerwaarden definieert, zoals gedefinieerd door *VarStap*. De tweede rij definieert de waarde van het eerste deel van de oplossing bij de overeenkomstige *Var*-waarden, enzovoort.

Uitdr is het rechterlid dat de gewone differentiaalvergelijking (GDV) definieert.

StelselUitdr is een stelsel van rechterleden dat het stelsel van GDV's definiëren (deze komt overeen met de volgorde van afhankelijke variabelen in *LijstVanAfhVars*).

LijstVanUitdr is een lijst van rechterleden die het stelsel van GDV's definiëren (deze komt overeen met de volgorde van afhankelijke variabelen in *LijstVanAfhVars*).

Var is de onafhankelijke variabele.

LijstVanAfhVars is een lijst van afhankelijke variabelen.

{*Var0*, *VarMax*} is een lijst met twee elementen die de functie de opdracht geeft om van *Var0* tot *VarMax* te integreren.

Differentiaalvergelijking:

$$y' = 0,001 \cdot y \cdot (100 - y) \text{ en } y(0) = 10$$

$$\text{rk23}\left\{0,001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right\}$$

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

Om het hele resultaat te zien drukt u op $\blacktriangle$ en gebruikt u vervolgens $\blacktriangleleft$ en $\blacktriangleright$ om de cursor te verplaatsen.

Dezelfde vergelijking met *fouttol* ingesteld op 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

Vergelijk bovengenoemde resultaten met de exacte CAS-oplossing die verkregen is met behulp van *deSolve*() en *seqGen*():

$$\text{deSolve}\left\{y' = 0,001 \cdot y \cdot (100 - y) \text{ and } y(0) = 10, t, y\right\}$$

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

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

{ 10., 10.9367, 11.9494, 13.0423, 14.2189, 15.49 }

Stelsel vergelijkingen:

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

met *y1*(0)=2 en *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

LijstVanAfhVars0 is een lijst met beginwaarden voor afhankelijke variabelen.

Als $VarStap$ wordt uitgewerkt tot een getal dat niet nul is, geldt: $sign(VarStap) = sign(VarMax - Var0)$ en oplossingen worden gegeven bij $Var0 + i * VarStap$ voor alle $i = 0, 1, 2, \dots$ zodanig dat $Var0 + i * VarStap$ valt binnen $[var0, VarMax]$ (mogelijk is er geen oplossingswaarde bij $VarMax$).

als *VarStep* wordt uitgewerkt naar nul, worden oplossingen gegeven voor de "Runge-Kutta" *Var*-waarden.

fouttol is de fouttolerantie
(standaardwaarde is 0,001).

root()

root(*Uitdr*) $\Rightarrow$ wortel

root(*Uitdr1*, *Uitdr2*) $\Rightarrow$ *wortel*

root(*Uitdr*) geeft de wortel van *Uitdr*.

root(*Uitdr1*, *Uitdr2*) geeft de *Uitdr2*-wortel van *Uitdr1*. *Uitdr1* kan bestaan uit een reële of complexe constante met drijvende komma, een geheel getal of een complexe rationale constante of een algemene symbolische uitdrukking.

Opmerking: zie ook **N-de wortelsjabloon**, pag. 1.

$\sqrt[3]{8}$	2
$\sqrt[3]{3}$	$\frac{1}{3^3}$
$\sqrt[3]{3}$	1.44225

rotate()

```
rotate(Geheel getal[,AantalRotaties]) ⇒  
geheel getal
```

In de Bin-grondtalmodus:

[illegible]

Om het hele resultaat te zien drukt u op **▲** en gebruikt u vervolgens **◀** en **▶** om de cursor te verplaatsen.

Roteert de bits in een binair geheel getal. U kunt *Geheel getal1* invoeren in elk grondtal; het wordt automatisch geconverteerd naar een 64-bits binaire vorm met een plus- of min-teken. Als de grootte van *Geheel getal1* te groot is voor deze vorm, dan wordt een symmetrische modulo-bewerking gebruikt om het binnen het bereik te brengen. Voor meer informatie zie ► **Grondtal2**, pag. 18.

Als *AantalRotaties* positief is, dan is de rotatie naar links. Als *AantalRotaties* negatief is, dan is de rotatie naar rechts. De standaardinstelling is -1 (één bit naar rechts roteren).

Bijvoorbeeld in een rotatie naar rechts:

Elk bit roteert naar rechts.

0b00000000000001111010110000110101

Het meest rechtse bit roteert naar het meest linkse.

Dit levert op:

0b10000000000000111101011000011010

Het resultaat wordt weergegeven volgens de grondtal-modus.

rotate(*Lijst1*[,*AantalRotaties*]) ⇒ *lijst*

Geeft een kopie van *Lijst1* die met *AantalRotaties* elementen naar rechts of links is gerooteerd. Verandert *Lijst1* niet.

Als *AantalRotaties* positief is, dan is de rotatie naar links. Als *AantalRotaties* negatief is, dan is de rotatie naar rechts. De standaardinstelling is -1 (één element naar rechts roteren).

rotate(*String1*[,*AantalRotations*]) ⇒ *string*

Geeft een kopie van *String1* die met *AantalRotations* tekens naar rechts of links is gerooteerd. Verandert *String1* niet.

In de Hex-grondtalmodus:

rotate(0h78E)	0h3C7
rotate(0h78E,-2)	0h80000000000001E3
rotate(0h78E,2)	0h1E38

Belangrijk: om een binair of hexadecimaal getal in te voeren moet u altijd het prefix 0b of 0h gebruiken (nul, niet de letter O).

In de Dec-grondtalmodus:

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

rotate("abcd")	"dabc"
rotate("abcd",-2)	"cdab"
rotate("abcd",1)	"bcda"

rotate()Catalogus > 

Als *AantalRotaties* positief is, dan is de rotatie naar links. Als *AantalRotaties* negatief is, dan is de rotatie naar rechts. De standaardinstelling is -1 (één teken naar rechts roteren).

round()Catalogus > 

round(Uitdr1[, cijfers]) $\Rightarrow$ uitdrukking

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

Geeft het argument, afgerond op het gespecificeerde aantal cijfers achter de komma.

cijfers moet een geheel getal zijn in het bereik 0-12. Als *cijfers* niet is inbegrepen, wordt het argument afgerond op 12 significante cijfers.

Opmerking: de modus voor cijferweergave kan invloed hebben op hoe dit wordt weergegeven.

round(Lijst1[, cijfers]) $\Rightarrow$ lijst

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

Geeft een lijst van elementen, afgerond op het gespecificeerde aantal cijfers.

round(Matrix1[, cijfers]) $\Rightarrow$ matrix

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

Geeft een matrix met de elementen, afgerond op het gespecificeerde aantal cijfers.

rowAdd()Catalogus > 

rowAdd(Matrix1, rIndex1, rIndex2) $\Rightarrow$ matrix

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

Geeft een kopie van *Matrix1* met rij *rIndex2* vervangen door de som van de rijen *rIndex1* en *rIndex2*.

$\text{rowAdd}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, 1, 2\right)$	$\begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$
--	--

rowDim()Catalogus > **rowDim(Matrix)** $\Rightarrow$ uitdrukkingGeeft het aantal rijen in *Matrix*.**Opmerking:** zie ook **colDim()**, pag. 27.

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

rowNorm()Catalogus > **rowNorm(Matrix)** $\Rightarrow$ uitdrukkingGeeft het maximum van de sommen van de absolute waarden van de elementen in de rijen in *Matrix*.**Opmerking:** alle matricelementen moeten vereenvoudigen tot getallen. Zie ook **colNorm()**, pag. 27.

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

rowSwap()Catalogus > **rowSwap(Matrix1, rIndex1, rIndex2)** $\Rightarrow$ matrixGeeft *Matrix1* met rijen *rIndex1* en *rIndex2* verwisseld.

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

rref()Catalogus > **rref(Matrix1[, Tol])** $\Rightarrow$ matrixGeeft de gereduceerde rij-echelonvorm van *Matrix1*.

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

Optioneel wordt elk matricelement behandeld als nul als de absolute waarde ervan minder is dan *Tol*. Deze tolerantie wordt alleen gebruikt als de matrix gegevens met een drijvende komma heeft, en geen symbolische variabelen bevat die geen waarde toegekend hebben gekregen. Anders wordt *Tol* genegeerd.

 rref $\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right)$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
---	--

- Als u   gebruikt, of de modus

Automatisch of Benaderend instelt op Benaderend, dan worden berekeningen uitgevoerd met behulp van de drijvende komma.

- Als *Tol* wordt weggelaten of niet wordt gebruikt, dan wordt de standaardtolerantie berekend als:
 $5E-14 \cdot \max(\dim(\text{Matrix } I)) \cdot \text{rowNorm}(\text{Matrix } I)$

Opmerking: zie ook `ref()`, page 156.

S

sec()



sec(Uitdr1) ⇒ uitdrukking

In de hoekmodus Graden:

sec(Lijst1) ⇒ lijst

Geeft de secans van *Uitdr1* of geeft een lijst met de secansen van alle elementen in *List1*.

$$\frac{\sec(45)}{\sec(\{1, 2, 3, 4\})} = \frac{\sqrt{2}}{\left\{ \frac{1}{\cos(1)}, 1.00081, \frac{1}{\cos(4)} \right\}}$$

Opmerking: het argument wordt geïnterpreteerd als een hoek in graden, decimale graden of radialen volgens de huidige hoekmodus-instelling. U kunt °, °G of °R gebruiken om de hoekmodus tijdelijk te onderdrukken.

sec⁻¹()

sec⁻¹(Uitdr1) ⇒ uitdrukking

In de hoekmodus Graden:

sec⁻¹(Lijst1) ⇒ lijst

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

Geeft de hoek waarvan de secans *Uitdr1* is of geeft een lijst met de inverse secans van elk element in *Lijst1*.

In de hoekmodus Decimale graden:

Opmerking: de uitkomst wordt in graden, decimale graden of radialen gegeven, volgens de ingestelde hoekmodus.

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

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door `arcsec(...)` in te typen.

In de hoekmodus Radialen:

$$\sec^{-1}(\{1, 2, 5\}) \quad \left\{ 0, \frac{\pi}{3}, \cos^{-1}\left(\frac{1}{5}\right) \right\}$$

sech()

Catalogus > 

sech(Uitdr1) $\Rightarrow$ uitdrukking

sech(Lijst1) $\Rightarrow$ lijst

Geeft de secans hyperbolicus van *Uitdr1* of geeft een lijst met de secansen hyperbolicus van de elementen in *Lijst1*.

$$\begin{aligned} \text{sech}(3) &= \frac{1}{\cosh(3)} \\ \text{sech}(\{1, 2, 3, 4\}) &= \left\{ \frac{1}{\cosh(1)}, 0.198522, \frac{1}{\cosh(4)} \right\} \end{aligned}$$

sech⁻¹()

Catalogus > 

sech⁻¹(Uitdr1) $\Rightarrow$ uitdrukking

sech⁻¹(Lijst1) $\Rightarrow$ lijst

Geeft de inverse secans hyperbolicus van *Uitdr1* of geeft een lijst met de inverse secans hyperbolicus van elk element in *Lijst1*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arcsech (...)** in te typen.

In de hoekmodus Radialen en rechthoekige complexe opmaak:

$$\begin{aligned} \text{sech}^{-1}(1) &= 0 \\ \text{sech}^{-1}(\{1, -2, 2, 1\}) &= \left\{ 0, \frac{2 \cdot \pi}{3} \cdot i, 8 \cdot \text{E-15} + 1.07448 \cdot i \right\} \end{aligned}$$

Send

Hub Menu

Send *exprOrString1* [, *exprOrString2*] ...

Programmeeropdracht: Verzendt een of meer TI-Innovator™ Hub opdrachten naar een aangesloten hub.

exprOrString moet een geldige TI-Innovator™ Hub opdracht zijn. Gewoonlijk bevat *exprOrString* een opdracht **"SET ..."** om een apparaat te besturen of een opdracht **"READ ..."** om gegevens op te vragen.

De argumenten worden na elkaar naar de hub verzonden.

Voorbeeld: Zet het blauwe element van de ingebouwde RGB LED 0,5 seconden aan.

Send "SET COLOR.BLUE ON TIME .5"
Done

Voorbeeld: Vraag de huidige waarde van de ingebouwde lichtniveau- sensor van de hub op. Een opdracht **Get** haalt de waarde op en wijst deze toe aan de variabele *lichtniveau*.

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

Opmerking: U kunt de opdracht **Send** gebruiken binnen een door de gebruiker gedefinieerd programma, maar niet binnen een functie.

Opmerking: Zie ook **Get** (pag. 82), **GetStr** (pag. 89), en **eval()** (pag. 66).

Voorbeeld: Stuur een berekende frequentie naar de ingebouwde luidspreker van de hub. Gebruik de speciale variabele *iostr.SendAns* om de hubopdracht met de uitgewerkte uitdrukking te tonen.

$n:=50$	50
$m:=4$	4
Send "SET SOUND eval(m·n)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

seq()

Catalogus >

seq(Uitdr, Var, Laag, Hoog[, Stap]) ⇒ lijst

Verhoogt *Var* van *Laag* naar *Hoog* met een stapgrootte van *Stap*, werkt *Uitdr* uit en geeft de resultaten terug in een lijst. De oorspronkelijke inhoud van *Var* is er nog steeds nadat **seq()** is uitgevoerd.

De standaardwaarde voor *Stap* = 1.

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

Opmerking: Om een decimale benaderende uitkomst te forceren,

Rekenmachine: Druk op .

Windows®: Druk op **Ctrl+Enter**.

Macintosh®: Druk op **⌘+Enter**.

iPad®: Houd **Enter** ingedrukt en selecteer



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

seqGen()

Catalogus >

seqGen(Uitdr, Var, afhVar, {Var0, VarMax}, [LijstVanBeginTermen [, VarStap [, PlafondWaarde]]) ⇒ lijst

Genereer de eerste 5 termen van de rij $u(n) = u(n-1)^2/2$, waarbij $u(1)=2$ en *VarStap* = 1.

$\text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right)$	$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$
---	---

Genereert een lijst met termen voor de rij $afhVar(Var)=Uitdr$, als volgt: Verhoogt de onafhankelijke variabele Var van $Var0$ naar $VarMax$ met $VarStap$, werkt $afhVar(Var)$ uit voor overeenkomstige waarden van Var met behulp van de formule $Uitdr$ en de $LijstVanBeginTermen$, en geeft de resultaten terug in een lijst.

seqGen($LijstOfStelselVanUitdr$, Var , $LijstVanAfhVars$, $\{Var0, VarMax\}$ [, $MatrixVanBeginTermen$ [, $VarStap$ [, $PlafondWaarde$]]]) $\Rightarrow$ matrix

Genereert een matrix van termen voor een stelsel (of lijst) van rijen $LijstVanAfhVars(Var)=LijstOfStelselVanUitdr$, als volgt: Verhoogt de onafhankelijke variabele Var van $Var0$ naar $VarMax$ met $VarStap$, werkt $LijstVanAfhVars(Var)$ uit voor overeenkomstige waarden van Var met behulp van de formule $LijstOfStelselUitdr$ en de $MatrixVanBeginTermen$, en geeft de resultaten terug in een lijst.

De oorspronkelijke inhoud van Var is ongewijzigd nadat **seqGen()** is uitgevoerd.

De standaardwaarde voor $VarStap = 1$.

Voorbeeld waarin $Var0=2$:

$$\text{seqGen}\left\{\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right\}$$

$$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$$

Voorbeeld waarin de beginterm symbolisch is:

$$\text{seqGen}\{u(n-1)+2, n, u, \{1, 5\}, \{a\}\}$$

$$\{a, a+2, a+4, a+6, a+8\}$$

Stelsel van twee rijen:

$$\text{seqGen}\left\{\left\{\frac{1}{n}, \frac{u_2(n-1)}{2} + u_1(n-1)\right\}, n, \{u_1, u_2\}, \{1, 5\}, \left[-\right]\right\}$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{12} & \frac{19}{24} \end{bmatrix}$$

Opmerking: De lege plaats () in de begintermenmatrix hierboven wordt gebruikt om aan te geven dat de beginterm voor $u_1(n)$ wordt berekend met behulp van de expliciete rijformule $u_1(n)=1/n$.

seqn()

seqn($Uitdr\{u, n$ [, $LijstVanBeginTermen$ [, $nMax$ [, $PlafondWaarde$]]]) $\Rightarrow$ lijst

Genereert een lijst met termen voor een rij $u(n)=Uitdr(u, n)$, als volgt: Verhoogt n van 1 tot $nMax$ met 1, werkt $u(n)$ uit voor overeenkomstige waarde van n met behulp van de formule $Uitdr(u, n)$ en de $LijstVanBeginTermen$ en geeft de resultaten terug in een lijst.

seqn($Uitdr\{n$ [, $nMax$ [, $PlafondWaarde$]]]) $\Rightarrow$ lijst

Genereer de eerste 6 termen van de rij $u(n) = u(n-1)/2$, waarbij $u(1)=2$.

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

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left\{\frac{1}{n^2}, 6\right\}$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

Genereert een lijst met termen voor een niet-recursieve rij $u(n)=\text{Uitdr}(n)$, als volgt: Verhoogt n van 1 tot $n\text{Max}$ met 1, werkt $u(n)$ uit voor overeenkomstige waarden van n met behulp van de formule $\text{Uitdr}(n)$ en geeft de resultaten terug in een lijst.

Als $n\text{Max}$ ontbreekt, dan wordt $n\text{Max}$ ingesteld op 2500

Als $n\text{Max}=0$, dan wordt $n\text{Max}$ ingesteld op 2500

Opmerking: **seqn()** roept **seqGen()** aan met $n0=1$ en $n\text{stap}=1$

series()

series(Uitdr1, Var, Orde [, Punt])
⇒ *uitdrukking*

series(Uitdr1, Var, Orde [, Punt]) | Var>Punt ⇒ *uitdrukking*

series(Uitdr1, Var, Orde [, Punt]) | Var<Punt ⇒ *uitdrukking*

Geeft een gegeneraliseerde ingekorte machtreeks-representatie van Uitdr1 uitgewerkt rond $Punt$ tot en met graad $Orde$. $Orde$ kan elk rationaal getal zijn. De resulterende machten van $(Var - Punt)$ kunnen negatieve en/of gebroken exponenten bevatten. De coëfficiënten van deze machten kunnen logaritmen bevatten van $(Var - Punt)$ en andere functies van Var die gedomineerd worden door alle machten van $(Var - Point)$ met hetzelfde teken in de exponent.

$Punt$ heeft als standaardwaarde 0. $Punt$ kan ∞ of $-\infty$ zijn; in die gevallen is de uitwerking tot en met graad $Orde$ in $1/(Var - Punt)$.

$$\text{series}\left(\frac{1-\cos(x-1)}{(x-1)^2}, x, 4, 1\right) \quad \frac{1}{2} - \frac{(x-1)^2}{24} + \frac{(x-1)^4}{720}$$

$$\text{series}\left(\frac{-1}{e^{z-}}, z, 1\right) \quad z - 1$$

$$\text{series}\left(\left(1 + \frac{1}{n}\right)^n, n, 2, \infty\right) \quad e - \frac{e}{2 \cdot n} + \frac{11 \cdot e}{24 \cdot n^2}$$

$$\text{series}\left(\tan^{-1}\left(\frac{1}{x}\right), x, 5\right), x > 0 \quad \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5}$$

$$\text{series}\left(\int \frac{\sin(x)}{x} dx, x, 6\right) \quad x - \frac{x^3}{18} + \frac{x^5}{600}$$

$$\text{series}\left(\int_0^x \sin(x \cdot \sin(t)) dt, x, 7\right) \quad \frac{x^3}{2} - \frac{x^5}{24} + \frac{29 \cdot x^7}{720}$$

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

series(...) geeft “**series(...)**” als hij niet in staat is om een dergelijke representatie te bepalen, zoals bij essentiële singulariteiten zoals $\sin(1/z)$ bij $z=0$, $e^{-1/z}$ bij $z=0$ of e^z bij $z = \infty$ of $-\infty$.

Als de reeks of één van zijn afgeleiden een sprong-discontinuïteit bij *Punt* heeft, dan bevat het resultaat waarschijnlijk sub-uitdrukkingen van de vorm **sign(...)** of **abs(...)** voor een reële uitbreidingsvariabele of **(-1)^{floor(...angle(...))}** voor een complexe uitbreidingsvariabele, dit is een variabele die eindigt op “_”. Als u de reeks alleen voor waarden aan één zijde van *Punt* wilt gebruiken, voeg dan de juiste specificatie toe: “| *Var* > *Punt*”, “| *Var* < *Punt*”, “| *Var* ≥ *Punt*” of “| *Var* ≤ *Punt*” om een eenvoudiger resultaat te krijgen.

series() kan symbolische benaderingen van onbepaalde integralen en bepaalde integralen leveren waarvoor symbolische oplossingen anders niet verkregen kunnen worden.

series() is distributief over 1ste-argument-lijsten en matrices.

series() is een gegeneraliseerde versie van **taylor()**.

Zoals geïllustreerd in het laatste voorbeeld rechts kunnen de weergaveroutines onder het resultaat dat geproduceerd wordt door **series(...)** termen opnieuw ordenen, zodat de dominante term niet de meest linkse is.

Opmerking: Zie ook **dominantTerm()**, pag. 59.

setMode()

setMode(modeNaamGeheel getal, instellingGeheel getal) ⇒ geheel getal

setMode(lijst) ⇒ lijst met gehele getallen

Alleen geldig binnen een functie of programma.

Geef de benaderde waarde van π weer met behulp van de standaardinstelling voor Cijfers weergeven, en geef π vervolgens weer met een instelling van Vast2. Controleer om te zien of de standaardinstelling hersteld wordt nadat het programma is uitgevoerd.

setMode(modeNaamGeheel getal, instellingGeheel getal) stelt de modus *modeNaamGeheel getal* tijdelijk in op de nieuwe instelling *instellingGeheel getal*, een geeft een geheel getal dat correspondeert met de oorspronkelijke instelling van die modus. De verandering is beperkt tot de duur van de uitvoering van het programma/de functie.

modeNaamGeheel getal specificeert welke modus u wilt instellen. Dit moet één van de gehele getallen voor modi in onderstaande tabel zijn.

instellingGeheel getal specificeert de nieuwe instelling voor de modus. Dit moet één van de gehele getallen voor instellingen in onderstaande tabel zijn, voor de specifieke modus die u instelt.

setMode(lijt) stelt u in staat meerdere instellingen te veranderen. *lijst* bevat paren van gehele getallen voor modi en instellingen. **setMode(lijt)** geeft een vergelijkbare lijst waarvan de paren gehele getallen de oorspronkelijke modi en instellingen representeren.

Als u alle modusinstellingen hebt opgeslagen met **getMode(0) → var**, dan kunt u **setMode(var)** gebruiken om die instellingen te herstellen tot de functie of het programma wordt afgesloten. Zie **getMode()**, pag. 88.

Opmerking: de huidige modusinstellingen worden doorgegeven aan opgeroepen subroutines. Als een subroutine een modusinstelling verandert, dan gaat de modusverandering verloren als de besturing terugkeert naar de oproeproutine.

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

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

Modus-naam	Modus nummer	Instellingsnummers
Cijfers weergeven	1	1=Drijvend, 2=Drijvend1, 3=Drijvend2, 4=Drijvend3, 5=Drijvend4, 6=Drijvend5, 7=Drijvend6, 8=Drijvend7, 9=Drijvend8, 10=Drijvend9, 11=Drijvend10, 12=Drijvend11, 13=Drijvend12, 14=Vast0, 15=Vast1, 16=Vast2, 17=Vast3, 18=Vast4, 19=Vast5, 20=Vast6, 21=Vast7, 22=Vast8, 23=Vast9, 24=Vast10, 25=Vast11, 26=Vast12
Hoek	2	1=Radialen, 2=Graden, 3=Decimale graden
Exponentiële opmaak	3	1=Normaal, 2=Wetenschappelijk, 3=Ingenieursnotatie
Reëel of complex	4	1=Reëel, 2=Rechthoekig, 3=Polair
Automatisch of benaderend	5	1=Automatisch, 2=Benaderend, 3=Exact
Vectoropmaak	6	1=Rechthoekig, 2=Cilindrisch, 3=Bolvormig
Grondtal	7	1=Decimaal, 2=Hexadecimaal, 3=Binair
Eenhedenstelsel	8	1=SI, 2=Eng/US

shift()

Catalogus > 

shift(*Geheel getal***[**,*#Verschuivingen***])**
⇒*geheel getal*

Verschuift de bits in een binair geheel getal. U kunt *Geheel getal* in elk grondtal invoeren; het wordt automatisch geconverteerd naar een 64-bits binaire vorm met een teken. Als de grootte van *Geheel getal* te groot is voor deze vorm, dan wordt een symmetrische modulo-bewerking gebruikt om het binnen het bereik te brengen. Zie voor meer informatie **►Base2**, pag. 18.

Als *#Verschuivingen* positief is, dan is de verschuiving naar links. Als *#Verschuivingen* negatief is, dan is de verschuiving naar rechts. De standaardinstelling is -1 (één bit naar rechts verschuiven).

In de Bin-grondtalmodus:

```
shift(0b1111010110000110101)
                                0b111101011000011010
shift(256,1)                    0b1000000000
```

In de Hex-grondtalmodus:

```
shift(0h78E)                    0h3C7
shift(0h78E,-2)                 0h1E3
shift(0h78E,2)                  0h1E38
```

Belangrijk: om een binair of hexadecimaal getal in te voeren moet u altijd het prefix 0b of 0h gebruiken (nul, niet de letter O).

In een verschuiving naar rechts vervalt de meest rechtse bit en wordt 0 of 1 ingevoegd om overeen te komen met de meest linkse bit. In een verschuiving naar links vervalt de meest linkse bit en wordt 0 ingevoegd als de meest rechtse bit.

Bijvoorbeeld in een verschuiving naar rechts:

Elke bit schuift naar rechts.

0b00000000000000111101011000011010

Voegt 0 in als de meest linkse bit 0 is, of 1 als de meest linkse bit 1 is.

Dit levert op:

0b00000000000000011110101100001101-0

Het resultaat wordt weergegeven volgens de grondtal-modus. Nullen aan het begin worden niet weergegeven.

shift(Lijst l [,#Verschuivingen])⇒lijst

Geeft een kopie van *Lijst l* die met *#Verschuivingen* elementen naar rechts of links is verschoven. Verandert *Lijst l* niet.

Als *#Verschuivingen* positief is, dan is de verschuiving naar links.

Als *#Verschuivingen* negatief is, dan is de verschuiving naar rechts.

De standaardinstelling is -1 (één element naar rechts verschuiven).

Elementen die aan het begin of eind van *lijst* ingevoegd worden door de verschuiving, worden ingesteld op het symbool "undef".

shift(String l [,#Verschuivingen])⇒string

Geeft een kopie van *String l* die met *#Verschuivingen* tekens naar rechts of links is verschoven. Verandert *String l* niet.

In de Dec-grondtalmodus:

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 "

Als *#Verschuivingen* positief is, dan is de verschuiving naar links.

Als *#Verschuivingen* negatief is, dan is de verschuiving naar rechts.

De standaardinstelling is -1 (één teken naar rechts verschuiven).

Tekens die aan het begin of eind van *string* ingevoegd worden door de verschuiving, worden ingesteld op een spatie.

sign()

Catalogus >

sign(Uitdr1) ⇒ uitdrukking

$$\text{sign}(-3.2) \quad -1.$$

sign(Lijst1) ⇒ lijst

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

sign(Matrix1) ⇒ matrix

$$\text{sign}(1+|x|) \quad 1$$

Geeft, bij reële en complexe *Uitdr1*, *Uitdr1/abs(Uitdr1)* wanneer *Uitdr1* ≠ 0.

Als de complexe opmaak-modus Reëel is:

Geeft 1 als *Uitdr1* positief is.

$$\text{sign}\left(\begin{bmatrix} -3 & 0 & 3 \end{bmatrix}\right) \quad \begin{bmatrix} -1 & \pm 1 & 1 \end{bmatrix}$$

Geeft -1 als *Uitdr1* negatief is.

sign(0) geeft ±1 als de complexe opmaak-modus Reëel is; anders geeft hij zichzelf.

sign(0) representeert de eenheidscirkel in het complexe vlak.

Geeft bij een lijst of matrix de tekens van alle elementen.

simult()

Catalogus >

simult(coëffMatrix, constVector[, Tol])
⇒ matrix

Los op naar x en y:

Geeft een kolomvector die de oplossingen voor een stelsel lineaire vergelijkingen bevat.

$$x + 2y = 1$$

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

Opmerking: zie ook **linSolve()**, pag. 108.

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

coëffMatrix moet een vierkante matrix zijn die de coëfficiënten van de vergelijkingen bevat.

De oplossing is x=-3 en y=2.

Los op:

constVector moet hetzelfde aantal rijen (dezelfde afmeting) als *coeffMatrix* hebben en de constanten bevatten.

Optioneel wordt elk matrixelement behandeld als nul als de absolute waarde ervan minder dan *Tol* is. Deze tolerantie wordt alleen gebruikt als de matrix gegevens met een drijvende komma heeft, en geen symbolische variabelen bevat die geen waarde toegekend hebben gekregen. Anders wordt *Tol* genegeerd.

- Als u de modus **Automatisch of Benaderend** instelt op Benaderend, dan worden berekeningen met behulp van de drijvende komma uitgevoerd.
- Als *Tol* wordt weggelaten of niet wordt gebruikt, dan wordt de standaardtolerantie berekend als:
 $5E-14 \cdot \max(\dim(\text{coeffMatrix})) \cdot \text{rowNorm}(\text{coeffMatrix})$

simult(coeffMatrix, constMatrix[, Tol])
 $\Rightarrow$ matrix

Lost meerdere stelsels lineaire vergelijkingen op, waarbij elk stelsel dezelfde vergelijgingscoëfficiënten, maar verschillende constanten heeft.

Elke kolom in *constMatrix* moet de constanten voor een stelsel vergelijkingen bevatten. Elke kolom in de resulterende matrix bevat de oplossing voor het corresponderende stelsel.

$$ax + by = 1$$

$$cx + dy = 2$$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow \text{matx1}$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$
$\text{simult}\left(\text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	$\begin{bmatrix} -(2 \cdot b - d) \\ a \cdot d - b \cdot c \\ 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix}$

Los op:

$$x + 2y = 1$$

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

$$x + 2y = 2$$

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

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

Voor het eerste stelsel: $x=-3$ en $y=2$. Voor het tweede stelsel: $x=-7$ en $y=9/2$.

Expr ▶sin

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>sin in te typen.

$(\cos(x))^2 \blacktriangleright \text{sin}$	$1 - (\sin(x))^2$
--	-------------------

Geeft *Uitdr* weer in termen van sinus. Dit is een operator voor weergaveconversie. Deze kan alleen op het eind van de invoerregel gebruikt worden.

sin verlaagt alle machten van $\cos(\dots)$ modulo $1 - \sin(\dots)^2$ zodat alle resterende machten van $\sin(\dots)$ exponenten in het bereik $(0, 2)$ hebben. Het resultaat zal dus vrij zijn van $\cos(\dots)$ dan en slechts dan als $\cos(\dots)$ voorkomt in de gegeven uitdrukking met uitsluitend even exponenten.

Opmerking: deze conversie-operator wordt niet ondersteund in de hoekmodi Graden en Decimale graden. Voordat u deze operator gebruikt, dient u ervoor te zorgen dat de hoekmodus is ingesteld op Radialen, en dat *Uitdr* geen expliciete verwijzingen naar graden of decimale graden bevat.

sin()

-toets

$\sin(\text{Uitdr}I) \Rightarrow \text{uitdrukking}$

$\sin(\text{Lijst}I) \Rightarrow \text{lijst}$

$\sin(\text{Uitdr}I)$ geeft de sinus van het argument als een uitdrukking.

$\sin(\text{Lijst}I)$ geeft een lijst van de sinussen van alle elementen in *LijstI*.

Opmerking: het argument wordt geïnterpreteerd als een hoek in graden, in decimale graden of in radialen, volgens de ingestelde hoekmodus. U kunt $^\circ$, G of r gebruiken om de hoekmodusinstelling tijdelijk te onderdrukken.

In de hoekmodus Graden:

$$\begin{array}{l} \sin\left(\frac{\pi_r}{4}\right) \quad \frac{\sqrt{2}}{2} \\ \sin(45^\circ) \quad \frac{\sqrt{2}}{2} \\ \sin(\{0, 60, 90\}) \quad \left\{0, \frac{\sqrt{3}}{2}, 1\right\} \end{array}$$

In de hoekmodus Decimale graden:

$$\sin(50^\circ) \quad \frac{\sqrt{2}}{2}$$

In de hoekmodus Radialen:

$$\begin{array}{l} \sin\left(\frac{\pi}{4}\right) \quad \frac{\sqrt{2}}{2} \\ \sin(45^\circ) \quad \frac{\sqrt{2}}{2} \end{array}$$

sin()



sin(vierkanteMatrix1)⇒vierkanteMatrix

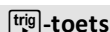
Geeft de matrixsinus van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de sinus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Radialen:

$$\sin \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$$

sin⁻¹()



sin⁻¹(Uitdr1)⇒uitdrukking

sin⁻¹(Lijst1)⇒lijst

sin⁻¹(Uitdr1) geeft de hoek waarvan de sinus *Uitdr1* is als een uitdrukking.

sin⁻¹(Lijst1) geeft een lijst van de inverse sinussen van elk element in *Lijst1*.

Opmerking: de uitkomst wordt in graden, decimale graden of radialen gegeven, volgens de ingestelde hoekmodus.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arcsin(...)** in te typen.

sin⁻¹(vierkanteMatrix1)⇒vierkanteMatrix

Geeft de inverse matrixsinus van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de inverse sinus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Graden:

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

In de hoekmodus Decimale graden:

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

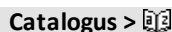
In de hoekmodus Radialen:

$$\sin^{-1}(\{0,0,2,0,5\}) \quad \{0,0,201358,0,523599\}$$

In de hoekmodus Radialen en rechthoekige complexe opmaak-modus:

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

sinh()



sinh(Uitdr1)⇒uitdrukking

sinh(Lijst1)⇒lijst

$$\sinh(1.2) \quad 1.50946$$
$$\sinh(\{0,1,2,3\}) \quad \{0,1.50946,10.0179\}$$

sinh (UitdrI) geeft de sinus hyperbolicus van het argument als een uitdrukking.

sinh (LijstI) geeft een lijst met de sinus hyperbolicus van elk element in *LijstI*.

sinh(vierkanteMatrixI) ⇒ *vierkanteMatrix*

Geeft de matrixsinus hyperbolicus van *vierkanteMatrixI*. Dit is niet hetzelfde als het berekenen van de sinus hyperbolicus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrixI moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Radialen:

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

sinh⁻¹(UitdrI) ⇒ *uitdrukking*

sinh⁻¹(LijstI) ⇒ *lijst*

sinh⁻¹(UitdrI) geeft de inverse sinus hyperbolicus van het argument als een uitdrukking.

sinh⁻¹(LijstI) geeft een lijst met de inverse sinus hyperbolicus van elk element in *LijstI*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arcsinh(...)** in te typen.

sinh⁻¹(vierkanteMatrixI)
⇒ *vierkanteMatrix*

Geeft de inverse matrixsinus hyperbolicus van *vierkanteMatrixI*. Dit is niet hetzelfde als het berekenen van de inverse sinus hyperbolicus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrixI moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

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

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

In de hoekmodus Radialen:

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

SinReg *X*, *Y* [, [*Iteraties*], [*Periode*] [, *Categorie*, *Opnemen*]]

Berekent de sinusoidale regressie op de lijsten *X* en *Y*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en *Y* zijn lijsten met onafhankelijke en afhankelijke variabelen.

Iteraties is een waarde die het maximaal aantal keer (1 tot en met 16) specificeert dat een oplossing wordt geprobeerd. Als dit wordt weggelaten, wordt 8 gebruikt. Doorgaans leiden grotere waarden tot een hogere nauwkeurigheid maar een langere berekeningstijd, en andersom.

Periode specificeert een geschatte periode. Als deze wordt weggelaten, dan moet het verschil tussen waarden in *X* gelijk zijn en in volgorde. Als u *Periode* specificeert, kunnen de verschillen tussen x-waarden ongelijk zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige *X*- en *Y*-gegevens..

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

De uitvoer van **SinReg** is altijd in radialen, ongeacht de instelling van de hoekmodus.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a \cdot \sin(bx+c)+d$
stat.a, stat.b, stat.c, stat.d	Regressiecoëfficiënten
stat.Resid	Residuen uit de regressie

Uitvoervariabele	Beschrijving
stat.XReg	Lijst van de gegevens in de gemodificeerde <i>XLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.YReg	Lijst van gegevens in de gemodificeerde <i>YLijst</i> die feitelijk gebruikt worden in de regressie op basis van beperkingen van <i>Freq</i> , <i>Categorielijst</i> en <i>Categorieën opnemen</i>
stat.FreqReg	Lijst van frequenties die corresponderen met <i>stat.XReg</i> en <i>stat.YReg</i>

solve()

Catalogus > 

solve(*Vergelijking*, *Var*) $\Rightarrow$ *Booleaanse uitdrukking*

solve(*Vergelijking*, *Var*=*Gok*)
 $\Rightarrow$ *Booleaanse uitdrukking*

solve(*Ongelijkheid*, *Var*) $\Rightarrow$ *Booleaanse uitdrukking*

Geeft mogelijke reële oplossingen van een vergelijking of een ongelijkheid voor *Var*. Het doel is om mogelijkheden voor alle oplossingen te geven. Er kunnen echter vergelijkingen of ongelijkheden zijn waarvoor het aantal oplossingen oneindig is.

Oplossingskandidaten kunnen bij sommige combinaties van waarden voor ongedefinieerde variabelen geen reële eindige oplossingen zijn.

Bij de instelling Automatisch van de modus **Automatisch of Benaderend** is het doel om exacte oplossingen te produceren als deze beknopt zijn, en aangevuld met iteratieve zoekacties met benaderende berekeningen wanneer exacte oplossingen niet praktisch zijn.

Als gevolg van de standaard wegdeling van de grootste gemene deler uit de teller en de noemer, kunnen oplossingen mogelijk alleen oplossingen zijn binnen de limiet van één van beide of van beide zijden.

$$\text{solve}(a \cdot x^2 + b \cdot x + c = 0, x)$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a} \text{ or } x = \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a}$$

$$\text{Ans}|a=1 \text{ and } b=1 \text{ and } c=1$$

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

$$\text{solve}(\{(x-a) \cdot e^x = x \cdot (x-a), x\})$$

$$x=a \text{ or } x=-0.567143$$

$$(x+1) \cdot \frac{x-1}{x-1} + x - 3$$

$$2 \cdot x - 2$$

Bij ongelijkheden van het type $\geq$, $\leq$, $<$ of $>$ zijn expliciete oplossingen onwaarschijnlijk, tenzij de ongelijkheid lineair is en alleen *Var* bevat.

Bij de exacte modus worden delen die niet kunnen worden opgelost teruggegeven als een impliciete vergelijking of ongelijkheid.

Gebruik de beperkende operator (" $|$ ") om het oplossingsinterval en/of andere variabelen die optreden in de vergelijking of ongelijkheid te beperken. Wanneer u een oplossing vindt in een interval, dan kunt u de ongelijkheidsoperatoren gebruiken om dat interval uit te sluiten van volgende zoekacties.

Als er geen reële oplossingen worden gevonden wordt **false** (onwaar) gegeven. **true** (waar) wordt gegeven als **solve()** kan bepalen dat een eindige reële waarde van *Var* voldoet aan de vergelijking of ongelijkheid.

Aangezien **solve()** altijd een Booleaans resultaat geeft, kunt u "and," "or" en "not" gebruiken om deze resultaten uit **solve()** met elkaar of met andere Booleaanse uitdrukkingen te combineren.

Oplossingen kunnen een unieke nieuwe onbepaalde constante van de vorm *nj* bevatten, waarbij *j* een geheel getal uit het interval 1–255 is. Dergelijke variabelen duiden een willekeurig geheel getal aan.

In de reële modus duiden machten met een gebroken exponent met een oneven noemer alleen de reële tak aan. Verder duiden meervoudige vertakte uitdrukkingen zoals machten met gebroken exponenten, logaritmes en inverse goniometrische functies alleen de hoofdtak aan. Dientengevolge produceert **solve()** alleen oplossingen die corresponderen met die ene reële of hoofdtak.

Opmerking: zie ook **cSolve()**, **cZeros()**, **nSolve()** en **zeros()**.

$\text{solve}(5 \cdot x - 2 \geq 2 \cdot x, x)$	$x \geq \frac{2}{3}$
---	----------------------

$\text{exact}(\text{solve}((x-a) \cdot e^x = x \cdot (x-a), x))$	$e^x + x = 0$ or $x = a$
--	--------------------------

In de hoekmodus Radialen:

$\text{solve}\left(\tan(x) = \frac{1}{x}, x\right) x > 0 \text{ and } x < 1$	$x = 0.860334$
--	----------------

$\text{solve}(x = x + 1, x)$	false
$\text{solve}(x = x, x)$	true

$2 \cdot x - 1 \leq 1 \text{ and } \text{solve}(x^2 \neq 9, x)$	$x \neq -3 \text{ and } x \leq 1$
---	-----------------------------------

In de hoekmodus Radialen:

$\text{solve}(\sin(x) = 0, x)$	$x = n \cdot \pi$
--------------------------------	-------------------

$\text{solve}\left(x^{\frac{1}{3}} = -1, x\right)$	$x = -1$
$\text{solve}(\sqrt{x} = -2, x)$	false
$\text{solve}(-\sqrt{x} = -2, x)$	$x = 4$

solve(*Vgl1* and *Vgl2* [and...], *VarOfGok1*,
VarOfGok2 [, ...])

⇒ *Booleaanse uitdrukking*

solve(*StelselVanVgl*, *VarOfGok1*,
VarOfGok2 [, ...])

⇒ *Booleaanse uitdrukking*

solve({*Vgl1*, *Vgl2* [...]} {*VarOfGok1*,
VarOfGok2 [, ...]})

⇒ *Booleaanse uitdrukking*

Geeft mogelijke reële oplossingen voor simultane algebraïsche vergelijkingen, waarbij elke *VarOfGok* een variabele specificeert waarnaar u wilt oplossen.

U kunt de vergelijkingen scheiden met de **and**-operator, of u kunt een *StelselVanVgl* invoeren met behulp van een template uit de Catalogus. Het aantal *VarOfGok*-argumenten moet overeenkomen met het aantal vergelijkingen. U kunt optioneel een begingok voor een variabele specificeren. Elke *VarOfGok* moet de volgende vorm hebben:

variabele

– of –

variabele = reëel of niet-reëel getal

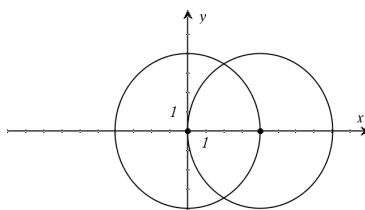
Bijvoorbeeld: x is geldig en $x=3$ ook.

Als alle vergelijkingen veeltermen zijn en als u GEEN begingokken specificeert, dan gebruikt **solve()** de lexicale Gröbner/Buchberger-eliminatiemethode om te proberen alle reële oplossingen te bepalen.

Stel dat u een cirkel heeft met straal r en het middelpunt in de oorsprong, en een andere cirkel met straal r gecentreerd waar de eerste cirkel de positieve x -as snijdt. Gebruik **solve()** om de snijpunten van de cirkels te vinden.

$$\text{solve}(y=x^2-2 \text{ and } x+2 \cdot y=-1, \{x, y\})$$

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



Zoals geïllustreerd door r in het voorbeeld rechts, kunnen simultane polynomiale vergelijkingen extra variabelen hebben die geen waarden hebben, maar die gegeven numerieke waarden representeren die later gesubstitueerd kunnen worden.

U kunt ook (of in plaats daarvan) oplossingsvariabelen opnemen die niet in de vergelijkingen verschijnen. U kunt bijvoorbeeld z opnemen als een oplossingsvariabele om het eerdere voorbeeld uit te breiden naar twee parallelle snijdende cilinders met straal r .

De cilinderoplossingen laten zien hoe families van oplossingen arbitraire constanten zouden kunnen bevatten van de vorm ck , waarbij k een geheel getal-suffix van 1 tot en met 255 is.

Bij veeltermstelsels kan de berekeningstijd of de belasting van het geheugen sterk afhangen van de volgorde waarin u oplossingsvariabelen plaatst. Als uw eerste keuze het geheugen uitput of teveel van uw geduld vraagt, probeer de variabelen in de vergelijkingen en/of *VarOfGok* dan te herschikken.

Als u geen gokken opneemt en als een vergelijking in enige variabele geen veelterm is, maar alle vergelijkingen lineair zijn in de oplossingsvariabelen, dan gebruikt **solve()** Gaussische eliminatie om te proberen alle reële oplossingen te bepalen.

Als een stelsel noch polynomiaal in al zijn variabelen, noch lineair in zijn oplossingsvariabelen is, dan bepaalt **solve()** maximaal één oplossing met behulp van een benaderende iteratieve methode. Om dit te doen moet het aantal oplossingsvariabelen gelijk zijn aan het aantal vergelijkingen, en moeten alle andere variabelen in de vergelijkingen vereenvoudigd worden tot getallen.

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ or } x=\frac{r}{2} \text{ and } y=\frac{-\sqrt{3}\cdot r}{2}$$

$$\text{solve}\left(x^2+y^2=r^2 \text{ and } (x-r)^2+y^2=r^2, \{x,y,z\}\right)$$

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=c1 \text{ or } x=\frac{r}{2} \text{ and } y\rightarrow$$

Om het hele resultaat te zien drukt u op $\blacktriangle$ en gebruikt u vervolgens $\blacktriangleleft$ en $\blacktriangleright$ om de cursor te verplaatsen.

$$\text{solve}\left(x+e^z\cdot y=1 \text{ and } x-y=\sin(z), \{x,y\}\right)$$

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

$$\text{solve}\left(e^z\cdot y=1 \text{ and } -y=\sin(z), \{y,z\}\right)$$

$$y=2.812\text{E}-10 \text{ and } z=21.9911 \text{ or } y=0.001871\text{P}$$

Om het hele resultaat te zien drukt u op $\blacktriangle$ en gebruikt u vervolgens $\blacktriangleleft$ en $\blacktriangleright$ om de cursor te verplaatsen.

solve()Catalogus > 

Elke oplossingsvariabele begint bij de gegokte waarde, als die er is; anders begint hij bij 0,0.

$$\text{solve}\left(e^z \cdot y = 1 \text{ and } \neg y = \sin(z), \{y, z = 2 \cdot \pi\}\right)$$

$$y = 0.001871 \text{ and } z = 6.28131$$

Gebruik gokken om één voor één aanvullende oplossingen te zoeken. Voor convergentie moet een gok mogelijk vrij dicht bij een oplossing liggen.

SortACatalogus > 

SortA *Lijst1* [, *Lijst2*] [, *Lijst3*] ...

 $\{2, 1, 4, 3\} \rightarrow \text{list1}$
 $\{2, 1, 4, 3\}$

SortA *Vector1* [, *Vector2*] [, *Vector3*] ...

SortA *list1* Done

Sorteert de elementen van het eerste argument in oplopende volgorde.

list1 $\{1, 2, 3, 4\}$

Als u extra argumenten opneemt, dan worden de elementen van elk daarvan gesorteerd, zodat de nieuwe posities overeenkomen met de nieuwe posities van de elementen in het eerste argument.

 $\{4, 3, 2, 1\} \rightarrow \text{list2}$
 $\{4, 3, 2, 1\}$

SortA *list2, list1* Done

Alle argumenten moeten namen van lijsten of vectoren zijn. Alle argumenten moeten gelijke afmetingen hebben.

list2 $\{1, 2, 3, 4\}$

Leg elementen binnen het eerste argument worden onderaan geplaatst. Voor meer informatie over lege elementen, zie pag. 267.

list1 $\{4, 3, 2, 1\}$

SortDCatalogus > 

SortD *Lijst1* [, *Lijst2*] [, *Lijst3*] ...

 $\{2, 1, 4, 3\} \rightarrow \text{list1}$
 $\{2, 1, 4, 3\}$

SortD *Vector1* [, *Vector2*] [, *Vector3*] ...

 $\{1, 2, 3, 4\} \rightarrow \text{list2}$
 $\{1, 2, 3, 4\}$

Identiek aan **SortA**, behalve dat **SortD** de elementen in aflopende volgorde sorteert.

SortD *list1, list2* Done

Leg elementen binnen het eerste argument worden onderaan geplaatst. Voor meer informatie over lege elementen, zie pag. 267.

list1 $\{4, 3, 2, 1\}$

list2 $\{3, 4, 1, 2\}$

Vector ►Sphere

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @>Sphere in te typen.

Geeft de rij- of de kolomvector in bolvorm weer [ρ $\angle \theta$ $\angle \phi$].

Vector moet de afmeting 3 hebben en kan een rij- of een kolomvector zijn.

Opmerking: ►Sphere is een weergave-opmaakinstructie, geen conversiefunctie. U kunt dit commando alleen op het eind van een invoerregel gebruiken.

Opmerking: Om een decimale benaderende uitkomst te forceren,

Rekenmachine: Druk op  .

Windows®: Druk op **Ctrl+Enter**.

Macintosh®: Druk op **⌘+Enter**.

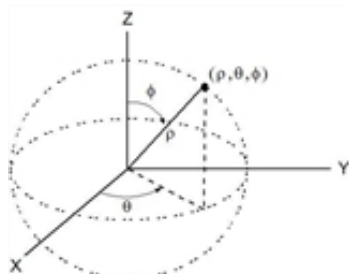
iPad®: Houd **Enter** ingedrukt en selecteer .

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \text{►Sphere} \\ \left[3.74166 \quad \angle 1.10715 \quad \angle 0.640522 \right]$$

$$\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix} \text{►Sphere} \\ \left[3.60555 \quad \angle 0.785398 \quad \angle 0.588003 \right]$$

Druk op .

$$\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix} \text{►Sphere} \\ \left[\sqrt{13} \quad \angle \frac{\pi}{4} \quad \angle \sin^{-1} \left(\frac{2 \cdot \sqrt{13}}{13} \right) \right]$$



sqrt()

sqrt(*Uitdr1*)⇒*uitdrukking*

$$\sqrt{4} \quad 2$$

sqrt(*Lijst1*)⇒*lijst*

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

Geeft de wortel van het argument.

Geeft bij een lijst de wortel van alle elementen in *Lijst1*.

Opmerking: zie ook **Wortel-template**, pag. 1.

stat.results (stat.resultaten)

Catalogus >

stat.results

Geeft resultaten van een statistische berekening weer.

De resultaten worden weergegeven als een serie naam-waarde-paren. De weergegeven specifieke namen zijn afhankelijk van de meest recent uitgewerkte statistiekfunctie of -commando.

U kunt een naam of waarde kopiëren en hem in andere locaties plakken.

Opmerking: vermijd het om variabelen te definiëren die dezelfde namen hebben als de variabelen die gebruikt worden bij statistische analyse. In bepaalde gevallen zou er dan een fout kunnen optreden. Variabelenamen die gebruikt worden voor statistische analyse staan in onderstaande tabel vermeld.

$xlist := \{1, 2, 3, 4, 5\}$	$\{1, 2, 3, 4, 5\}$
$ylist := \{4, 8, 11, 14, 17\}$	$\{4, 8, 11, 14, 17\}$
LinRegMx <i>xlist</i> , <i>ylist</i> , 1: <i>stat.results</i>	
"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r ² "	0.996109
"t"	0.998053
"Resid"	" {... } "
<i>stat.values</i>	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	" {-0.4, 0.4, 0.2, 0., -0.2} "

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBLOCK
stat.AdjR ²	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r ²	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.σx	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.σy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.σx1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.σx2	stat.UpperVal

stat.b8	stat.F	stat.n	stat.Σx	stat.Σ̄
stat.b9	stat.F Block	stat.β̂	stat.Σx ²	stat.Σ̄1
stat.b10	stat.F col	stat.β̂ 1	stat.Σxy	stat.Σ̄2
stat.bList	stat.F Interact	stat.β̂ 2	stat.Σy	stat.Σ̄ Diff
stat.χ ²	stat.FreqReg	stat.β̂ Diff	stat.Σy ²	stat.Σ̄ List
stat.c	stat.F row	stat.PList	stat.s	stat.XReg
stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.Complist	stat.LowerVal	stat.PValCol	stat.SEPred	stat.ȳ
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat.ŷ
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat.ŷ List
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

Opmerking: telkens wanneer de Lijsten & Spreadsheet-toepassing statistische resultaten berekent, kopieert deze de variabelen uit de “stat groep.” naar een groep “stat#.”, waarbij # een getal is dat automatisch toeneemt. Hierdoor kunt u eerdere resultaten behouden terwijl u meerdere berekeningen uitvoert.

stat.values

Catalogus > 

stat.values

Zie het voorbeeld
stat.results.

Geeft een matrix met de waarden die berekend zijn voor de meest recent uitgewerkte statistiekfunctie of -commando.

In tegenstelling tot **stat.results** laat **stat.values** de namen die geassocieerd zijn met de waarden weg.

U kunt een waarde kopiëren en deze op andere locaties plakken.

stDevPop()

Catalogus > 

stDevPop(Lijst[,freqLijst])⇒*uitdrukking*

In de hoekmodus Radialen en de automatisch modus:

Geeft de populatiestandaarddeviatie van de elementen in *Lijst*.

stDevPop()Catalogus > 

Elk element uit *freqLijst* telt het aantal malen dat het overeenkomstige element in *Lijst* achter elkaar voorkomt.

Opmerking: *Lijst* moet tenminste twee elementen hebben. Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

stDevPop(Matrix1[,freqMatrix])⇒matrix

Geeft een rijvector met de populatiestandaarddeviaties van de kolommen in *Matrix1*.

Elk element van *freqMatrix* telt het aantal opeenvolgende malen dat het overeenkomstige element voorkomt in *Matrix1*.

Opmerking: *Matrix1* moet tenminste twee rijen hebben. Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

$$\begin{aligned} \text{stDevPop}(\{a,b,c\}) &= \frac{\sqrt{2 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3} \\ \text{stDevPop}(\{1,2,5,-6,3,-2\}) &= \frac{\sqrt{465}}{6} \\ \text{stDevPop}(\{1.3,2.5,-6.4\},\{3,2,5\}) &= 4.11107 \\ \text{stDevPop}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right) &= \begin{bmatrix} \frac{4 \cdot \sqrt{6}}{3} & \frac{\sqrt{78}}{3} & \frac{2 \cdot \sqrt{6}}{3} \end{bmatrix} \\ \text{stDevPop}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right) &= \begin{bmatrix} 2.52608 & 5.21506 \end{bmatrix} \end{aligned}$$

stDevSamp()Catalogus > 

stDevSamp(Lijst[,freqLijst])
⇒uitdrukking

Geeft de steekproefstandaarddeviatie van de elementen in *Lijst*.

Elk element uit *freqLijst* telt het aantal malen dat het overeenkomstige element in *Lijst* achter elkaar voorkomt.

Opmerking: *Lijst* moet tenminste twee elementen hebben. Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

$$\begin{aligned} \text{stDevSamp}(\{a,b,c\}) &= \frac{\sqrt{3 \cdot (a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2)}}{3} \\ \text{stDevSamp}(\{1,2,5,-6,3,-2\}) &= \frac{\sqrt{62}}{2} \\ \text{stDevSamp}(\{1.3,2.5,-6.4\},\{3,2,5\}) &= 4.33345 \end{aligned}$$

stDevSamp()Catalogus > 

stDevSamp(*Matrix1* [, *freqMatrix*])
 $\Rightarrow$ *matrix*

Geeft een rijvector met de steekproefstandaarddeviaties van de kolommen in *Matrix1*.

Elk element van *freqMatrix* telt het aantal opeenvolgende malen dat het overeenkomstige element voorkomt in *Matrix1*.

Opmerking: *Matrix1* moet tenminste twee rijen hebben. Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

$\text{stDevSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix}\right)$	$\begin{bmatrix} 4 & \sqrt{13} & 2 \end{bmatrix}$
$\text{stDevSamp}\left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}\right)$	$\begin{bmatrix} 2.7005 & 5.44695 \end{bmatrix}$

StopCatalogus > **Stop**

Programmeringscommando: beëindigt het programma.

Stop is niet toegestaan in functies.

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

<i>i</i> :=0	0
Define <i>prog1</i> ()=Prgm	Done
For <i>i</i> ,1,10,1	
If <i>i</i> =5	
Stop	
EndFor	
EndPrgm	
<i>prog1</i> ()	Done
<i>i</i>	5

Storezie $\rightarrow$ (store), pag. 249.**string()**Catalogus > 

string(*Uitdr*) $\Rightarrow$ *string*

Vereenvoudigt *Uitdr* en geeft het resultaat als een tekenreeks.

<i>string</i> (1.2345)	"1.2345"
<i>string</i> (1+2)	"3"
<i>string</i> ($\cos(x) + \sqrt{3}$)	"cos(x)+ $\sqrt{3}$ "

subMat()Catalogus > 

subMat(*MatrixI* [, *startRij*] [, *startKol*] [, *eindRij*] [, *eindKol*]) $\Rightarrow$ *matrix*

Geeft de gespecificeerde submatrix van *MatrixI*.

Standaardinstellingen: *startRij*=1, *startKol*=1, *eindRij*=laatste rij, *eindKol*=laatste kolom.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
$\text{subMat}(m1, 2, 1, 3, 2)$	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
$\text{subMat}(m1, 2, 2)$	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)Zie $\Sigma()$, pag. 239.**sum()**Catalogus > 

sum(*Lijst* [, *Start*] [, *Eind*]) $\Rightarrow$ *uitdrukking*

Geeft de som van de elementen in *Lijst*.

Start en *Eind* zijn optioneel. Ze specificeren een bereik van elementen.

Elk leeg argument levert een leeg resultaat op. Lege elementen in *Lijst* worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

sum(*MatrixI* [, *Start*] [, *Eind*]) $\Rightarrow$ *matrix*

Geeft een rijvector met de sommen van de elementen in de kolommen van *MatrixI*.

Start en *Eind* zijn optioneel. Ze specificeren een bereik van rijen.

Elk leeg argument levert een leeg resultaat op. Lege elementen in *MatrixI* worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

$\text{sum}(\{1, 2, 3, 4, 5\})$	15
$\text{sum}(\{a, 2 \cdot a, 3 \cdot a\})$	$6 \cdot a$
$\text{sum}(\text{seq}(n, n, 1, 10))$	55
$\text{sum}(\{1, 3, 5, 7, 9\}, 3)$	21

$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right)$	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
$\text{sum}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, 2, 3\right)$	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

sumIf()Catalogus > 

sumIf(*Lijst*, *Criteria* [, *SomLijst*]) $\Rightarrow$ *waarde*

$\text{sumIf}(\{1, 2, e, 3, \pi, 4, 5, 6\}, 2.5 < ? < 4.5)$	$e + \pi + 7$
$\text{sumIf}(\{1, 2, 3, 4\}, 2 < ? < 5, \{10, 20, 30, 40\})$	70

Geeft de cumulatieve som van alle elementen in *Lijst* die voldoen aan de gespecificeerde *Criteria*. Optioneel kunt u een alternatieve lijst specificeren, *somLijst*, om de elementen te leveren die opgeteld moeten worden.

Lijst kan een uitdrukking, een lijst of een matrix zijn. *SomLijst*, indien gespecificeerd, moet dezelfde afmeting(en) hebben als *Lijst*.

Criteria kan zijn:

- Een waarde, uitdrukking of tekenreeks. Bijvoorbeeld: **34** telt alleen die elementen in *Lijst* op die vereenvoudigd worden tot de waarde 34.
- Een Booleaanse uitdrukking met het symbool **?** als plaatsaanduiding voor elk element. Bijvoorbeeld, **?<10** telt alleen die elementen in *Lijst* op die kleiner zijn dan 10.

Als een *Lijst*-element voldoet aan de *Criteria*, dan wordt het element opgeteld bij de cumulatieve som. Als u *somLijst* opneemt, dan wordt in plaats daarvan het overeenkomstige element van *somLijst* bij de som opgeteld.

In de toepassing Lijsten & Spreadsheet kunt u een reeks cellen gebruiken op de plaats van *Lijst* en *somLijst*.

Lege elementen worden genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

Opmerking: zie ook `countIf()`, pag. 37.

system()Catalogus > **system**(*Vgl1* [, *Vgl2* [, *Vgl3* [, ...]]])

$$\text{solve}\left\{\begin{matrix} x+y=0 \\ x-y=8 \end{matrix}, x, y\right\} \quad x=4 \text{ and } y=-4$$

system(*Uitdr1* [, *Uitdr2* [, *Uitdr3* [, ...]]])

Geeft een stelsel vergelijkingen, in de vorm van een lijst. U kunt ook een stelsel creëren met behulp van een template.

Opmerking: zie ook **Stelsel van vergelijkingen**, pag. 3.

T**T (transponeren)**Catalogus > *Matrix* **T** $\Rightarrow$ *matrix*

Geeft de complex geconjugeerde transponering van *Matrix* *l*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @t in te typen.

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

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^T \quad \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

$$\begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}^T \quad \begin{bmatrix} 1-i & 3-i \\ 2-i & 4-i \end{bmatrix}$$

tan() **-toets****tan**(*Uitdr1*) $\Rightarrow$ *uitdrukking*

In de hoekmodus Graden:

tan(*Lijst1*) $\Rightarrow$ *lijst*

tan(*Uitdr1*) geeft de tangens van het argument als een uitdrukking.

tan(*Lijst1*) geeft een lijst met de tangensen van alle elementen in *Lijst1*.

Opmerking: het argument wordt geïnterpreteerd als een hoek in graden, in decimale graden of in radialen, volgens de ingestelde hoekmodus. U kunt $^{\circ}$, G of r gebruiken om de instelling van de hoekmodus tijdelijk te onderdrukken.

In de hoekmodus Decimale graden:

$$\tan\left(\frac{\pi}{4}^r\right) \quad 1$$

$$\tan(45^{\circ}) \quad 1$$

$$\tan(\{0,60,90\}) \quad \{0,\sqrt{3},\text{undef}\}$$

$$\tan\left(\frac{\pi}{4}\right) \quad 1$$

$$\tan(50^G) \quad 1$$

$$\tan(\{0,50,100\}) \quad \{0,1,\text{undef}\}$$

In de hoekmodus Radialen:

$\tan\left(\frac{\pi}{4}\right)$	1
$\tan(45^\circ)$	1
$\tan\left(\left\{\pi, \frac{\pi}{3}, \pi, \frac{\pi}{4}\right\}\right)$	$\{0, \sqrt{3}, 0, 1\}$

tan(vierkanteMatrix1) ⇒ vierkanteMatrix

Geeft de matrixtangens van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de tangens van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Radialen:

$\tan\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$	$\begin{bmatrix} -28.2912 & 26.0887 & 11.1142 \\ 12.1171 & -7.83536 & -5.48138 \\ 36.8181 & -32.8063 & -10.4594 \end{bmatrix}$
--	--

tan⁻¹()

tan⁻¹(Uitdr1) ⇒ uitdrukking

tan⁻¹(Lijst1) ⇒ lijst

tan⁻¹(Uitdr1) geeft de hoek waarvan de tangens *Uitdr1* is als een uitdrukking.

tan⁻¹(Lijst1) geeft een lijst met de inverse tangens van elk element in *Lijst1*.

Opmerking: de uitkomst wordt in graden, decimale graden of radialen gegeven, volgens de ingestelde hoekmodus.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arctan(...)** in te typen.

tan⁻¹(vierkanteMatrix1) ⇒ vierkanteMatrix

Geeft de inverse matrixtangens van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de inverse tangens van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Graden:

$\tan^{-1}(1)$	45
----------------	----

In de hoekmodus Decimale graden:

$\tan^{-1}(1)$	50
----------------	----

In de hoekmodus Radialen:

$\tan^{-1}(\{0, 0.2, 0.5\})$	$\{0, 0.197396, 0.463648\}$
------------------------------	-----------------------------

In de hoekmodus Radialen:

$\tan^{-1}\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$	$\begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$
---	---

tangentLine()Catalogus > 

tangentLine(Uitdr1,Var,Punt)
 $\Rightarrow$ uitdrukking

tangentLine(Uitdr1,Var=Punt)
 $\Rightarrow$ uitdrukking

Geeft de raaklijn aan de kromme die gerepresenteerd wordt door *Uitdr1* in het punt dat gespecificeerd is door *Var=Punt*.

Zorg ervoor dat de onafhankelijke variabele niet gedefinieerd is. Bijvoorbeeld: als $f_1(x) := 5$ en $x := 3$, dan geeft **tangentLine**($f_1(x), x, 2$) "false."

$\text{tangentLine}(x^2, x, 1)$	$2 \cdot x - 1$
$\text{tangentLine}((x-3)^2 - 4, x=3)$	-4
$\text{tangentLine}\left(\frac{1}{x^3}, x=0\right)$	$x=0$
$\text{tangentLine}(\sqrt{x^2-4}, x=2)$	undef
$x:=3; \text{tangentLine}(x^2, x, 1)$	5

tanh()Catalogus > 

tanh(Uitdr1) $\Rightarrow$ uitdrukking

tanh(Lijst1) $\Rightarrow$ lijst

tanh(Uitdr1) geeft de tangens hyperbolicus van het argument als een uitdrukking.

tanh(Lijst1) geeft een lijst met de tangens hyperbolicus van elk element in *Lijst1*.

tanh(vierkanteMatrix1) $\Rightarrow$ vierkanteMatrix

Geeft de matrixtangens hyperbolicus van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de tangens hyperbolicus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

$\tanh(1.2)$	0.833655
$\tanh(\{0, 1\})$	$\{0, \tanh(1)\}$

In de hoekmodus Radialen:

$\tanh\begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix}$	$\begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$		
---	---	--	--

tanh⁻¹()Catalogus > 

tanh⁻¹(Uitdr1) $\Rightarrow$ uitdrukking

tanh⁻¹(Lijst1) $\Rightarrow$ lijst

tanh⁻¹(Uitdr1) geeft de inverse tangens hyperbolicus van het argument als een uitdrukking.

In rechthoekige complexe opmaak:

$\tanh^{-1}(0)$	0
$\tanh^{-1}(\{1, 2, 1, 3\})$	$\left\{ \text{undef}, 0.518046 - 1.5708 \cdot i, \frac{\ln(2)}{2} - \frac{\pi}{2} \cdot i \right\}$

tanh⁻¹(LijstI) geeft een lijst van de inverse tangens hyperbolicus van elk element in *LijstI*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **arctanh(...)** in te typen.

tanh⁻¹(vierkanteMatrixI)
 $\Rightarrow$ vierkanteMatrix

Geeft de inverse matrixtangens hyperbolicus van *vierkanteMatrixI*. Dit is niet hetzelfde als het berekenen van de inverse tangens hyperbolicus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

vierkanteMatrixI moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.

In de hoekmodus Radialen en rechthoekige complexe opmaak:

$$\tanh^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} -0.099353+0.164058 \cdot i & 0.267834-1.4908 \\ -0.087596-0.725533 \cdot i & 0.479679-0.94730 \\ 0.511463-2.08316 \cdot i & -0.878563+1.7901 \end{bmatrix}$$

Om het hele resultaat te zien drukt u op $\blacktriangle$ en gebruikt u vervolgens $\blacktriangleleft$ en $\blacktriangleright$ om de cursor te verplaatsen.

taylor(UitdrI, Var, Orde[, Punt])
 $\Rightarrow$ uitdrukking

Geeft de gevraagde Taylor-polynoom. De polynoom bevat niet-nul-termen van gehele graden van nul tot *Orde* in (*Var* min *Punt*).

taylor() geeft zichzelf terug als er geen ingekorte machtsenserie van deze orde is, of als er negatieve of gebroken exponenten nodig zouden zijn. Gebruik substitutie en/of tijdelijke vermenigvuldiging met een macht van (*Var* min *Punt*) om algemene machtsenseries te bepalen.

Punt heeft als standaardwaarde nul en is het uitbreidingspunt.

$$\begin{array}{ll} \text{taylor}(e^{\sqrt{x}}, x, 2) & \text{taylor}(e^{\sqrt{x}}, x, 2, 0) \\ \text{taylor}(e^t, t, 4) | t = \sqrt{x} & \frac{3}{24} + \frac{x^2}{6} + \frac{x}{2} + \sqrt{x} + 1 \\ \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right) & \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right) \\ \text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}, x\right) & -x^3 - x^2 - x - \frac{1}{x} - 1 \end{array}$$

tCdf(ondergrens, bovengrens, df) $\Rightarrow$ getal als *ondergrens* en *bovengrens* getallen zijn, *lijst* als *ondergrens* en *bovengrens* lijsten zijn

Berekent de Student-*t*-verdelingskans tussen *ondergrens* en *bovengrens* bij de gespecificeerde vrijheidsgraden *df*.

Voor $P(X \leq \text{bovengrens})$ stelt u *ondergrens* = $-\infty$.

tCollect()

tCollect(Uitdr1) ⇒ *uitdrukking*

Geeft een uitdrukking waarin de producten en gehele machten van sinussen en cosinussen geconverteerd worden naar een lineaire combinatie van sinussen en cosinussen van meervoudige hoeken, hoeksommen en hoekverschillen. De transformatie converteert goniometrische veeltermen in een lineaire combinatie van hun harmonische elementen.

Soms bereikt u met **tCollect()** uw doelen als dit met de standaard goniometrische vereenvoudiging niet lukt. **tCollect()** keert vaak transformaties die uitgevoerd zijn door **tExpand()** om. Soms kan het toepassen van **tExpand()** op een resultaat uit **tCollect()**, of andersom, in twee aparte stappen, een uitdrukking vereenvoudigen.

$$\begin{array}{l} \text{tCollect}(\{\cos(\alpha)\}^2) \quad \frac{\cos(2 \cdot \alpha) + 1}{2} \\ \text{tCollect}(\sin(\alpha) \cdot \cos(\beta)) \quad \frac{\sin(\alpha - \beta) + \sin(\alpha + \beta)}{2} \end{array}$$

tExpand()

tExpand(Uitdr1) ⇒ *uitdrukking*

Geeft een uitdrukking waarin sinussen en cosinussen van gehele meervoudige hoeken, hoeksommen en hoekverschillen uitgewerkt worden. Vanwege de gelijkheid $(\sin(x))^2 + (\cos(x))^2 = 1$ zijn er vele equivalente resultaten mogelijk. Daarom kan een resultaat verschillen van een resultaat dat weergegeven wordt in andere publicaties.

$$\begin{array}{l} \text{tExpand}(\sin(3 \cdot \phi)) \quad 4 \cdot \sin(\phi) \cdot \{\cos(\phi)\}^2 - \sin(\phi) \\ \text{tExpand}(\cos(\alpha - \beta)) \quad \frac{\cos(\alpha) \cdot \cos(\beta) + \sin(\alpha) \cdot \sin(\beta)}{} \end{array}$$

Soms bereikt u met **tExpand()** uw doelen als dit met de standaard goniometrische vereenvoudiging niet lukt. **tExpand()** keert vaak transformaties die uitgevoerd zijn door **tCollect()** om. Soms kan het toepassen van **tCollect()** op een resultaat uit **tExpand()**, of andersom, in twee aparte stappen, een uitdrukking vereenvoudigen.

Opmerking: schaling in de graden-modus met $\pi/180$ interfereert met de mogelijkheid van **tExpand()** om uitwerkbare vormen te herkennen. Om de beste resultaten te krijgen moet **tExpand()** in de radialen-modus gebruikt worden.

Text

TextpromptString[, ToonVlag]

Programmeringscommando: Pauzeert het programma en geeft de tekenreeks *promptString* in een dialoogvenster weer.

Als de gebruiker **OK** selecteert, gaat het programma verder.

Het optionele argument *vlag* kan elke willekeurige uitdrukking zijn.

- Als *ToonVlag* wordt weggelaten of wordt uitgewerkt tot **1**, dan wordt het tekstbericht toegevoegd aan de Rekenmachinegeschiedenis.
- Als *ToonVlag* wordt uitgewerkt tot **0**, dan wordt het tekstbericht niet toegevoegd aan de geschiedenis.

Als het programma een getypte respons van de gebruiker nodig heeft, zie dan **Request**, pag. 158 of **RequestStr**, pag. 160.

Opmerking: u kunt dit commando binnen een door de gebruiker gedefinieerd programma gebruiken, maar niet binnen een functie.

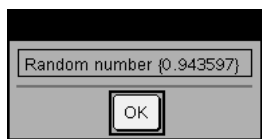
Definieer een programma dat pauzeert om vijf verschillende toevalsgetallen in een dialoogvenster weer te geven.

Maak binnen de template Prgm...EndPrgm elke regel af door op  in plaats van op **enter** te drukken. Op het toetsenbord van de computer houdt u **Alt** ingedrukt en drukt u op **Enter**.

```
Define text_demo()=Prgm
  For i,1,5
    strinfo:="Random number
" & string(rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Voer het programma uit:
text_demo()

Voorbeeld van een
dialoogvenster:

**tInterval**

tInterval *Lijst[,Freq[,CNiveau]]*

(Invoer van een gegevenslijst)

tInterval $\bar{x}, S_x, n[, CNiveau]$

(Invoer van samenvattingsstatistieken)

Berekent een *t*-betrouwbaarheidsinterval. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.CLower, stat.CUpper	Betrouwbaarheidsinterval voor een onbekend populatiegemiddelde
stat. $\bar{x}$	Steekproefgemiddelde van de gegevensverzameling uit een normale willekeurige verdeling
stat.ME	Foutmarge
stat.df	Vrijheidsgraden
stat. σ_x	Standaarddeviatie steekproef
stat.n	Lengte van de gegevensverzameling met het steekproefgemiddelde

tInterval_2Samp

tInterval_2Samp *Lijst1,Lijst2[,Freq1[,Freq2]*

[,CNiveau[,Gepoold]]])

(Invoer van een gegevenslijst)

tInterval_2Samp $\bar{x}1, s_{x1}, n1, \bar{x}2, s_{x2}, n2$ [,CNiveau
[,Gepoold]]

(Invoer van samenvattingsstatistieken)

Berekent een *t*-betrouwbaarheidsinterval met twee steekproeven. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Gepoold=1 poolt de varianties; *Gepoold=0* poolt de varianties niet.

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.CLower, stat.CUpper	Betrouwbaarheidsinterval met de betrouwbaarheidskans gebaseerd op de verdeling
stat. $\bar{x}1$ - $\bar{x}2$	Steekproefgemiddelden van de gegevensverzameling uit de willekeurige normale verdeling
stat.ME	Foutmarge
stat.df	Vrijheidsgraden
stat. $\bar{x}1$, stat. $\bar{x}2$	Steekproefgemiddelden van de gegevensverzameling uit de willekeurige normale verdeling
stat. $\sigma x1$, stat. $\sigma x2$	Steekproefstandaarddeviaties voor <i>Lijst 1</i> en <i>Lijst 2</i>
stat.n1, stat.n2	Aantal steekproeven in de gegevensverzamelingen
stat.sp	De gepoolde standaarddeviatie. Berekend wanneer <i>Gepoold</i> = JA.

tmpCnv()

tmpCnv(*Uitdr*, *tempEenheid*, *tempEenheid2*) $\Rightarrow$ *uitdrukking* *tempEenheid2*

Converteert een temperatuurwaarde die gespecificeerd is door *Uitdr* van de ene eenheid naar een andere. Geldige temperatuureenheden zijn:

_°CCelsius

tmpCnv(100·_°C,_°F)	212·_°F
tmpCnv(32·_°F,_°C)	0·_°C
tmpCnv(0·_°C,_°K)	273.15·_°K
tmpCnv(0·_°F,_°R)	459.67·_°R

Opmerking: u kunt de Catalogus gebruiken om temperatuureenheden te selecteren.

_°FFahrenheit

_°KKelvin

_°RRankine

Om ° te typen selecteert u dit uit de symbolenlijst in de Catalogus.

Om _ te typen drukt u op  .

Bijvoorbeeld: 100_°C wordt geconverteerd naar 212_°F.

Om een temperatuurbereik te converteren gebruikt u Δ tmpCnv().

Δ tmpCnv()

Δ tmpCnv(Uitdr °tempEenheid, _ °tempEenheid2) ⇒ uitdrukking _ °tempEenheid2

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **del taTmpCnv (...)** in te typen.

Converteert een temperatuurbereik (het verschil tussen twee temperatuurwaarden) dat gespecificeerd is door *Uitdr* van de ene eenheid naar een andere. Geldige temperatuureenheden zijn:

_°CCelsius

_°FFahrenheit

_°KKelvin

_°RRankine

Om ° in te voeren selecteert u dit teken in het symboolpalet of typt u @d.

Om _ te typen drukt u op  .

1_°C en 1_°K hebben dezelfde grootte, net als 1_°F en 1_°R. 1_°C is echter 9/5 maal zo groot als 1_°F.

Om Δ te typen selecteert u dit uit de symbolenlijst in de Catalogus.

Δ tmpCnv(100_°C,_°F)	180_°F
Δ tmpCnv(180_°F,_°C)	100_°C
Δ tmpCnv(100_°C,_°K)	100_°K
Δ tmpCnv(100_°F,_°R)	100_°R
Δ tmpCnv(1_°C,_°F)	1.8_°F

Opmerking: u kunt de Catalogus gebruiken om temperatuureenheden te selecteren.

Bijvoorbeeld: een bereik van 100_°C (van 0_°C tot 100_°C) is hetzelfde als een bereik van 180_°F.

Om een bepaalde temperatuurwaarde in plaats van een bereik te converteren gebruikt u **tmpCnv()**.

tPdf()

tPdf(XWaarde,df)⇒getal als *XWaarde* een getal is,
lijst als *XWaarde* een lijst is

Berekent de kansdichtheidsfunctie (pdf) voor de Student-*t*-verdeling bij een gespecificeerde *x*-waarde met de gespecificeerde vrijheidsgraden *df*.

trace()

trace(vierkante matrix)⇒uitdrukking

Geeft het spoor (som van alle elementen van de hoofddiagonaal) van *vierkanteMatrix*.

$\text{trace}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}\right)$	15
$\text{trace}\left(\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}\right)$	2· <i>a</i>

Try

Try
blok1
Else
blok2
EndTry

Voert *blok1* uit tenzij er een fout optreedt. De uitvoering van het programma gaat over naar *blok2* als er een fout optreedt in *blok1*. Systeemvariabele *errCode* bevat de foutcode zodat het programma fouterherstel kan uitvoeren. Zie “Foutcodes en meldingen”, pag. 277 voor een lijst met foutcodes.

blok1 en *blok2* kunnen een enkele bewering of een serie beweringen zijn die gescheiden worden door het teken “:”.

Define <i>prog1()</i> =Prgm	
Try	
z:=z+1	
Disp "z incremented."	
Else	
Disp "Sorry, z undefined."	
EndTry	
EndPrgm	
	Done
z:=1:prog1()	
	z incremented.
	Done
DelVar z:prog1()	
	Sorry, z undefined.
	Done

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Om de commando's **Try**, **ClrErr** en **PassErr** in werking te zien, voert u het `eigenvals()` programma in dat rechts wordt weergegeven. Voer het programma uit door elk van de volgende uitdrukkingen uit te voeren.

$$\text{eigenvals}\left(\begin{bmatrix} -3 \\ -41 \\ 5 \end{bmatrix}, \begin{bmatrix} -1 & 2 & -3.1 \end{bmatrix}\right)$$

$$\text{eigenvals}\left(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

Opmerking: zie ook **ClrErr**, pag. 27 en **PassErr**, pag. 139.

Define `eigenvals(a,b)=Prgm`

© Het programmeren van `eigenvals(A,B)` geeft de eigenwaarden van $A \cdot B$ weer

Try

Disp "A= ",a

Disp "B= ",b

Disp " "

Disp "Eigenwaarden van A·B zijn:",eigVl(a*b)

Else

If `errCode=230` Then

Disp "Fout: Product van A·B moet een vierkante matrix zijn"

ClrErr

Else

PassErr

EndIf

EndTry

EndPrgm

tTest

tTest $\mu 0, \text{Lijst}[, \text{Freq}[, \text{Hypoth}]]$

(Invoer van een gegevenslijst)

tTest $\mu 0, \bar{x}, s_x, n, [\text{Hypoth}]$

(Invoer van samenvattingsstatistieken)

Voert een hypothesetoets uit voor één onbekend populatiegemiddelde, μ , wanneer de populatiestandaarddeviate, σ , onbekend is. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Toets $H_0: \mu = \mu_0$ tegen één van de volgende alternatieven:

Voor $H_1: \mu < \mu_0$ stelt u *Hypoth*<0 in

Voor $H_1: \mu \neq \mu_0$ (standaardinstelling) stelt u *Hypoth*=0 in

Voor $H_1: \mu > \mu_0$ stelt u *Hypoth*>0 in

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.t	$(\bar{x} - \mu_0) / (\text{stdev} / \sqrt{n})$
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.df	Vrijheidsgraden
stat. $\bar{x}$	Steekproefgemiddelde van de gegevensverzameling in <i>Lijst</i>
stat.sx	Steekproefstandaarddeviatie van de gegevensverzameling
stat.n	Omvang van de steekproef

tTest_2Samp

tTest_2Samp *Lijst1,Lijst2[,Freq1[,Freq2[,Hypoth [,Gepoold]]]]*

(Invoer van een gegevenslijst)

tTest_2Samp $\bar{x}1,sx1,n1,\bar{x}2,sx2,n2[,Hypoth [,Gepoold]]$

(Invoer van samenvattingsstatistieken)

Berekent een *t*-toets met twee steekproeven. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Toets $H_0: \mu_1 = \mu_2$ tegen een van de volgende alternatieven:

Voor $H_1: \mu_1 < \mu_2$ stelt u *Hypoth*<0 in

Voor $H_1: \mu_1 \neq \mu_2$ (standaardinstelling) stelt u $Hypoth=0$ in

Voor $H_1: \mu_1 > \mu_2$ stelt u $Hypoth>0$ in

$Gepoold=1$ poolt de varianties

$Gepoold=0$ poolt de varianties niet

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.t	Standaard normale waarde berekend voor het verschil tussen de gemiddelden
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.df	Vrijheidsgraden voor de t-statistiek
stat.x̄1, stat.x̄2	Steekproefgemiddelden van de gegevensverzamelingen in <i>Lijst 1</i> en <i>Lijst 2</i>
stat.sx1, stat.sx2	Steekproefstandaarddeviaties van de gegevensverzamelingen in <i>Lijst 1</i> en <i>Lijst 2</i>
stat.n1, stat.n2	Grootte van de steekproeven
stat.sp	De gepoolde standaarddeviatie. Berekend wanneer $Gepoold=1$.

tvmFV()

tvmFV($N, I, PV, Pmt, [PpY], [CpY], [PmtAt]$)
 $\Rightarrow$ waarde

tvmFV(120,5,0,-500,12,12) 77641.1

Financiële functie die de toekomstige waarde van geld berekent.

Opmerking: de argumenten die in de TVM-functies worden gebruikt, worden beschreven in de tabel met TVM-argumenten, pag. 208. Zie ook **amortTbl()**, pag. 8.

tvmI()

tvmI($N, PV, Pmt, FV, [PpY], [CpY], [PmtAt]$)
 $\Rightarrow$ waarde

tvmI(240,100000,-1000,0,12,12) 10.5241

Financiële functie die het rentepercentage per jaar berekent.

Opmerking: de argumenten die in de TVM-functies worden gebruikt, worden beschreven in de tabel met TVM-argumenten, pag. 208. Zie ook **amortTbl()**, pag. 8.

tvmN()

tvmN(*I,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*)
⇒ *waarde*

tvmN(5,0,-500,77641,12,12)	120.
----------------------------	------

Financiële functie die het aantal betalingsperioden berekent.

Opmerking: de argumenten die in de TVM-functies worden gebruikt, worden beschreven in de tabel met TVM-argumenten, pag. 208. Zie ook **amortTbl()**, pag. 8.

tvmPmt()

tvmPmt(*N,I,PV,FV,[PpY],[CpY],[PmtAt]*)
⇒ *waarde*

tvmPmt(60,4,30000,0,12,12)	-552.496
----------------------------	----------

Financiële functie die het bedrag van elke betaling berekent.

Opmerking: de argumenten die in de TVM-functies worden gebruikt, worden beschreven in de tabel met TVM-argumenten, pag. 208. Zie ook **amortTbl()**, pag. 8.

tvmPV()

tvmPV(*N,I,Pmt,FV,[PpY],[CpY],[PmtAt]*)
⇒ *waarde*

tvmPV(48,4,-500,30000,12,12)	-3426.7
------------------------------	---------

Financiële functie die de contante waarde berekent.

Opmerking: de argumenten die in de TVM-functies worden gebruikt, worden beschreven in de tabel met TVM-argumenten, pag. 208. Zie ook **amortTbl()**, pag. 8.

TVM-argument*	Beschrijving	Gegevenstype
N	Aantal betalingsperioden	reëel getal
I	Rentepercentage per jaar	reëel getal
PV	Contante waarde	reëel getal
Pmt	Betalingsbedrag	reëel getal
FV	Toekomstige waarde	reëel getal
PpY	Betalingen per jaar, standaardinstelling=1	geheel getal > 0
CpY	Rentetermijnen per jaar, standaardinstelling=1	geheel getal > 0
PmtAt	Betaling vindt plaats aan het begin of op het eind van elke periode, standaardinstelling=eind	geheel getal (0=einde, 1=begin)

* Deze tijdwaarde-van-geld-argumentnamen zijn gelijk aan de TVM-variabelenamen (zoals **tvm.pv** en **tvm.pmt**) die gebruikt worden door de financiële oplosser van de *Rekenmachine*. Financiële functies slaan hun argumentwaarden of resultaten echter niet op naar de TVM-variabelen.

TwoVar

Catalogus > 

TwoVar *X*, *Y* [, [*Freq*] [, *Categorie*, *Opnemen*]]

Berekent de statistieken voor twee variabelen. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 188).

Alle lijsten moeten gelijke afmetingen hebben, behalve *Opnemen*.

X en *Y* zijn lijsten met onafhankelijke en afhankelijke variabelen.

Freq is een optionele lijst met frequentiewaarden. Elk element in *Freq* specificeert de frequentie waarmee elk overeenkomstig *X*- en *Y*-gegeven voorkomt. De standaardwaarde is 1. Alle elementen moeten gehele getallen ≥ 0 zijn.

Categorie is een lijst met categoriecodes voor de overeenkomstige *X*- en *Y*-gegevens..

Opnemen is een lijst met één of meer van de categoriecodes. Alleen de gegevens waarvan de categoriecode is opgenomen in deze lijst worden opgenomen in de berekening.

Een leeg element in een van de lijsten X , $Freq$ of $Categorie$ resulteert in een lege plaats voor het overeenkomstige element in al deze lijsten. Een leeg element in een van de lijsten $X1$ tot en met $X20$ resulteert in een lege plaats voor het overeenkomstige element in al deze lijsten. Voor meer informatie over lege elementen, zie pag. 267.

Uitvoervariabele	Beschrijving
stat. $\bar{x}$	Gemiddelde van de x-waarden
stat. x	Som van de x-waarden
stat. x2	Som van de x2-waarden
stat.sx	Steekproef-standaarddeviatie van x
stat. x	Populatie-standaarddeviatie van x
stat.n	Aantal gegevens
stat. $\bar{y}$	Gemiddelde van y-waarden
stat. y	Som van de y-waarden
stat. y^2	Som van de y^2 -waarden
stat.sy	Steekproefstandaarddeviatie van y
stat. y	Populatiestandaarddeviatie van y
stat. xy	Som van de $x \cdot y$ -waarden
stat.r	Correlatiecoëfficiënt
stat.MinX	Minimum van de x-waarden
stat.Q ₁ X	1ste kwartiel van x
stat.MedianX	Mediaan van x
stat.Q ₃ X	3de kwartiel van x
stat.MaxX	Maximum van de x-waarden
stat.MinY	Minimum van de y-waarden
stat.Q ₁ Y	1ste kwartiel van y
stat.MedY	Mediaan van y
stat.Q ₃ Y	3de kwartiel van y
stat.MaxY	Maximum van y-waarden

Uitvoervariabele	Beschrijving
stat. $(x-)^2$	Som van de kwadraten van de afwijkingen ten opzichte van het gemiddelde van x
stat. $(y-)^2$	Som van de kwadraten van afwijkingen ten opzichte van het gemiddelde van y

U

unitV()
Catalogus > 

unitV(*Vector1*)⇒*vector*

Geeft een rij- of kolom-eenheidsvector, afhankelijk van de vorm van *Vector1*.

Vector1 moet een matrix met één rij of een matrix met één kolom zijn.

$$\text{unitV}\left(\begin{bmatrix} a & b & c \end{bmatrix}\right)$$

$$\left[\frac{a}{\sqrt{a^2+b^2+c^2}} \quad \frac{b}{\sqrt{a^2+b^2+c^2}} \quad \frac{c}{\sqrt{a^2+b^2+c^2}} \right]$$

$$\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right)$$

$$\left[\frac{\sqrt{6}}{6} \quad \frac{\sqrt{6}}{3} \quad \frac{\sqrt{6}}{6} \right]$$

$$\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right)$$

$$\begin{bmatrix} \frac{\sqrt{14}}{14} \\ \frac{\sqrt{14}}{7} \\ \frac{3\sqrt{14}}{14} \end{bmatrix}$$

Om het hele resultaat te zien drukt u op  en gebruikt u vervolgens  en  om de cursor te verplaatsen.

unLock
Catalogus > 

unLockVar1[, Var2] [, Var3] ...

unLockVar.

Ontgrendelt de gespecificeerde variabelen of variabelegroep. Vergrendelde variabelen kunnen niet worden gewijzigd of gewist.

Zie **Lock**, pag. 112 en **getLockInfo()**, pag. 88.

$a:=65$
65

Lock *a*
Done

getLockInfo(*a*)
1

$a:=75$
"Error: Variable is locked."

DelVar *a*
"Error: Variable is locked."

Unlock *a*
Done

$a:=75$
75

DelVar *a*
Done

varPop()

Catalogus >

varPop(Lijst[,freqLijst])⇒*uitdrukking*

Geeft de populatievariantie van *Lijst*.

Elk element uit *freqLijst* telt het aantal malen dat het overeenkomstige element in *Lijst* voorkomt.

Opmerking: *Lijst* moet minimaal twee elementen bevatten.

Als een element in een van beide lijsten leeg is, wordt dat element genegeerd, en wordt het overeenkomstige element in de andere lijst ook genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

varPop({5,10,15,20,25,30})	875
	12
Ans:1.	72.9167

varSamp()

Catalogus >

varSamp(Lijst[,freqLijst])⇒*uitdrukking*

Geeft de steekproefvariantie van *Lijst*.

Elk element uit *freqLijst* telt het aantal malen dat het overeenkomstige element in *Lijst* voorkomt.

Opmerking: *Lijst* moet minimaal twee elementen bevatten.

Als een element in een van beide lijsten leeg is, wordt dat element genegeerd, en wordt het overeenkomstige element in de andere lijst ook genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

varSamp(Matrix1[,freqMatrix])⇒*matrix*

Geeft een rijvector met de steekproefvariantie van elke kolom in *Matrix1*.

Elk element van *freqMatrix* telt het aantal opeenvolgende keer dat het overeenkomstige element voorkomt in *Matrix1*.

varSamp({a,b,c})	
	$\frac{a^2 - a \cdot (b+c) + b^2 - b \cdot c + c^2}{3}$
varSamp({1,2,5,6,3,2})	31
	2
varSamp({1,3,5},{4,6,2})	68
	33

varSamp($\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}$)	[4.75 1.03 4]
varSamp($\begin{bmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{bmatrix}, \begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}$)	[3.91731 2.08411]

Opmerking: *Matrix1* moet minimaal twee rijen bevatten.

Als een element in een van beide matrices leeg is, wordt dat element genegeerd, en wordt het overeenkomstige element in de andere matrix ook genegeerd. Voor meer informatie over lege elementen, zie pag. 267.

W

Wait

Wait *tijdInSeconden*

Stelt uitvoering uit voor de duur van *tijdInSeconden* seconden.

Wait is vooral handig in een programma dat een korte vertraging nodig heeft om aangevraagde gegevens beschikbaar te maken.

Het argument *tijdInSeconden* moet een uitdrukking zijn die vereenvoudigt tot een decimale waarde tussen 0 tot 100. De opdracht rondt deze waarde naar boven af op de dichtstbijzijnde 0,1 seconde.

Voor het annuleren van een **Wait**-opdracht die in uitvoering is,

- **Rekenmachine:** Houd de toets  ingedrukt en druk enkele malen op .
- **Windows®:** Houd **F12** ingedrukt en druk enkele malen op **Enter**.
- **Macintosh®:** Houd **F5** ingedrukt en druk enkele malen op **Enter**.
- **iPad®:** De app toont een instructie. U kunt blijven wachten of annuleren.

Opmerking: U kunt de opdracht **Wait** binnen een door de gebruiker gedefinieerd programma gebruiken, maar niet binnen een functie.

Om 4 seconden te wachten:

Wait 4

Om 1/2 seconde te wachten:

Wait 0.5

Om 1,3 seconden te wachten met gebruik van de variabele *seccount*:

seccount:=1.3

Wait seccount

Dit voorbeeld schakelt gedurende 0,5 seconden een groen led-lampje in en schakelt het vervolgens uit.

Send "SET GREEN 1 ON"

Wait 0.5

Send "SET GREEN 1 OFF"

warnCodes(*Uitdr1*, *StatusVar*)⇒ *uitdrukking*

Werkt uitdrukking *Uitdr1* uit, geeft het resultaat en slaat de codes van eventuele gegenereerde waarschuwingen op in de lijstvariabele *StatusVar*. Als er geen waarschuwingen gegenereerd zijn, dan wijst deze functie aan *StatusVar* een lege lijst toe.

Uitdr1 kan elke geldige wiskundige uitdrukking in TI-Nspire™ of TI-Nspire™ CAS zijn. U kunt geen commando of taak als *Uitdr1* gebruiken.

StatusVar moet een geldige variabelenaam zijn.

Zie pag. 285 voor een lijst met waarschuwingscodes en bijbehorende berichten.

	$\text{warnCodes}\left(\text{solve}\left(\sin(10 \cdot x) = \frac{x^2}{x}, x\right), \text{warn}\right)$
	$x = -0.84232 \text{ or } x = -0.706817 \text{ or } x = -0.2852$
<i>warn</i>	$\{10007, 10009\}$

Om het hele resultaat te zien drukt u op ▲ en gebruikt u vervolgens ◀ en ▶ om de cursor te verplaatsen.

when()**when**(*Conditie*, *waarResultaat* [, *onwaarResultaat* [, *onbekendResultaat*])⇒ *uitdrukking*

Geeft *waarResultaat*, *onwaarResultaat* of *onbekendResultaat*, afhankelijk van of *Conditie* waar, onwaar of onbekend is. Geeft de invoer terug als er te weinig argumenten zijn om het betreffende resultaat te specificeren.

Laat zowel *onwaarResultaat* als *onbekendResultaat* weg om voor een uitdrukking te zorgen die alleen gedefinieerd is in het gebied waarin *Conditie* waar is.

Gebruik een **undef** *onwaarResultaat* om een uitdrukking te definiëren waarvan alleen op een interval de grafiek getekend wordt.

$\text{when}(x < 0, x + 3) x = 5$	undef
-------------------------------------	-------

Catalogus >

when() is nuttig voor het definiëren van recursieve functies.

$\text{when}(n > 0, n \cdot \text{factorial}(n-1), 1) \rightarrow \text{factorial}(n)$	<i>Done</i>
$\text{factorial}(3)$	6
$3!$	6

Catalogus >

```
While Conditie
    Blok
EndWhile
```

Voert de beweringen in *Blok* uit zolang *Conditie* waar is.

Blok kan een enkele bewering of een reeks beweringen zijn die gescheiden worden door het teken “:”.

Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Define $sum_of_recip(n) = \text{Func}$	
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	
$sum_of_recip(3)$	$\frac{11}{6}$

X

Catalogus >

BooleaanseUitdr1 **xor** *BooleaanseUitdr2*
 levert *Booleaanse uitdrukking*

BooleaanseLijst1 **xor** *BooleaanseLijst2*
 levert *Booleaanse lijst*

BooleaanseMatrix1 **xor** *BooleaanseMatrix2*
 levert *Booleaanse matrix*

Geeft waar als *BooleaanseUitdr1* waar is en *BooleaanseUitdr2* onwaar is, of andersom.

Geeft onwaar als beide argumenten waar zijn of als beide argumenten onwaar zijn. Geeft een vereenvoudigde Booleaanse uitdrukking als een van de argumenten niet omgezet kan worden naar waar of onwaar.

true xor true	false
5>3 xor 3>5	true

Opmerking: zie or, pag. 137.

Geheel getal1 xor Geheel getal2 ⇒ geheel getal

Vergelijkt twee reële gehele getallen bit-voor-bit met behulp van een **xor**-bewerking. Intern worden beide gehele getallen geconverteerd naar 64-bits binaire getallen met een teken (positief of negatief). Wanneer overeenkomstige bits vergeleken worden, is het resultaat 1 als een van de bits (maar niet beide) 1 is; het resultaat is 0 als beide bits 0 zijn of als beide bits 1 zijn. De geretourneerde waarde representeert de bitresultaten, en wordt weergegeven volgens de grondtal-modus.

U kunt de gehele getallen invoeren in elk grondtal. Voor een binaire of hexadecimale invoer moet u respectievelijk het prefix 0b of 0h gebruiken. Zonder prefix worden gehele getallen behandeld als decimaal (grondtal 10).

Als u een decimaal geheel getal invoert dat te groot is voor een 64-bits binaire vorm met een teken (positief of negatief), dan wordt er een symmetrische modulo-bewerking gebruikt om de waarde binnen het betreffende bereik te brengen. Zie voor meer informatie ►Base2, pag. 18.

Opmerking: zie or, pag. 137.

In de Hex-grondtalmodus:

Belangrijk: nul, niet de letter O.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

In de Bin-grondtalmodus:

0b100101 xor 0b100	0b100001
--------------------	----------

Opmerking: een binaire invoer kan maximaal 64 cijfers hebben (het prefix 0b niet meegeteld). Een hexadecimale invoer kan maximaal 16 cijfers hebben.

Z

zeros()

zeros(Uitdr, Var)⇒lijst

zeros(Uitdr, Var=Gok)⇒lijst

Geeft een lijst met mogelijke reële waarden van *Var* die *Uitdr*=0 maken. **zeros()** doet dit door **explist(solve(Uitdr=0,Var),Var)** te berekenen.

$$\text{zeros}\left(a \cdot x^2 + b \cdot x + c, x\right) = \left[\frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}, \frac{-\left(\sqrt{b^2 - 4 \cdot a \cdot c} + b\right)}{2 \cdot a} \right]$$

$$a \cdot x^2 + b \cdot x + c | x = \text{Ans}[2] \quad 0$$

Voor bepaalde doeleinden is het resultaat van **zeros()** handiger dan dat van **solve()**. Het resultaat van **zeros()** kan echter geen impliciete oplossingen, oplossingen waarvoor ongelijkheden nodig zijn of oplossingen waarin geen *Var* betrokken is weergeven.

Opmerking: zie ook **cSolve()**, **cZeros()** en **solve()**.

zeros{*Uitdr1*, *Uitdr2*}, {*VarOfGok1*, *VarOfGok2* [, ...]} $\Rightarrow$ *matrix*

Geeft mogelijke reële nulpunten voor simultane algebraïsche vergelijkingen, waarbij elke *VarOfGok* een onbekende specificeert waarnaar u wilt oplossen.

U kunt optioneel een begingok voor een variabele specificeren. Elke *VarOfGok* moet de volgende vorm hebben:

variabele

– of –

variabele = *reëel of niet-reëel getal*

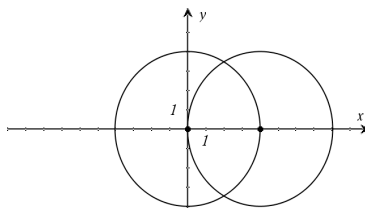
Bijvoorbeeld: *x* is geldig en *x=3* ook.

Als alle uitdrukkingen veeltermen zijn en als u GEEN begingokken specificeert, dan gebruikt **zeros()** de lexicale Gröbner/Buchberger-eliminatiemethode om te proberen alle reële nulpunten te bepalen.

Stel dat u een cirkel heeft met straal *r* en het middelpunt in de oorsprong, en een andere cirkel met straal *r* gecentreerd waar de eerste cirkel de positieve *x*-as snijdt. Gebruik **zeros()** om de snijpunten te vinden.

Zoals geïllustreerd door de *r* in het voorbeeld rechts, kunnen simultane polynomiale uitdrukkingen extra variabelen hebben die geen waarden hebben, maar die gegeven numerieke waarden representeren die later gesubstitueerd kunnen worden.

$$\begin{array}{l} \text{exact}\left(\text{zeros}\left(a \cdot (e^x + x) \cdot (\text{sign}(x) - 1), x\right)\right) \quad \{\emptyset\} \\ \text{exact}\left(\text{solve}\left(a \cdot (e^x + x) \cdot (\text{sign}(x) - 1) = 0, x\right)\right) \\ e^x + x = 0 \text{ or } x > 0 \text{ or } a = 0 \end{array}$$



$$\text{zeros}\left(\left\{x^2 + y^2 - r^2, (x - r)^2 + y^2 - r^2\right\}, \{x, y\}\right)$$

$$\begin{array}{cc} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \\ \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \end{array}$$

Elke rij van de resulterende matrix representeert een alternatief nulpunt, met de componenten op dezelfde manier geordend als in de lijst *VarOfGok*. Om een rij te extraheren, indexeert u de matrix met *[rij]*.

U kunt ook (of in plaats daarvan) onbekenden opnemen die niet in de uitdrukkingen verschijnen. U kunt bijvoorbeeld *z* opnemen als een onbekende om het eerdere voorbeeld uit te breiden naar twee parallelle snijdende cilinders met straal *r*. De cilindernulpunten laten zien hoe families van nulpunten willekeurige constanten zouden kunnen bevatten in de vorm *ck*, waarbij *k* een geheel getal-suffix van 1 tot en met 255 is.

Bij stelsels veeltermen kan de berekeningstijd of de belasting van het geheugen sterk afhangen van de volgorde waarin u de onbekende variabelen plaatst. Als uw eerste keuze het geheugen uitput of teveel van uw geduld vraagt, probeer de variabelen in de uitdrukkingen en/of de lijst *VarOfGok* dan te herschikken.

Als u geen gokken opneemt en als een uitdrukking in enige variabele niet-polynomiaal is, maar alle uitdrukkingen lineair zijn in de onbekenden, dan gebruikt **zeros()** Gaussische eliminatie om te proberen alle reële nulpunten te bepalen.

Als een stelsel noch polynomiaal in al zijn variabelen, noch lineair in zijn onbekenden is, dan bepaalt **zeros()** maximaal één nulpunt met behulp van een benaderende iteratieve methode. Om dit te doen moet het aantal onbekende variabelen gelijk zijn aan het aantal uitdrukkingen, en moeten alle andere variabelen in de uitdrukkingen vereenvoudigd worden tot getallen.

Elke onbekende begint bij de gegokte waarde, als die er is; anders begint deze bij 0,0.

Extraheer rij 2:

$$\text{Ans}[2] \quad \left[\begin{array}{cc} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \end{array} \right]$$

$$\text{zeros}\left(\left\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\right\}, \{x, y, z\}\right) \quad \left[\begin{array}{cc} \frac{r}{2} & \frac{-\sqrt{3} \cdot r}{2} & c1 \\ \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} & c1 \end{array} \right]$$

$$\text{zeros}\left(\left\{x+e^z \cdot y-1, x-y-\sin(z)\right\}, \{x, y\}\right) \quad \left[\begin{array}{cc} \frac{e^z \cdot \sin(z)+1}{e^z+1} & \frac{-(\sin(z)-1)}{e^z+1} \end{array} \right]$$

$$\text{zeros}\left(\left\{e^z \cdot y-1, y-\sin(z)\right\}, \{y, z\}\right) \quad \left[\begin{array}{cc} 0.041458 & 3.18306 \\ 0.001871 & 6.28131 \\ 4.76\text{E-}11 & 1796.99 \\ 2.\text{E-}13 & 254.469 \end{array} \right]$$

zeros()Catalogus > 

Gebruik gokken om één voor één aanvullende nulpunten te zoeken. Voor convergentie moet een gok mogelijk vrij dicht bij een nulpunt liggen.

$$\text{zeros}\left(\left\{e^z \cdot y - 1, y - \sin(z)\right\}, \left\{y, z = 2 \cdot \pi\right\}\right)$$

$$[0.001871 \quad 6.28131]$$
zIntervalCatalogus > 
zInterval $\sigma, \text{Lijst}, \text{Freq}, \text{CNiveau}$]

(Invoer van een gegevenslijst)

zInterval $\sigma, \bar{x}, n, \text{CNiveau}$]

(Invoer van samenvattingsstatistieken)

Berekent een z -betrouwbaarheidsinterval. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.CLower, stat.CUpper	Betrouwbaarheidsinterval voor een onbekend populatiegemiddelde
stat. $\bar{x}$	Steekproefgemiddelde van de gegevensverzameling uit de willekeurige normale verdeling
stat.ME	Foutmarge
stat.sx	Standaarddeviatie steekproef
stat.n	Lengte van de gegevensverzameling met het steekproefgemiddelde
stat. σ	Bekende populatiestandaarddeviatie voor gegevensverzameling <i>Lijst</i>

zInterval_1PropCatalogus > 
zInterval_1Prop $x, n, \text{CNiveau}$]

Berekent een z -betrouwbaarheidsinterval voor één proportie. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

x is een niet-negatief geheel getal.

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.CLower, stat.CUpper	Betrouwbaarheidsinterval met de betrouwbaarheidskans gebaseerd op de verdeling
stat. $\hat{p}$	De berekende proportie van successen
stat.ME	Foutmarge
stat.n	Aantal steekproeven in de gegevensverzameling

zInterval_2Prop

Catalogus > 

zInterval_2Prop $x1, n1, x2, n2, CNiveau$

Berekent een z-betrouwbaarheidsinterval voor twee proporties. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

$x1$ en $x2$ zijn niet-negatieve gehele getallen.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.CLower, stat.CUpper	Betrouwbaarheidsinterval met de betrouwbaarheidskans gebaseerd op de verdeling
stat. $\hat{p}$ Diff	Het berekende verschil tussen de proporties
stat.ME	Foutmarge
stat. $\hat{p}1$	Eerste schatting van de steekproefproportie
stat. $\hat{p}2$	Tweede schatting van de steekproefproportie
stat.n1	Steekproefomvang in gegevensverzameling één
stat.n2	Steekproefomvang in gegevensverzameling twee

zInterval_2Samp

Catalogus > 

zInterval_2Samp $\sigma_1, \sigma_2, Lijst1, Lijst2[, Freq1[, Freq2, [CNiveau]]]$

(Invoer van een gegevenslijst)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2, [CNiveau]$

(Invoer van samenvattingsstatistieken)

Berekent een z -betrouwbaarheidsinterval voor twee steekproeven. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.CLower, stat.CUpper	Betrouwbaarheidsinterval met de betrouwbaarheidskans gebaseerd op de verdeling
stat. $\bar{x}1$ - $\bar{x}2$	Steekproefgemiddelden van de gegevensverzameling uit de willekeurige normale verdeling
stat.ME	Foutmarge
stat. $\bar{x}1$, stat. $\bar{x}2$	Steekproefgemiddelden van de gegevensverzameling uit de willekeurige normale verdeling
stat. $\sigma x1$, stat. $\sigma x2$	Steekproefstandaarddeviaties voor <i>Lijst 1</i> en <i>Lijst 2</i>
stat.n1, stat.n2	Aantal steekproeven in de gegevensverzamelingen
stat.r1, stat.r2	Bekende populatiestandaarddeviatie voor gegevensverzameling <i>Lijst 1</i> en <i>Lijst 2</i>

zTest

zTest $\mu0, \sigma, Lijst, [Freq[, Hypoth]]$

(Invoer van een gegevenslijst)

zTest $\mu0, \sigma, \bar{x}, n, [Hypoth]$

(Invoer van samenvattingsstatistieken)

Voer een z -toets uit met frequentie *freqlijst*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Toets $H_0: \mu = \mu0$ tegen één van de volgende alternatieven:

Voor $H_1: \mu < \mu0$ stelt u *Hypoth*<0 in

Voor $H_1: \mu \neq \mu0$ (standaardinstelling) stelt u *Hypoth*≠0 in

Voor $H_1: \mu > \mu0$ stelt u *Hypoth*>0 in

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.z	$(\bar{x} - \mu_0) / (\sigma / \sqrt{n})$
stat.P Value	Kleinste kans waarbij de nulhypothese verworpen kan worden
stat. $\bar{x}$	Steekproefgemiddelde van de gegevensverzameling in <i>Lijst</i>
stat.sx	Steekproefstandaarddeviatie van de gegevensverzameling. Wordt alleen gegeven bij <i>Gegevens</i> -invoer.
stat.n	Omvang van de steekproef

zTest_1Prop

zTest_1Prop $p_0, x, n[, Hypoth]$

Berekent een z-toets voor één proportie. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

x is een niet-negatief geheel getal.

Toets $H_0: p = p_0$ tegen één van de volgende alternatieven:

Voor $H_1: p > p_0$ stelt u *Hypoth*>0 in

Voor $H_1: p \neq p_0$ (standaardinstelling) stelt u *Hypoth*=0 in

Voor $H_1: p < p_0$ stelt u *Hypoth*<0 in

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 267).

Uitvoervariabele	Beschrijving
stat.p0	Veronderstelde populatieproportie
stat.z	Standaard normale waarde berekend voor de proportie
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat. $\hat{p}$	Geschatte steekproefproportie
stat.n	Omvang van de steekproef

zTest_2Prop $x1, n1, x2, n2[, Hypoth]$

Berekent een z -toets met twee proporties. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

$x1$ en $x2$ zijn niet-negatieve gehele getallen.

Toets $H_0: p1 = p2$ tegen één van de volgende alternatieven:

Voor $H_a: p1 > p2$ stelt u *Hypoth*>0 in

Voor $H_a: p1 \neq p2$ (standaardinstelling) stelt u *Hypoth*=0 in

Voor $H_a: p < p0$ stelt u *Hypoth*<0 in

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.z	Standaard normale waarde berekend voor het verschil tussen de proporties
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat. $\hat{p}1$	Eerste schatting van de steekproefproportie
stat. $\hat{p}2$	Tweede schatting van de steekproefproportie
stat. $\hat{p}$	Gepoolde schatting van de steekproefproportie
stat.n1, stat.n2	Aantal steekproeven genomen in pogingen 1 en 2

zTest_2Samp**zTest_2Samp** $\sigma_1, \sigma_2, Lijst1, Lijst2[, Freq1[, Freq2[, Hypoth]]]$

(Invoer van een gegevenslijst)

zTest_2Samp $\sigma_1, \sigma_2, \bar{x}1, n1, \bar{x}2, n2[, Hypoth]$

(Invoer van samenvattingsstatistieken)

Berekent een z -toets voor twee steekproeven. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 188).

Toets $H_0: \mu1 = \mu2$ tegen een van de volgende alternatieven:

Voor $H_1: \mu_1 < \mu_2$ stelt u *Hypoth*<0 in

Voor $H_1: \mu_1 \neq \mu_2$ (standaardinstelling) stelt u *Hypoth*=0 in

Voor $H_1: \mu_1 > \mu_2$, *Hypoth*>0

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 267).

Uitvoervariabele	Beschrijving
stat.z	Standaard normale waarde berekend voor het verschil tussen de gemiddelden
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.x̄1, stat.x̄2	Steekproefgemiddelde van de gegevensverzamelingen in <i>Lijst1</i> en <i>Lijst2</i>
stat.sx1, stat.sx2	Steekproefstandaarddevaties van de gegevensverzamelingen in <i>Lijst1</i> en <i>Lijst2</i>
stat.n1, stat.n2	Grootte van de steekproeven

Symbolen

+ (optellen)		+ -toets
$Uitdr1 + Uitdr2 \Rightarrow uitdrukking$	56	56
Geeft de som van de twee argumenten.	$56+4$	60
	$60+4$	64
	$64+4$	68
	$68+4$	72
$Lijst1 + Lijst2 \Rightarrow lijst$	$\left\{22,\pi,\frac{\pi}{2}\right\} \rightarrow I1$	$\left\{22,\pi,\frac{\pi}{2}\right\}$
$Matrix1 + Matrix2 \Rightarrow matrix$	$\left\{10,5,\frac{\pi}{2}\right\} \rightarrow I2$	$\left\{10,5,\frac{\pi}{2}\right\}$
Geeft een lijst (of matrix) met de som van de overeenkomstige elementen in $Lijst1$ en $Lijst2$ (of $Matrix1$ en $Matrix2$).	$I1+I2$	$\{32,\pi+5,\pi\}$
	$Ans+\{\pi,-5,\pi\}$	$\{\pi+32,\pi,0\}$
De afmetingen van de argumenten moeten gelijk zijn.	$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$
$Uitdr + Lijst1 \Rightarrow lijst$	$15+\{10,15,20\}$	$\{25,30,35\}$
$Lijst1 + Uitdr \Rightarrow lijst$	$\{10,15,20\}+15$	$\{25,30,35\}$
Geeft een lijst met de som van $Uitdr$ en elk element in $Lijst1$.		
$Uitdr + Matrix1 \Rightarrow matrix$	$20+\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
$Matrix1 + Uitdr \Rightarrow matrix$		
Geeft een matrix met $Uitdr$ opgeteld bij elk element op de diagonaal van $Matrix1$. $Matrix1$ moet vierkant zijn.		
Opmerking: gebruik .+ (punt plus) om een uitdrukking bij elk element op te tellen.		

- (aftrekken)		- -toets
$Uitdr1 - Uitdr2 \Rightarrow uitdrukking$	$6-2$	4
Geeft $Uitdr1$ min $Uitdr2$.	$\pi - \frac{\pi}{6}$	$\frac{5 \cdot \pi}{6}$
$Lijst1 - Lijst2 \Rightarrow lijst$	$\left\{22,\pi,\frac{\pi}{2}\right\} - \left\{10,5,\frac{\pi}{2}\right\}$	$\{12,\pi-5,0\}$
$Matrix1 - Matrix2 \Rightarrow matrix$	$\begin{bmatrix} 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & 2 \end{bmatrix}$

–(aftrekken)



Trekt elk element in *Lijst2* (of *Matrix2*) af van het overeenkomstige element in *Lijst1* (of *Matrix1*), en geeft de uitkomsten.

De afmetingen van de argumenten moeten gelijk zijn.

Uitdr – *Lijst1* $\Rightarrow$ *lijst*

$$15 - \{10, 15, 20\} \quad \{5, 0, -5\}$$

Lijst1 – *Uitdr* $\Rightarrow$ *lijst*

$$\{10, 15, 20\} - 15 \quad \{-5, 0, 5\}$$

Trekt elk *Lijst1*-element af van *Uitdr* of trekt *Uitdr* af van elk *Lijst1*-element, en geeft een lijst met de uitkomsten.

Uitdr – *Matrix1* $\Rightarrow$ *matrix*

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

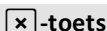
Matrix1 – *Uitdr* $\Rightarrow$ *matrix*

Uitdr – *Matrix1* geeft een matrix van *Uitdr* keer de eenheidsmatrix min *Matrix1*. *Matrix1* moet vierkant zijn.

Matrix1 – *Uitdr* geeft een matrix van *Uitdr* keer de identiteitsmatrix afgetrokken van *Matrix1*. *Matrix1* moet vierkant zijn.

Opmerking: gebruik .– (punt min) om een uitdrukking van elk element af te trekken.

• (vermenigvuldigen)



Uitdr1 • *Uitdr2* $\Rightarrow$ *uitdrukking*

$$2 \cdot 3 \cdot 45 \quad 6.9$$

Geeft het product van de twee argumenten.

$$x \cdot y \cdot x \quad x^2 \cdot y$$

Lijst1 • *Lijst2* $\Rightarrow$ *lijst*

$$\{1, 2, 3\} \cdot \{4, 5, 6\} \quad \{4, 10, 18\}$$

Geeft een lijst met de producten van de overeenkomstige elementen in *Lijst1* en *Lijst2*.

$$\left\{ \frac{2}{a}, \frac{3}{a} \right\} \cdot \left\{ a^2, \frac{b}{3} \right\} \quad \left\{ 2 \cdot a, \frac{b}{2} \right\}$$

De afmetingen van de lijsten moeten gelijk zijn.

Matrix1 • *Matrix2* $\Rightarrow$ *matrix*

Geeft het matrixproduct van *Matrix1* en *Matrix2*.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix} \quad \begin{bmatrix} a+2 \cdot b+3 \cdot c & d+2 \cdot e+3 \cdot f \\ 4 \cdot a+5 \cdot b+6 \cdot c & 4 \cdot d+5 \cdot e+6 \cdot f \end{bmatrix}$$

Het aantal kolommen in *Matrix1* moet gelijk zijn aan het aantal rijen in *Matrix2*.

• (vermenigvuldigen)

✖-toets

Uitdr • *Lijst1* ⇒ *lijst*

$$\pi \cdot \{4,5,6\}$$

$$\{4 \cdot \pi, 5 \cdot \pi, 6 \cdot \pi\}$$

Lijst1 • *Uitdr* ⇒ *lijst*

Geeft een lijst met de producten van *Uitdr* en elk element in *Lijst1*.

Uitdr • *Matrix1* ⇒ *matrix*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01$$

$$\begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

Matrix1 • *Uitdr* ⇒ *matrix*

$$\lambda \cdot \text{identity}(3)$$

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

Geeft een matrix met de producten van *Uitdr* en elk element in *Matrix1*.

Opmerking: gebruik $\cdot$ (punt vermenigvuldigen) om een uitdrukking met elk element te vermenigvuldigen.

/ (delen)

÷-toets

Uitdr1 / *Uitdr2* ⇒ *uitdrukking*

$$\frac{2}{3.45}$$

$$.57971$$

Geeft het quotiënt van *Uitdr1* gedeeld door *Uitdr2*.

$$\frac{x^3}{x}$$

$$x^2$$

Opmerking: zie ook **Breuk-template**, pag. 1.

Lijst1 / *Lijst2* ⇒ *lijst*

$$\frac{\{1.,2,3\}}{\{4,5,6\}}$$

$$\left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

Geeft een lijst met de quotiënten van *Lijst1* gedeeld door *Lijst2*.

De afmetingen van de lijsten moeten gelijk zijn.

Uitdr / *Lijst1* ⇒ *lijst*

$$\frac{a}{\{3,a,\sqrt{a}\}}$$

$$\left\{\frac{a}{3}, 1, \sqrt{a}\right\}$$

Lijst1 / *Uitdr* ⇒ *lijst*

$$\frac{\{a,b,c\}}{a \cdot b \cdot c}$$

$$\left\{\frac{1}{b \cdot c}, \frac{1}{a \cdot c}, \frac{1}{a \cdot b}\right\}$$

Geeft een lijst met de quotiënten van *Uitdr* gedeeld door *Lijst1* of *Lijst1* gedeeld door *Uitdr*.

Matrix1 / *Uitdr* ⇒ *matrix*

$$\frac{\begin{bmatrix} a & b & c \end{bmatrix}}{a \cdot b \cdot c}$$

$$\begin{bmatrix} \frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b} \end{bmatrix}$$

Geeft een matrix met de quotiënten van *Matrix1* / *Uitdr*.

Opmerking: gebruik $/$ (punt gedeeld door) om een uitdrukking door elk element te delen.

Uitdr1 ^ *Uitdr2* ⇒ uitdrukking

$$4^2 \quad 16$$

Lijst1 ^ *Lijst2* ⇒ lijst

$$\{a, 2, c\}^{\{1, b, 3\}} \quad \{a, 2^b, c^3\}$$

Geeft het eerste argument, verheven tot de macht van het twee argument.

Opmerking: zie ook **Exponent-template**, pag. 1.

Geeft bij een lijst de elementen in *Lijst1* verheven tot de macht van de overeenkomstige elementen in *Lijst2*.

In het reële domein gebruiken gebroken machten die te vereenvoudigen zijn tot exponenten met oneven noemers de reële tak, versus de principaaltak voor de complexe modus.

Uitdr ^ *Lijst1* ⇒ lijst

$$p^{\{a, 2, -3\}} \quad \left\{p^a, p^2, \frac{1}{p^3}\right\}$$

Geeft *Uitdr* verheven tot de macht van de elementen in *Lijst1*.

Lijst1 ^ *Uitdr* ⇒ lijst

$$\{1, 2, 3, 4\}^{-2} \quad \left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}\right\}$$

Geeft de elementen in *Lijst1* verheven tot de macht van *Uitdr*.

vierkanteMatrix1 ^ *geheel getal* ⇒ matrix

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2 \quad \begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$$

Geeft *vierkanteMatrix1* verheven tot de *gehele* macht.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

vierkanteMatrix1 moet een vierkante matrix zijn.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ -15 & 7 \\ 4 & 4 \end{bmatrix}$$

Als *geheel getal* = -1 berekent dit commando de inverse matrix.

Als *geheel getal* < -1 berekent dit commando de inverse matrix tot de passende positieve macht.

x^2 (kwadraat) **x^2 -toets***Uitdr*¹² $\Rightarrow$ uitdrukking

4^2	16
-------	----

Geeft het kwadraat van het argument.

$\{2,4,6\}^2$	$\{4,16,36\}$
---------------	---------------

*Lijst*¹² $\Rightarrow$ lijst

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2$	$\begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$
---	--

Geeft een lijst met de kwadraten van de elementen in *Lijst*¹.*vierkanteMatrix*¹² $\Rightarrow$ matrix

$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix} \wedge 2$	$\begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$
--	--

Geeft het matrixkwadraat van *vierkanteMatrix*¹. Dit is niet hetzelfde als het berekenen van het kwadraat van elk element. Gebruik $\wedge 2$ om het kwadraat van elk element te berekenen. **$+$ (punt optellen)** **$+$ -toetsen***Matrix1* $+$ *Matrix2* $\Rightarrow$ matrix

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$
---	--

Uitdr $+$ *Matrix1* $\Rightarrow$ matrix

$x + \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$
--	--

Matrix1 $+$ *Matrix2* geeft een matrix met de som van elk paar overeenkomstige elementen in *Matrix1* en *Matrix2*.*Uitdr* $+$ *Matrix1* geeft een matrix met de sommen van *Uitdr* en elk element in *Matrix1*. **$-$ (punt aftrekken)** **$-$ -toetsen***Matrix1* $-$ *Matrix2* $\Rightarrow$ matrix

$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$
---	--

Uitdr $-$ *Matrix1* $\Rightarrow$ matrix

$x - \begin{bmatrix} c & 4 \\ d & 5 \end{bmatrix}$	$\begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$
--	--

Matrix1 $-$ *Matrix2* geeft een matrix met het verschil tussen elk paar overeenkomstige elementen in *Matrix1* en *Matrix2*.*Uitdr* $-$ *Matrix1* geeft een matrix met de verschillen van *Uitdr* en elk element in *Matrix1*.

. (punt vermenigvuldigen)

. x -toetsen

Matrix1 . *Matrix2* ⇒ *matrix*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$$

Uitdr . *Matrix1* ⇒ *matrix*

$$x \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$$

Matrix1 . *Matrix2* geeft een matrix met het product van elk paar overeenkomstige elementen in *Matrix1* en *Matrix2*.

Uitdr . *Matrix1* geeft een matrix met de producten van *Uitdr* en elk element in *Matrix1*.

. / (punt delen)

. ÷ -toetsen

Matrix1 . / *Matrix2* ⇒ *matrix*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \left(\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \right) = \begin{bmatrix} \frac{a}{c} & \frac{1}{2} \\ \frac{b}{5} & \frac{3}{d} \end{bmatrix}$$

Uitdr . / *Matrix1* ⇒ *matrix*

$$x \cdot \left(\begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} \right) = \begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$$

Matrix1 . / *Matrix2* geeft een matrix met het quotiënt van elk paar overeenkomstige elementen in *Matrix1* en *Matrix2*.

Uitdr . / *Matrix1* geeft een matrix met de quotiënten van *Uitdr* en elk element in *Matrix1*.

. ^ (punt machtsverheffen)

. ^ -toetsen

Matrix1 . ^ *Matrix2* ⇒ *matrix*

$$\begin{bmatrix} a & 2 \\ b & 3 \end{bmatrix} \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} a^c & 16 \\ b^5 & 3^d \end{bmatrix}$$

Uitdr . ^ *Matrix1* ⇒ *matrix*

$$x \cdot \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix} = \begin{bmatrix} x^c & x^4 \\ x^5 & x^d \end{bmatrix}$$

Matrix1 . ^ *Matrix2* geeft een matrix waarbij elk element in *Matrix2* de exponent voor het overeenkomstige element in *Matrix1* is.

Uitdr . ^ *Matrix1* geeft een matrix waarbij elk element in *Matrix1* de exponent voor *Uitdr* is.

- (negatief)

 -toets

-Uitdr1 $\Rightarrow$ *uitdrukking*

-Lijst1 $\Rightarrow$ *lijst*

-Matrix1 $\Rightarrow$ *matrix*

Geeft het tegengestelde van het argument.

Geeft bij een lijst of matrix het tegengestelde van elk element.

Als het argument een binair of hexadecimaal geheel getal is, geeft de negatie het twee-complement.

-2.43	-2.43
$-\{-1,0,4,1,2\}$	$\{1,-0,4,-1,2\}$
$-a \cdot -b$	$a \cdot b$

In de Bin-grondtalmodus:

Belangrijk: nul, niet de letter O.

```
-0b100101
0b11111111111111111111111111111111▶
```

Om het hele resultaat te zien drukt u op $\blacktriangle$ en gebruikt u vervolgens $\blacktriangleleft$ en $\blacktriangleright$ om de cursor te verplaatsen.

% (percentage)

 -toetsen

Uitdr1 % $\Rightarrow$ *uitdrukking*

Lijst1 % $\Rightarrow$ *lijst*

Matrix1 % $\Rightarrow$ *matrix*

argument

Geeft **100**

Geeft bij een lijst of matrix een lijst of matrix met elk element gedeeld door 100.

Opmerking: Om een decimale benaderende uitkomst te forceren,

Rekenmachine: Druk op  .

Windows®: Druk op **Ctrl+Enter**.

Macintosh®: Druk op **⌘+Enter**.

iPad®: Houd **Enter** ingedrukt en selecteer .

13%	0.13
-----	------

$\{\{1,10,100\}\}\%$	$\{0.01,0.1,1.\}$
----------------------	-------------------

= (is gelijk)

 -toets

Uitdr1 = Uitdr2 $\Rightarrow$ *Booleaanse uitdrukking*

Lijst1 = Lijst2 $\Rightarrow$ *Booleaanse lijst*

Matrix1 = Matrix2 $\Rightarrow$ *Booleaanse matrix*

Voorbeeldfunctie waarin wiskundige test-symbolen worden gebruikt: =, $\neq$, <, >, $\geq$

= (is gelijk)

-toets

Geeft waar als vastgesteld is dat *Uitdr1* gelijk is aan *Uitdr2*.

Geeft onwaar als vastgesteld is dat *Uitdr1* niet gelijk is aan *Uitdr2*.

Elke andere invoer geeft een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Define $g(x)$ =Func

If $x \leq -5$ Then

Return 5

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

Return $-x$

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

Return x

ElseIf $x = 10$ Then

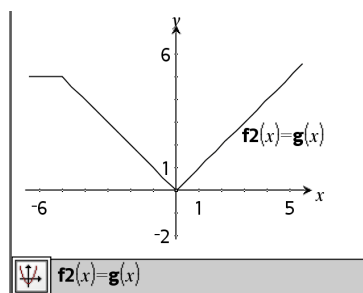
Return 3

EndIf

EndFunc

Done

Resultaat van het tekenen van de grafiek $g(x)$



≠ (is niet gelijk)

-toetsen

$Uitdr1 \neq Uitdr2 \Rightarrow$ Booleaanse uitdrukking

$Lijst1 \neq Lijst2 \Rightarrow$ Booleaanse lijst

$Matrix1 \neq Matrix2 \Rightarrow$ Booleaanse matrix

Geeft waar als vastgesteld is dat *Uitdr1* niet gelijk is aan *Uitdr2*.

Geeft onwaar als vastgesteld is dat *Uitdr1* gelijk is aan *Uitdr2*.

Elke andere invoer geeft een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

Zie het voorbeeld voor “=” (is gelijk).

$\neq$ (is niet gelijk)

  -toetsen

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door $\neq$ in te typen

$<$ (kleiner dan)

  -toetsen

$Uitdr1 < Uitdr2 \Rightarrow$ Booleaanse uitdrukking

Zie het voorbeeld voor “=” (is gelijk).

$Lijst1 < Lijst2 \Rightarrow$ Booleaanse lijst

$Matrix1 < Matrix2 \Rightarrow$ Booleaanse matrix

Geeft waar als vastgesteld is dat $Uitdr1$ kleiner is dan $Uitdr2$.

Geeft onwaar als vastgesteld is dat $Uitdr1$ groter dan of gelijk is aan $Uitdr2$.

Elke andere invoer geeft een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

$\leq$ (kleiner dan of gelijk aan)

  -toetsen

$Uitdr1 \leq Uitdr2 \Rightarrow$ Booleaanse uitdrukking

Zie het voorbeeld voor “=” (is gelijk).

$Lijst1 \leq Lijst2 \Rightarrow$ Booleaanse lijst

$Matrix1 \leq Matrix2 \Rightarrow$ Booleaanse matrix

Geeft waar als vastgesteld is dat $Uitdr1$ kleiner dan of gelijk is aan $Uitdr2$.

Geeft onwaar als vastgesteld is dat $Uitdr1$ groter is dan $Uitdr2$.

Elke andere invoer geeft een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door $\leq$ in te typen

> (groter dan)

  -toetsen

$Uitdr1 > Uitdr2 \Rightarrow$ Booleaanse uitdrukking

Zie het voorbeeld voor “=” (is gelijk).

$Lijst1 > Lijst2 \Rightarrow$ Booleaanse lijst

$Matrix1 > Matrix2 \Rightarrow$ Booleaanse matrix

Geeft waar als vastgesteld is dat *Uitdr1* groter is dan *Uitdr2*.

Geeft onwaar als vastgesteld is dat *Uitdr1* kleiner dan of gelijk is aan *Uitdr2*.

Elke andere invoer geeft een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

$\geq$ (groter dan of gelijk aan)

  -toetsen

$Uitdr1 \geq Uitdr2 \Rightarrow$ Booleaanse uitdrukking

Zie het voorbeeld voor “=” (is gelijk).

$Lijst1 \geq Lijst2 \Rightarrow$ Booleaanse lijst

$Matrix1 \geq Matrix2 \Rightarrow$ Booleaanse matrix

Geeft waar als vastgesteld is dat *Uitdr1* groter dan of gelijk is aan *Uitdr2*.

Geeft onwaar als vastgesteld is dat *Uitdr1* kleiner is dan *Uitdr2*.

Elke andere invoer geeft een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door $\geq$ in te typen

⇒ (logische implicatie)**ctrl** **=** **toetsen***BooleaanseUitdr1 ⇒ BooleaanseUitdr2*
levert *Booleaanse uitdrukking* $5 > 3 \text{ or } 3 > 5$ true*BooleaanseLijst1 ⇒ BooleaanseLijst2*
levert *Booleaanse lijst* $5 > 3 \Rightarrow 3 > 5$ false $3 \text{ or } 4$ 7*BooleaanseMatrix1 ⇒*
BooleaanseMatrix2 levert *Booleaanse*
matrix $3 \Rightarrow 4$ -4 $\{1,2,3\} \text{ or } \{3,2,1\}$ $\{3,2,3\}$ $\{1,2,3\} \Rightarrow \{3,2,1\}$ $\{-1,-1,-3\}$ *Geheel getal1 ⇒ Geheel getal2* levert
Geheel getal

Werkt de uitdrukking **not** <argument1> **or** <argument2> uit en geeft waar, onwaar of een vereenvoudigde vorm van de vergelijking.

Bij lijsten en matrices vergelijkt dit commando element voor element.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoegen door => in te typen

⇔ (logische dubbele implicatie, XNOR)**ctrl** **=** **toetsen***BooleaanseUitdr1 ⇔ BooleaanseUitdr2*
levert *Booleaanse uitdrukking* $5 > 3 \text{ xor } 3 > 5$ true*BooleaanseLijst1 ⇔ BooleaanseLijst2*
levert *Booleaanse lijst* $5 > 3 \Leftrightarrow 3 > 5$ false $3 \text{ xor } 4$ 7*BooleaanseMatrix1 ⇔*
BooleaanseMatrix2 levert *Booleaanse*
matrix $3 \Leftrightarrow 4$ -8 $\{1,2,3\} \text{ xor } \{3,2,1\}$ $\{2,0,2\}$ $\{1,2,3\} \Leftrightarrow \{3,2,1\}$ $\{-3,-1,-3\}$ *Geheel getal1 ⇔ Geheel getal2* levert
Geheel getal

Geeft de ontkenning (negatie) van een **XOR** Booleaanse bewerking op de twee argumenten. Geeft waar, onwaar of een vereenvoudigde vorm van de vergelijking.

⇔ (logische dubbele implicatie, XNOR)

ctrl [= toetsen

Bij lijsten en matrices vergelijkt dit commando element voor element.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoegen door $\Leftrightarrow$ in te typen

! (faculteit)

?! ▶ -toets

Uitdr1! ⇒ *uitdrukking*

5! 120

Lijst1! ⇒ *lijst*

$\{\{5,4,3\}\}!$ $\{120,24,6\}$

Matrix1! ⇒ *matrix*

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

Geeft de faculteit van het argument.

Geeft bij een lijst of een matrix een lijst of een matrix met de faculteiten van de elementen.

& (toevoegen)

ctrl [-toetsen

String1 & *String2* ⇒ *string*

"Hello " & "Nick" "Hello Nick"

Geeft een tekststring die bestaat uit *String2* toegevoegd aan *String1*.

d() (afgeleide)

Catalogus > [d] [3]

d(*Uitdr1*, *Var*[,*Orde*]) ⇒ *uitdrukking*

$\frac{d}{dx}(f(x) \cdot g(x))$ $\frac{d}{dx}(f(x)) \cdot g(x) + \frac{d}{dx}(g(x)) \cdot f(x)$

d(*Lijst1*, *Var*[,*Orde*]) ⇒ *lijst*

$\frac{d}{dy}\left(\frac{d}{dx}(x^2 \cdot y^3)\right)$ $6 \cdot y^2 \cdot x$

d(*Matrix1*, *Var*[,*Order*]) ⇒ *matrix*

$\frac{d}{dx}\left(\left\{x^2, x^3, x^4\right\}\right)$ $\left\{2 \cdot x, 3 \cdot x^2, 4 \cdot x^3\right\}$

Geeft de eerste afgeleide van het eerste argument ten opzichte van variabele *Var*.

Orde moet, indien opgenomen, een geheel getal zijn. Als de orde kleiner dan nul is, dan is het resultaat een primitieve.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **derivative** (...) in te typen.

d() volgt niet het normale uitwerkingsmechanisme van het volledig vereenvoudigen van de argumenten en het vervolgens toepassen van de functiedefinitie op deze volledig vereenvoudigde argumenten. In plaats daarvan voert **d()** de volgende stappen uit:

1. Het tweede argument wordt slechts in zoverre vereenvoudigd dat het niet tot een non-variabele leidt.
2. Het eerste argument wordt slechts in zoverre vereenvoudigd dat het een opgeslagen waarde voor de variabele die bepaald is door stap 1, oproept.
3. De symbolische afgeleide van het resultaat van stap 2 wordt bepaald ten opzichte van de variabele uit stap 1.

Als de variabele uit stap 1 een opgeslagen waarde of een waarde die gespecificeerd wordt door de beperkende operator ("|") heeft, dan wordt die waarde gesubstitueerd in het resultaat uit stap 3.

Opmerking: zie ook **Eerste afgeleide**, pag. 5; **Tweede afgeleide**, pag. 6; **ofn-de afgeleide**, pag. 6.

∫() (integraal)

$\int(\text{Uitdr1}, \text{Var}[, \text{Onder}, \text{Boven}]) \Rightarrow$
uitdrukking

$\int(\text{Uitdr1}, \text{Var}[, \text{Constante}]) \Rightarrow$ uitdrukking

Geeft de integraal van *Uitdr1* ten opzichte van de variabele *Var* van *Onder* tot *Boven*.

Opmerking: zie ook **Bepaalde of Onbepaalde integraal-template**, pag. 6.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **integral (...)** in te typen.

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

Geeft een primitieve als *Onder* en *Boven* weggelaten worden. Een symbolische integratieconstante wordt weggelaten tenzij u het *Constante*-argument invoert.

$\int x^2 dx$	$\frac{x^3}{3}$
$\int (a \cdot x^2, x, c)$	$\frac{a \cdot x^3}{3} + c$

Geldige primitieven een numerieke constante verschillen. Zo'n constante kan 'vermomd' zijn—vooral als een primitieve logaritmes of inverse goniometrische functies bevat. Bovendien worden er soms stuksgewijs gedefinieerde constante-uitdrukkingen toegevoegd waardoor een primitieve geldig wordt over een groter interval dan de gebruikelijke formule.

∫() geeft zichzelf voor stukken van *Uitdr1* die het niet kan bepalen als een expliciete eindige combinatie van zijn ingebouwde functies en operatoren.

$\int b \cdot e^{-x^2} + \frac{a}{x^2 + a^2} dx$	$b \cdot \int e^{-x^2} dx + \tan^{-1}\left(\frac{x}{a}\right)$
--	--

Wanneer u *Onder* en *Boven* invoert, wordt er een poging gedaan om eventuele discontinuïteiten of discontinue afgeleiden in het interval *Onder* < *Var* < *Boven* te lokaliseren en om het interval op die plaatsen onder te verdelen.

Bij de automatische instelling van de **Automatische of Benaderende** modus wordt indien van toepassing numerieke integratie gebruikt, wanneer er geen primitieve of limiet kan worden bepaald.

Bij de Benaderende instelling wordt eerst numerieke integratie geprobeerd, indien van toepassing. Primitieven worden alleen gezocht waar een dergelijke numerieke integratie niet van toepassing is of mislukt.

Opmerking: Om een decimale benaderende uitkomst te forceren,

Rekenmachine: Druk op  .

Windows®: Druk op **Ctrl+Enter**.

Macintosh®: Druk op **⌘+Enter**.

iPad®: Houd **Enter** ingedrukt en selecteer



$\int_{-1}^1 e^{-x^2} dx$	1.49365
---------------------------	---------

∫() (integraal)

Catalogus >

∫() kan genest worden om meervoudige integralen te bepalen. Integratiegrenzen kunnen afhangen van integratievariabelen erbuiten.

Opmerking: zie ook `nInt()`, pag. 129.

$$\int_0^a \int_0^x \ln(x+y) \, dy \, dx$$

$$\frac{a^2 \cdot \ln(a)}{2} + \frac{a^2 \cdot (4 \cdot \ln(2) - 3)}{4}$$

√() (wortel)

`ctrl` `x²` -toetsen

√(*Uitdr1*) ⇒ uitdrukking

$$\sqrt{4} \quad 2$$

√(*Lijst1*) ⇒ lijst

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

Geeft de wortel van het argument.

Geeft bij een lijst de wortel van alle elementen in *Lijst1*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door `sqrt(...)` in te typen.

Opmerking: zie ook **Wortel-template**, pag. 1.

Π() (prodSeq)

Catalogus >

Π(*Uitdr1*, *Var*, *Laag*, *Hoog*) ⇒ uitdrukking

Werkt *Uitdr1* uit voor elke waarde van *Var* van *Laag* tot *Hoog*, en geeft het product van de resultaten.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door `prodSeq(...)` in te typen.

Werkt *Uitdr1* uit voor elke waarde van *Var* van *Laag* tot *Hoog*, en geeft het product van de resultaten.

Opmerking: zie ook **Product-template (Π)**, pag. 5.

Π(*Uitdr1*, *Var*, *Laag*, *Laag-1*) ⇒ 1

Π(*Uitdr1*, *Var*, *Laag*, *Hoog*) ⇒ 1/Π(*Uitdr1*, *Var*, *Hoog+1*, *Laag-1*) als *Hoog* < *Laag-1*

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

$$\prod_{k=1}^n (k^2) \quad (n!)^2$$

$$\prod_{n=1}^5 \left(\left\{ \frac{1}{n}, n, 2 \right\} \right) \quad \left\{ \frac{1}{120}, 120, 32 \right\}$$

$$\prod_{k=4}^3 (k) \quad 1$$

Π() (prodSeq)

Catalogus >

De gebruikte productformules zijn afkomstig uit de volgende bron:

Ronald L. Graham, Donald E. Knuth en Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) = 6$$

$$\prod_{k=4}^1 \left(\frac{1}{k} \right) \cdot \prod_{k=2}^4 \left(\frac{1}{k} \right) = \frac{1}{4}$$

Σ() (sumSeq)

Catalogus >

Σ(Uitdr1, Var, Laag, Hoog) ⇒ uitdrukking

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **sumSeq (...)** in te typen.

Werkt *Uitdr1* uit voor elke waarde van *Var* van *Laag* naar *Hoog*, en geeft de som van de resultaten.

Opmerking: Zie ook **Som-template**, pag. 5.

Σ(Uitdr1, Var, Laag, Laag-1) ⇒ 0

Σ(Uitdr1, Var, Laag, Hoog) ⇒ -Σ(Uitdr1, Var, Hoog+1, Laag-1) als Hoog < Laag-1

$$\sum_{n=1}^5 \left(\frac{1}{n} \right) = \frac{137}{60}$$

$$\sum_{k=1}^n (k^2) = \frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$$

$$\sum_{n=1}^{\infty} \left(\frac{1}{n^2} \right) = \frac{\pi^2}{6}$$

$$\sum_{k=4}^3 (k) = 0$$

De gebruikte somformules zijn afkomstig uit de volgende bron:

Ronald L. Graham, Donald E. Knuth en Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Reading, Massachusetts: Addison-Wesley, 1994.

$$\sum_{k=4}^1 (k) = -5$$

$$\sum_{k=4}^1 (k) + \sum_{k=2}^4 (k) = 4$$

ΣInt()

Catalogus >

ΣInt(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [afgerondeWaarde]) ⇒ waarde

ΣInt(NPmt1, NPmt2, amortTable) ⇒ waarde

$$\Sigma \text{Int}(1, 3, 12, 4.75, 20000, 12, 12) = -213.48$$

Aflossingsfunctie die de som van de rente gedurende een gespecificeerd aantal betalingen berekent.

$NPmt1$ en $NPmt2$ definiëren de begin- en eindgrenzen van het betalingsbereik.

N , I , PV , Pmt , FV , PpY , CpY en $PmtAt$ worden beschreven in de tabel met TVM-argumenten, pag. 208.

- Als u Pmt weglaat, dan wordt de standaardwaarde $Pmt=\text{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ gebruikt.
- Als u FV weglaat, dan wordt de standaardwaarde $FV=0$ gebruikt.
- De standaardwaarden voor PpY , CpY en $PmtAt$ zijn hetzelfde als voor de TVM-functies.

afgerondeWaarde specificeert het aantal decimalen voor afronding.
Standaardwaarde=2.

$\Sigma\text{Int}(NPmt1, NPmt2, amortTable)$ berekent de som van de rente op basis van de aflossingstabel *amortTable*. Het argument *amortTable* moet een matrix zijn met de vorm die beschreven wordt onder **amortTbl()**, pag. 8.

Opmerking: zie ook $\Sigma\text{Prn}()$, hieronder, en **Bal()**, pag. 17.

$tbl:=\text{amortTbl}(12, 12, 4.75, 20000., 12, 12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.6
2	-71.17	-1638.75	16728.8
3	-64.82	-1645.1	15083.7
4	-58.44	-1651.48	13432.2
5	-52.05	-1657.87	11774.4
6	-45.62	-1664.3	10110.1
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma\text{Int}(1, 3, tbl)$ -213.48

$\Sigma\text{Prn}(NPmt1, NPmt2, N, I, PV, [Pmt], [FV], [PpY], [CpY], [PmtAt], [afgerondeWaarde]) \Rightarrow \text{waarde}$

$\Sigma\text{Prn}(1, 3, 12, 4.75, 20000., 12, 12)$ -4916.28

$\Sigma\text{Prn}(NPmt1, NPmt2, amortTable) \Rightarrow \text{waarde}$

Aflossingsfunctie die de som van de hoofdsom gedurende een gespecificeerd aantal betalingen berekent.

$NPmt1$ en $NPmt2$ definiëren de begin- en eindgrenzen van het betalingsbereik.

$N, I, PV, Pmt, FV, PpY, CpY$ en $PmtAt$ worden beschreven in de tabel met TVM-argumenten, pag. 208.

- Als u Pmt weglaat, dan wordt de standaardwaarde $Pmt = \mathbf{tvmPmt}(N, I, PV, FV, PpY, CpY, PmtAt)$ gebruikt.
- Als u FV weglaat, dan wordt de standaardwaarde $FV = 0$ gebruikt.
- De standaardwaarden voor PpY, CpY en $PmtAt$ zijn hetzelfde als voor de TVM-functies.

afgeronde Waarde specificeert het aantal decimalen voor afronding.
Standaardwaarde=2.

$\Sigma Prn(NPmt1, NPmt2, amortTable)$ berekent de som van de betaalde hoofdsom op basis van de aflossingstabel *amortTable*. Het argument *amortTable* moet een matrix zijn met de vorm die beschreven wordt onder $\mathbf{amortTbl}()$, pag. 8.

Opmerking: zie ook $\Sigma Int()$, hierboven, en $Bal()$, pag. 17.

$tbl := \mathbf{amortTbl}(12, 12, 4.75, 20000., 12, 12)$

0	0.	0.	20000.
1	-77.49	-1632.43	18367.57
2	-71.17	-1638.75	16728.82
3	-64.82	-1645.1	15083.72
4	-58.44	-1651.48	13432.24
5	-52.05	-1657.87	11774.37
6	-45.62	-1664.3	10110.07
7	-39.17	-1670.75	8439.32
8	-32.7	-1677.22	6762.1
9	-26.2	-1683.72	5078.38
10	-19.68	-1690.24	3388.14
11	-13.13	-1696.79	1691.35
12	-6.55	-1703.37	-12.02

$\Sigma Prn(1, 3, tbl)$ -4916.28

(indirectie)

  -toetsen

varNaamString

Verwijst naar de variabele met de naam *varNaamString*. Hiermee kunt u strings gebruiken om variabelenamen binnen een functie te creëren.

$\#("x" \& "y" \& "z")$ xyz

Creëert of verwijst naar de variabele xyz.

$10 \rightarrow r$	10
$"r" \rightarrow s1$	"r"
$\#s1$	10

Geeft de waarde van de variabele (r) waarvan de naam is opgeslagen in variabele s1.

E (wetenschappelijke notatie)

EE-toets

*mantisse***E***exponent*

23000.	23000.
2300000000.+4.1E15	4.1E15
$3 \cdot 10^4$	30000

Voert een getal in wetenschappelijke notatie in. Het getal wordt geïnterpreteerd als *mantisse* $\times 10^{\text{exponent}}$.

Tip: als u een macht van 10 wilt invoeren zonder een resultaat met decimalen te veroorzaken, gebruik dan $10^{\text{geheel getal}}$.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **@E** in te typen. typ bijvoorbeeld **2 . 3@E4** om $2.3E4$ in te voeren.

g (decimale graden)

π -toets

*Uitdr1***g** $\Rightarrow$ *uitdrukking*

In de modus Graden, Decimale graden of Radialen:

*Lijst1***g** $\Rightarrow$ *lijst*

$$\cos(50^g) \quad \frac{\sqrt{2}}{2}$$

*Matrix1***g** $\Rightarrow$ *matrix*

$$\cos(\{0, 100^g, 200^g\}) \quad \{1, 0, -1\}$$

Deze functie geeft u een manier om een hoek in decimale graden te specificeren terwijl u in de modus Graden of Radialen bent.

In de hoekmodus Radialen: vermenigvuldigt *Uitdr1* met $\pi/200$.

In de hoekmodus Graden: vermenigvuldigt *Uitdr1* met $g/100$.

In de modus Decimale graden: geeft *Uitdr1* ongewijzigd.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door **@g** in te voeren.

r (radialen)

π -toets

*Uitdr1***r** $\Rightarrow$ *uitdrukking*

In de hoekmodus Graden, Decimale graden of Radialen:

*Lijst1***r** $\Rightarrow$ *lijst*

*Matrix1***r** $\Rightarrow$ *matrix*

ʀ (radialen)

π -toets

Deze functie geeft u een manier om een hoek in radialen te specificeren terwijl u in deodus Graden of Decimale graden bent.

In de hoekmodus Graden: vermenigvuldigt het argument met $180/\pi$.

In de hoekmodus Radialen: geeft het argument ongewijzigd.

In deodus Decimale graden: vermenigvuldigt het argument met $200/\pi$.

Tip: gebruik ʀ als u radialen wilt forceren in een functiedefinitie, ongeacht deodus die de voorkeur heeft wanneer de functie wordt gebruikt.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @ʀ in te voeren.

$$\cos\left(\frac{\pi}{4ʀ}\right) \quad \frac{\sqrt{2}}{2}$$

$$\cos\left(\left\{0ʀ, \frac{\pi}{12}, \piʀ, (\pi)ʀ\right\}\right) \quad \left\{1, \frac{(\sqrt{3+1}) \cdot \sqrt{2}}{4}, -1\right\}$$

° (graden)

π -toets

Uitdrʀ° ⇒ uitdrukking

Lijstʀ° ⇒ lijst

Matrixʀ° ⇒ matrix

Deze functie geeft u een manier om een hoek in graden te specificeren terwijl u in deodus Decimale graden of Radialen bent.

In de hoekmodus Radialen: vermenigvuldigt het argument met $\pi/180$.

In de hoekmodus Graden: geeft het argument ongewijzigd.

In de hoekmodus Decimale graden: vermenigvuldigt het argument met $10/9$.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @d in te voeren.

In de hoekmodus Graden, Decimale graden of Radialen:

$$\cos(45^\circ) \quad \frac{\sqrt{2}}{2}$$

In de hoekmodus Radialen:

Opmerking: Om een decimale benaderende uitkomst te forceren,

Rekenmachine: Druk op $\boxed{\text{ctrl}} \boxed{\text{enter}}$.

Windows®: Druk op **Ctrl+Enter**.

Macintosh®: Druk op **⌘+Enter**.

iPad®: Houd **Enter** ingedrukt en selecteer $\approx$.

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) \quad \{1., 0.707107, 0., 0.864976\}$$

° , ' , " (graad/minuut/seconde)

  -toetsen

$dd^{\circ}mm'ss.ss'' \Rightarrow$ uitdrukking

In de hoekmodus Graden:

dd Een positief of negatief getal

$25^{\circ}13'17.5''$ 25.2215

mm Een niet-negatief getal

$25^{\circ}30'$ $\frac{51}{2}$

$ss.ss$ Een niet-negatief getal

Geeft $dd+(mm/60)+(ss.ss/3600)$.

Met deze grondtal-60-invoeropmaak kunt u:

- Een hoek in graden/minuten/seconden invoeren, ongeacht de ingestelde hoekmodus.
- Tijd in uren/minuten/seconden invoeren.

Opmerking: laat $ss.ss$ volgen door twee apostrofes ('), niet door een dubbel aanhalingsteken (").

∠ (hoek)

  -toetsen

$[Straal, \angle \theta_Hoek] \Rightarrow$ vector

In de modus Radialen en met vectoropmaak ingesteld op:

(polaire invoer)

rechthoekig

$[Straal, \angle \theta_Hoek, Z_Coördinaat] \Rightarrow$ vector

(cilindrische invoer)

$[5 \angle 60^{\circ} \angle 45^{\circ}] \left[\frac{5 \cdot \sqrt{2}}{4} \quad \frac{5 \cdot \sqrt{6}}{4} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$

$[Straal, \angle \theta_Hoek, \angle \theta_Hoek] \Rightarrow$ vector

(bolvormige invoer)

cilindrisch

Geeft coördinaten als een vector op basis van de modusinstelling voor vectoropmaak: rechthoekig, cilindrisch of bolvormig.

$[5 \angle 60^{\circ} \angle 45^{\circ}] \left[\frac{5 \cdot \sqrt{2}}{2} \quad \angle \frac{\pi}{3} \quad \frac{5 \cdot \sqrt{2}}{2} \right]$

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @< in te voeren.

bolvormig

$[5 \angle 60^{\circ} \angle 45^{\circ}] \left[5 \angle \frac{\pi}{3} \angle \frac{\pi}{4} \right]$

$(Grootte \angle Hoek) \Rightarrow$ complexe Waarde

In de hoekmodus Radialen en rechthoekige complexe opmaak:

(polaire invoer)

$5+3 \cdot i \left(10 \angle \frac{\pi}{4} \right) \quad 5-5 \cdot \sqrt{2} + (3-5 \cdot \sqrt{2}) \cdot i$

∠ (hoek)

  -toetsen

Voert een complexe waarde in ($r\angle\theta$) polaire vorm in. De *Hoek* wordt geïnterpreteerd volgens de huidige instelling van de hoekmodus.

Opmerking: Om een decimale benaderende uitkomst te forceren,

Rekenmachine: Druk op  .

Windows®: Druk op **Ctrl+Enter**.

Macintosh®: Druk op **⌘+Enter**.

iPad®: Houd **Enter** ingedrukt en selecteer



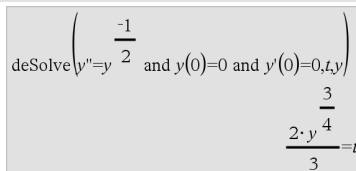
' (prime)

 -toetsen

variabele '

variabele ''

Voert een prime-symbool in een differentiaalvergelijking in. Een enkel prime-symbool duidt een differentiaalvergelijking van de eerste orde aan, twee prime-symbolen duiden een differentiaalvergelijking van de tweede orde aan, enz.



_ (onderstrepingssteken als een leeg element)

**Zie “Lege elementen”
pag. 267.**

_ (onderstrepingssteken als eenheidsaanduiding)

  -toetsen

Uitdr_Eenheid

3·_m►_ft 9.84252·_ft

Duidt de eenheden voor een *Uitdr* aan. Alle namen van eenheden moeten beginnen met een onderstrepingssteken.

Opmerking: u vindt het conversiesymbool,

►, in de Catalogus. Klik op  en klik vervolgens op **Wiskundige operatoren**.

U kunt voorgedefinieerde eenheden gebruiken of uw eigen eenheden creëren. Voor een lijst met voorgedefinieerde eenheden opent u de Catalogus en geeft u de tab Eenhedenconversies weer. U kunt namen van eenheden selecteren uit de Catalogus of de namen voor de eenheden direct intypen.

Variabele_

Aangenomen dat z onbepaald is:

(onderstrepingsteken als eenheidsaanduiding)

  -toetsen

Als *Variable* geen waarde heeft, dan wordt hij behandeld alsof hij een complex getal representeert. Standaard wordt de variabele zonder het als reëel behandeld.

Als *Variable* een waarde heeft, dan wordt het genegeerd en behoudt *Variable* zijn oorspronkelijke gegevenstype.

Opmerking: u kunt een complex getal in een variabele opslaan zonder te gebruiken. Voor de beste resultaten in berekeningen als **cSolve()** en **cZeros()** wordt echter aangeraden om het te gebruiken.

$\text{real}(z)$	z
$\text{real}(z_{\text{—}})$	$\text{real}(z_{\text{—}})$
$\text{imag}(z)$	0
$\text{imag}(z_{\text{—}})$	$\text{imag}(z_{\text{—}})$

► (convert)

  -toetsen

Uitdr_Eenheid1 ► *_Eenheid2* ⇒ *Uitdr_ Eenheid2*

3·_m ►_ft 9.84252·_ft

Converteert een uitdrukking van de ene eenheid naar een andere.

Het onderstrepingsteken duidt de eenheden aan. De eenheden moeten in dezelfde categorie vallen, zoals Lengte of Oppervlakte.

Voor een lijst met voorgedefinieerde eenheden opent u de Catalogus en geeft u de tab Eenhedenconversies weer:

- U kunt de naam van een eenheid selecteren uit de lijst.
- U kunt de conversie-operator, ►, selecteren bovenaan de lijst.

U kunt de namen van eenheden ook met de hand intypen. Om “” te typen als u namen van eenheden op de rekenmachine typt, drukt u op  .

Opmerking: gebruik **tmpCnv()** en **ΔtmpCnv()** om temperatuureenheden te converteren. De conversie-operator ► converteert geen temperatuureenheden.

10^()

Catalogus >

10^ (Uitdr1)⇒*uitdrukking*

$$10^{1.5} \quad 31.6228$$

10^ (Lijst1)⇒*lijst*

$$10^{\{0, -2.2, a\}} \quad \left\{1, \frac{1}{100}, 100, 10^a\right\}$$

Geeft 10 tot de macht van het argument.

Geeft bij een lijst 10 tot de macht van de elementen in *Lijst1*.**10^ (vierkanteMatrix1)**⇒*vierkanteMatrix*Geeft 10 tot de macht van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van 10 tot de macht van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} \quad \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

vierkanteMatrix1 moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.**^-1 (omgekeerde)**

Catalogus >

Uitdr1 ^-1⇒*uitdrukking*

$$(3.1)^{-1} \quad 0.322581$$

Lijst1 ^-1⇒*lijst*

$$\{a, 4, -0.1, x, -2\}^{-1} \quad \left\{\frac{1}{a}, \frac{1}{4}, -10, \frac{1}{x}, \frac{-1}{2}\right\}$$

Geeft de omgekeerde van het argument.

Geeft bij een lijst de omgekeerden van de elementen in *Lijst1*.*vierkanteMatrix1* ^-1⇒*vierkanteMatrix*Geeft de inverse van *vierkanteMatrix1*.*vierkanteMatrix1* moet een niet-singuliere vierkante matrix zijn.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} \frac{-2}{a-2} & \frac{1}{a-2} \\ \frac{a}{2 \cdot (a-2)} & \frac{-1}{2 \cdot (a-2)} \end{bmatrix}$$

| (beperkende operator)

-toetsen

Uitdr | BooleaanseUitdr1
[andBooleaanseUitdr2]...

$$x+1|x=3 \quad 4$$

$$x+y|x=\sin(y) \quad \sin(y)+y$$

Uitdr | BooleaanseUitdr1
[orBooleaanseUitdr2]...

$$x+y|\sin(y)=x \quad x+y$$

Het beperkingssymbool ("|") dient als een binaire operator. De operand aan de linkerkant van | is een uitdrukking. De operand aan de rechterkant van | specificeert één of meer relaties die bedoeld zijn om de vereenvoudiging van de uitdrukking te beïnvloeden. Meerdere relaties na | moeten gekoppeld worden door logische **"and"** of **"or"**-operatoren.

De beperkings-operator biedt drie basistypen functionaliteit:

- Substituties
- Intervalbeperkingen
- Uitsluitingen

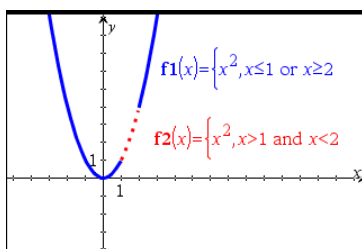
Substituties zijn in de vorm van een gelijkheid, zoals $x=3$ of $y=\sin(x)$. Om het meest effectief te zijn moet de linkerkant een enkelvoudige variabele zijn. *Uitdr | Variabele = waarde* substitueert *waarde* elke keer dat *Variabele* voorkomt in *Uitdr*.

Intervalbeperkingen kunnen de vorm aannemen van één of meer ongelijkheden die gekoppeld worden door logische **"and"** of **"or"**-operatoren. Intervalbeperkingen maken ook vereenvoudigingen mogelijk, die anders ongeldig of niet te berekenen zouden kunnen zijn.

Uitsluitingen gebruiken de relationele operator "is niet gelijk aan" ($\neq$ of $\neq$) om een specifieke waarde buiten beschouwing te laten. Ze worden voornamelijk gebruikt om een exacte oplossing uit te sluiten bij het gebruik van **cSolve()**, **cZeros()**, **fMax()**, **fMin()**, **solve()**, **zeros()** etc.

$x^3-2\cdot x+7\rightarrow f(x)$	Done
$f(x) x=\sqrt{3}$	$\sqrt{3}+7$
$(\sin(x))^2+2\cdot\sin(x)-6 \sin(x)=d$	$d^2+2\cdot d-6$

$\text{solve}(x^2-1=0,x) x>0 \text{ and } x<2$	$x=1$
$\sqrt{x}\cdot\sqrt{\frac{1}{x}} x>0$	1
$\sqrt{x}\cdot\sqrt{\frac{1}{x}}$	$\sqrt{\frac{1}{x}}\cdot\sqrt{x}$



$\text{solve}(x^2-1=0,x) x\neq 1$	$x=-1$
-----------------------------------	--------

→ (opslaan)

ctrl var -toets

Uitdr → *Var*

$$\frac{\pi}{4} \rightarrow myvar \quad \frac{\pi}{4}$$

Lijst → *Var*

$$2 \cdot \cos(x) \rightarrow y1(x) \quad Done$$

Matrix → *Var*

$$\{1, 2, 3, 4\} \rightarrow lst5 \quad \{1, 2, 3, 4\}$$

Uitdr → *Functie*(*Param1*,...)

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

Lijst → *Functie*(*Param1*,...)

$$"Hello" \rightarrow str1 \quad "Hello"$$

Matrix → *Functie*(*Param1*,...)

Als de variabele *Var* niet bestaat, dan wordt deze gecreëerd en geïnitieerd naar *Uitdr*, *Lijst* of *Matrix*.

Als de variabele *Var* reeds bestaat en niet vergrendeld of beveiligd is, dan wordt de inhoud ervan vervangen door *Uitdr*, *Lijst* of *Matrix*.

Tip: als u symbolische berekeningen wilt uitvoeren met behulp van ongedefinieerde variabelen, sla dan niets op in veelgebruikte éénletterige variabelen zoals a, b, c, x, y, z enz.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door =: als sneltoets in te voeren. Typ bijvoorbeeld `pi/4 =: mijnvar`.

:= (toewijzen)

ctrl |toetsen -toetsen

Var := *Uitdr*

$$myvar := \frac{\pi}{4} \quad \frac{\pi}{4}$$

Var := *Lijst*

$$y1(x) := 2 \cdot \cos(x) \quad Done$$

Var := *Matrix*

$$lst5 := \{1, 2, 3, 4\} \quad \{1, 2, 3, 4\}$$

Functie(*Param1*,...) := *Uitdr*

$$matg := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Functie(*Param1*,...) := *Lijst*

$$str1 := "Hello" \quad "Hello"$$

Functie(*Param1*,...) := *Matrix*

Als variabele *Var* niet bestaat, dan wordt *Var* gecreëerd en geïnitieerd naar *Uitdr*, *Lijst* of *Matrix*.

:= (toewijzen)

  -toetsen

Als *Var* reeds bestaat en niet vergrendeld of beveiligd is, dan wordt de inhoud ervan vervangen door *Uitdr*, *Lijst* of *Matrix*.

Tip: als u symbolische berekeningen wilt uitvoeren met behulp van ongedefinieerde variabelen, sla dan niets op in veelgebruikte éénletterige variabelen zoals *a*, *b*, *c*, *x*, *y*, *z* enz.

© (commentaar)

  -toetsen

© [*tekst*]

© verwerkt *tekst* als een commentaarregel, waardoor u door u gecreëerde functies en programma's kunt annoteren.

© kan aan het begin of op een willekeurige plaats in de regel staan. Alles rechts van ©, tot aan het eind van de regel, is het commentaar.

Opmerking bij het invoeren van het

voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

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

  -toetsen,   -toetsen

0b *binairGetal*

In de Dec-grondtalmodus:

0h *hexadecimaalGetal*

0b10+0hF+10

27

Duidt respectievelijk een binair of hexadecimaal getal aan. Om een binair of hexadecimaal getal in te voeren moet u het 0b- of 0h-prefix invoeren, ongeacht de instelling van de grondtal-modus. Zonder prefix wordt een getal behandeld als decimaal (grondtal 10).

In de Bin-grondtalmodus:

0b10+0hF+10

0b11011

Resultaten worden weergegeven volgens de grondtal-modus.

In de Hex-grondtalmodus:

0b10+0hF+10

0h1B

TI-Nspire™ CX II - Tekenopdrachten

Dit is een aanvullend document voor de TI-Nspire™-Referentiehandleiding en de TI-Nspire™-CAS-referentiehandleiding. Alle TI-Nspire™ CX II-opdrachten zullen worden opgenomen en gepubliceerd in versie 5.1 van de TI-Nspire™-referentiehandleiding en de TI-Nspire™-CAS-referentiehandleiding.

Programmeren van grafische weergaven

Nieuwe opdrachten zijn toegevoegd aan TI-Nspire™ CX II-rekenmachines en TI-Nspire-desktopapplicaties voor het programmeren van grafische weergaven

De TI-Nspire™ CX II-rekenmachines schakelen over naar deze grafische modus tijdens het uitvoeren van grafische opdrachten en schakelen terug naar de context waarin het programma werd uitgevoerd na voltooiing van het programma.

Op het scherm verschijnt "Running ..." in de bovenste balk terwijl het programma wordt uitgevoerd. Het toont "Finished" wanneer het programma is voltooid. Een druk op een willekeurige toets brengt het systeem uit de grafische modus.

- De overgang naar de grafische modus wordt automatisch geactiveerd wanneer een van de tekenopdrachten (afbeeldingen) wordt aangetroffen tijdens de uitvoering van het TI Basic-programma.
- Deze overgang gebeurt alleen als een programma vanaf de rekenmachine wordt uitgevoerd; in een document of rekenmachine in het kladblok.
- De overgang van de grafische modus gebeurt bij het beëindigen van het programma.
- De grafische modus is alleen beschikbaar op de TI-Nspire™ CX II-rekenmachines en de desktop TI-Nspire™ CX II-rekenmachines. Dit betekent dat het niet beschikbaar is in de computerdocumentweergave of PublishView (.tnsp) op het bureaublad en niet op iOS.
 - Als een grafische opdracht wordt aangetroffen tijdens het uitvoeren van een TI Basic-programma vanuit de onjuiste context, wordt een foutmelding weergegeven en wordt het TI Basic-programma beëindigd.

Grafisch scherm

Het grafische scherm bevat een titeltekst bovenaan het scherm waarop niet kan worden geschreven door grafische opdrachten.

Het tekengebied van het grafische scherm wordt gewist (kleur = 255,255,255) wanneer het grafische scherm wordt geïnitieerd.

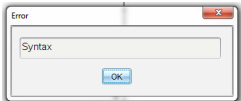
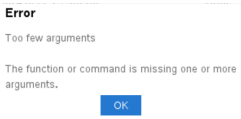
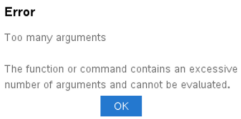
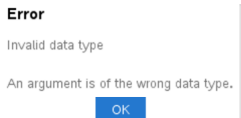
Grafisch scherm	Standaard
Hoogte	212
Breedte	318
Kleur	wit: 255,255,255

Standaardweergave en instellingen

- De statuspictogrammen in de bovenste balk (batterijstatus, teststatus, netwerkindicator, enzovoort) zijn niet zichtbaar als een grafisch programma wordt uitgevoerd.
- Standaard tekenkleur: Zwart (0,0,0)
- Standaard penstijl - normaal, glad
 - Dikte: 1 (dun), 2 (normaal), 3 (dikste)
 - Stijl 1 (glad), 2 (gestippeld), 3 (gestreept)
- Alle tekenopdrachten gebruiken de huidige instellingen voor kleur en pen; standaardwaarden of waarden die zijn ingesteld met TI-Basic-opdrachten.
- Het lettertype dat voor de tekst gebruikt wordt, ligt vast en kan niet worden gewijzigd.
- Alle uitvoer naar het grafische scherm wordt getekend in een uitknipvenster dat de grootte heeft van het tekengebied van het grafische scherm. Alle getekende uitvoer die buiten dit geknipte gebied van tekening van het grafische scherm valt, wordt niet getekend. Er wordt geen foutmelding weergegeven.
- Alle x, y-coördinaten die zijn opgegeven voor tekenopdrachten, worden zodanig gedefinieerd dat 0,0 zich in de linkerbovenhoek van het tekengebied van het grafische scherm bevindt.
 - **Uitzonderingen:**
 - **DrawText** gebruikt de coördinaten als de linkerbenedenhoek van het selectiekader voor de tekst.
 - **SetWindow** gebruikt de linkerbenedenhoek van het scherm
- Alle parameters voor de opdrachten kunnen worden geleverd als expressies die worden geëvalueerd naar een getal dat vervolgens wordt afgerond op het dichtstbijzijnde, gehele getal.

Foutmeldingen op het grafische scherm

Als de validatie mislukt, verschijnt er een foutmelding.

Foutbericht	Beschrijving	Weergave
Fout Syntax	Als de syntaxcontrole eventuele syntaxfouten vindt, wordt er een foutmelding weergegeven. Ook wordt dan geprobeerd de cursor bij de eerste fout te plaatsen, zodat u deze kunt corrigeren.	
Fout Te weinig argumenten	In de functie of het commando ontbreken één of meer argumenten	
Fout Te veel argumenten	De functie of opdracht bevat een te groot aantal argumenten en kan niet worden geëvalueerd.	
Fout Ongeldig gegevenstype	Een argument is van het verkeerde gegevenstype.	

Ongeldige opdrachten in de grafische modus

Sommige opdrachten zijn niet toegestaan als het programma overschakelt naar de grafische modus. Als deze opdrachten worden aangetroffen in de grafische modus, wordt er een foutmelding weergegeven en wordt het programma beëindigd.

Ontoelaatbare opdracht	Foutbericht
Verzoek	Verzoek kan niet in de grafische modus worden uitgevoerd
VerzoekStr	RequestStr kan niet in de grafische modus worden uitgevoerd
Tekst	Text kan niet in de grafische modus worden uitgevoerd

De opdrachten die tekst naar de rekenmachine verzenden - **disp** en **dispAt** - worden ondersteunde opdrachten in de grafische context. De tekst van deze opdrachten wordt naar het Calculatorscherm gestuurd (niet naar Graphics) en is zichtbaar nadat het programma is afgesloten en het systeem terugschakelt naar de Calculator-app

Wiss.

x , y , *breedte*, *hoogte* **wissen**

Wist het volledige scherm als er geen parameters zijn opgegeven.

Als x , y , *breedte* en *hoogte* zijn opgegeven, wordt de rechthoek gewist die door de parameters is gedefinieerd.

Wiss.

Wist het volledige scherm

10,10,100,50 **wissen**

Wist een rechthoekig gebied met linkerbovenhoek op (10, 10) en met breedte 100, hoogte 50

DrawArc

Catalogus > 
CXII

DrawArc *x, y, width, height, startAngle, arcAngle*

Teken een boog binnen de gedefinieerde begrenzende rechthoek met de voorziene begin- en booghoeken.

x, y: coördinaat linksboven van begrenzende rechthoek

breedte, hoogte: afmetingen van de grensrechthoek

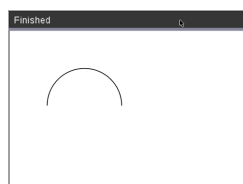
De "booghoek" definieert de zwaai van de boog.

Deze parameters kunnen worden geleverd als expressies die worden geëvalueerd naar een getal dat vervolgens wordt afgerond op het dichtstbijzijnde, gehele getal.

DrawArc 20,20,100,100,0,90



DrawArc 50,50,100,100,0,180



Zie ook: [FillArc](#)

DrawCircle

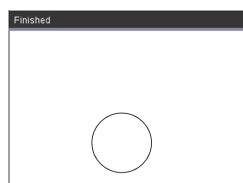
Catalogus > 
CXII

DrawCircle *x, y, radius*

x, y: coördinaat van het middelpunt

radius: radius van de cirkel

DrawCircle 150,150,40



Zie ook: [FillCircle](#)

DrawLine

Catalogus > 
CXII

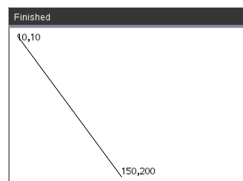
DrawLine $x1, y1, x2, y2$

Teken een lijn van $x1, y1, x2, y2$.

Uitdrukkingen die evalueren naar een getal dat vervolgens wordt afgerond op het dichtstbijzijnde, gehele getal.

Schermgrenzen: Als de opgegeven coördinaten ervoor zorgen dat een deel van de lijn buiten het grafische scherm wordt getekend, wordt dat gedeelte van de lijn afgekapt en wordt er geen foutmelding weergegeven.

DrawLine 10,10,150,200



DrawPoly

Catalogus > 
CXII

De opdrachten hebben twee varianten:

DrawPoly $xlist, ylist$

of

DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$

Opmerking: DrawPoly $xlist, ylist$

Shape verbindt $x1, y1$ to $x2, y2, x2, y2$ to $x3, y3$ enzovoort.

Opmerking: DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$ xn, yn wordt **NIET** automatisch verbonden met $x1, y1$.

Uitdrukkingen die evalueren naar een lijst met echte drijvers

$xlist, ylist$

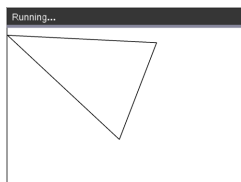
Uitdrukkingen die evalueren tot een enkele, echte drijver

$x1, y1 ... xn, yn$ = coördinaten voor hoekpunten van veelhoek

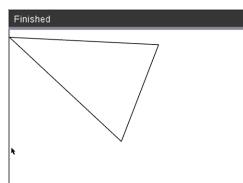
$xlist:=\{0,200,150,0\}$

$ylist:=\{10,20,150,10\}$

DrawPoly $xlist,ylist$



DrawPoly
 $0,10,200,20,150,150,0,10$



Opmerking: DrawPoly: Afmetingen invoer (breedte/hoogte) ten opzichte van getrokken lijnen. De lijnen worden getekend in een selectiekader rond de opgegeven coördinaat en dimensies, zodat de werkelijke grootte van de getekende polygoon groter zal zijn dan de breedte en hoogte.

Zie ook: [FillPoly](#)

DrawRect

DrawRect *x, y, breedte, hoogte*

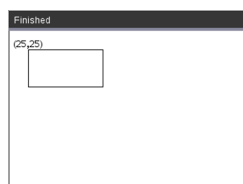
x, y: coördinaat linksboven van rechthoek

breedte, hoogte: breedte en hoogte van de rechthoek (rechthoek naar beneden getrokken en rechts vanaf de startcoördinaat).

Opmerking: De lijnen worden getekend in een selectiekader rond de opgegeven coördinaat en afmetingen, zodat de werkelijke grootte van de getekende rechthoek groter zal zijn dan de breedte en hoogte aangeven.

Zie ook: [FillRect](#)

DrawRect 25,25,100,50



DrawText

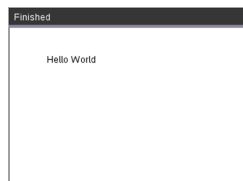
DrawText *x, y, exprOrString1 [,exprOrString2]...*

x, y: coördinaat van tekstuitvoer

Tekent de tekst in *exprOrString* op de opgegeven *x, y*-coördinaatlocatie.

De regels voor *exprOrString* zijn dezelfde als voor **Disp** – **DrawText** kan meerdere argumenten bevatten.

DrawText 50,50,"Hello World"



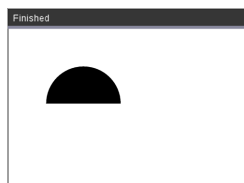
FillArcCatalogus > 
CXII**FillArc** *x, y, breedte, hoogte startAngle, arcAngle**x, y*: coördinaat linksboven van begrenzende rechthoek

Teken en vul een boog binnen de gedefinieerde begrenzende rechthoek met de voorziene begin- en booghoeken.

Standaard vulkleur is zwart. De vulkleur kan worden ingesteld met de opdracht [SetColor](#)

De "booghoek" definieert de zwaai van de boog

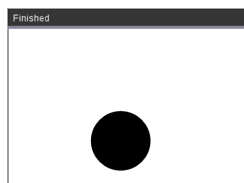
FillArc 50,50,100,100,0,180

**FillCircle**Catalogus > 
CXII**FillCircle** *x, y, radius**x, y*: coördinaat van het middelpunt

Teken en vul een cirkel in het opgegeven midden met de opgegeven radius.

Standaard vulkleur is zwart. De vulkleur kan worden ingesteld met de opdracht [SetColor](#).

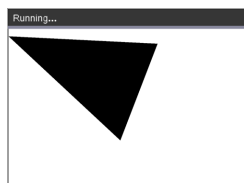
FillCircle 150,150,40



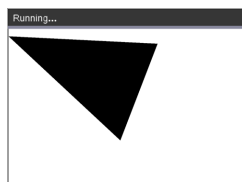
Hier!

FillPolyCatalogus > 
CXII**FillPoly** *xlist, ylist*

of

FillPoly *x1, y1, x2, y2, x3, y3...xn, yn***Opmerking:** De lijn en kleur worden gespecificeerd door [SetColor](#) en [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, breedte, hoogte*

FillRect 25, 25, 100, 50

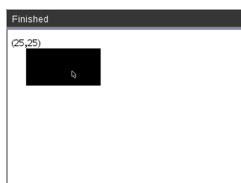
x, y: coördinaat linksboven van rechthoek

breedte, hoogte: breedte en hoogte van de rechthoek

Teken en vul een rechthoek met de linkerbovenhoek op de coördinaat die is opgegeven door (*x, y*)

Standaard vulkleur is zwart. De vulkleur kan worden ingesteld met de opdracht [SetColor](#)

Opmerking: De lijn en kleur worden gespecificeerd door [SetColor](#) en [SetPen](#)



getPlatform()Catalogus > 
CXII**getPlatform()**

getPlatform()

"dt"

Geeft als resultaat:

"dt" op desktop softwaretoepassingen

"hh" op TI-Nspire™ CX-rekenmachines

"ios" op de TI-Nspire™ CX iPad®-app

PaintBuffer

Catalogus > 
CXII

PaintBuffer

Verf grafische buffer naar scherm

Deze opdracht wordt gebruikt in combinatie met UseBuffer, om de weergavesnelheid op het scherm te verhogen wanneer het programma meerdere grafische objecten genereert.

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

Dit programma toont alle 10 cirkels tegelijkertijd.

Als de opdracht "UseBuffer" wordt verwijderd, wordt elke cirkel weergegeven zoals deze is getekend.

Zie ook: [UseBuffer](#)

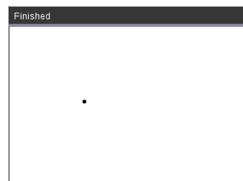
PlotXY *x, y, vorm*

x, y: coördinaat om vorm te plotten

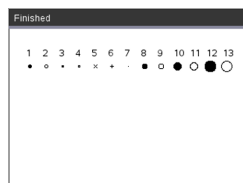
vorm: een getal tussen 1 en 13 dat de vorm specificeert

- 1 - Gevulde cirkel
- 2 - Lege cirkel
- 3 - Gevuld vierkant
- 4 - Leeg vierkant
- 5 - Kruis
- 6 - Plus
- 7 - Dun
- 8 - gemiddeld punt, vast
- 9 - gemiddeld punt, leeg
- 10 - groter punt, vast
- 11 - groter punt, leeg
- 12 - grootste punt, vast
- 13 - grootste punt, leeg

PlotXY 100,100,1

Voor *n*, 1, 13DrawText 1+22**n*, 40, *n*PlotXY 5+22**n*, 50, *n*

EndFor



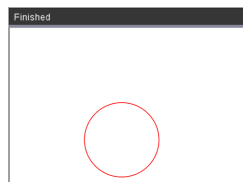
SetColorCatalogus > 
CXII**SetColor**

Roodwaarde, Groenwaarde, Blauwwaarde

Geldige waarden voor rood, groen en blauw liggen tussen 0 en 255

Stelt de kleur in voor de volgende tekenopdrachten

```
SetColor 255,0,0
DrawCircle 150,150,100
```

**SetPen**Catalogus > 
CXII**SetPen**

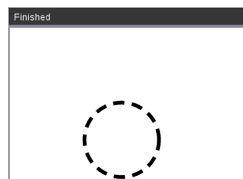
dikte, stijl

dikte: 1 <= dikte <= 3 | 1 is de dunste, 3 is de dikste

stijl: 1 = Egaal, 2 = Gestippeld, 3 = Onderbroken

Stelt de penstijl in voor de volgende tekenopdrachten

```
SetPen 3,3
DrawCircle 150,150,50
```

**SetWindow**Catalogus > 
CXII**SetWindow**

xMin, xMax, yMin, yMax

Hiermee stelt u een logisch venster in dat wordt toegewezen aan het grafische tekengebied. Alle parameters zijn verplicht.

Als het deel van het getekende object zich buiten het venster bevindt, wordt de uitvoer afgekapt (niet weergegeven) en wordt er geen foutmelding weergegeven.

```
SetWindow 0,160,0,120
```

zal het uitvoervenster instellen op 0,0 in de linkeronderhoek met een breedte van 160 en een hoogte van 120

```
DrawLine 0,0,100,100
```

```
SetWindow 0,160,0,120
```

```
SetPen 3,3
```

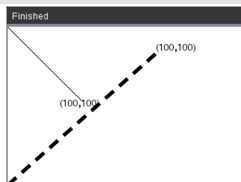
```
DrawLine 0,0,100,100
```

Als $x_{\min}$ groter is dan of gelijk is aan $x_{\max}$ of $y_{\min}$ is groter dan of gelijk aan $y_{\max}$, wordt een foutmelding weergegeven.

Alle objecten die vóór een SetWindow-opdracht zijn getekend, worden niet opnieuw getekend in de nieuwe configuratie.

Gebruik de volgende stappen om de vensterparameters opnieuw in te stellen:

SetWindow 0,0,0,0



UseBuffer

Catalogus > 
CXII

UseBuffer

Teken naar een niet-scherm grafische buffer in plaats van het scherm (om de prestaties te verbeteren)

Deze opdracht wordt gebruikt in combinatie met PaintBuffer, om de weergavesnelheid op het scherm te verhogen wanneer het programma meerdere grafische objecten genereert.

Met UseBuffer worden alle afbeeldingen pas weergegeven nadat de volgende PaintBuffer-opdracht is uitgevoerd.

UseBuffer hoeft maar één keer in het programma te worden aangeroepen, d.w.z. elk gebruik van PaintBuffer heeft geen overeenkomstige UseBuffer nodig

Zie ook: [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
```

Dit programma toont alle 10 cirkels tegelijkertijd.

Als de opdracht "UseBuffer" wordt verwijderd, wordt elke cirkel weergegeven zoals deze is getekend.

Lege elementen

Bij het analyseren van gegevens uit de werkelijkheid heeft u misschien niet altijd een volledige set gegevens. Op de TI-Nspire™ CAS zijn lege elementen toegestaan, zodat u kunt doorgaan met een bijna complete set gegevens in plaats van opnieuw te moeten te beginnen of de onvolledige gevallen te moeten weggooien.

U kunt een voorbeeld van een set gegevens met lege elementen vinden in het hoofdstuk Lijsten & Spreadsheet, onder “*Spreadsheet-gegevens in een grafiek tekenen.*”

Met de functie **delVoid()** kunt u lege elementen uit een lijst verwijderen. Met de functie **isVoid()** kunt u nagaan of er lege elementen zijn. Zie voor meer informatie **delVoid()**, pag. 52 en **isVoid()**, pag. 100.

Opmerking: Om handmatig een leeg element in een wiskundige uitdrukking in te voeren typt u “_” of het woord **void**. Het woord **void** wordt automatisch geconverteerd in een “_” symbool wanneer de uitdrukking wordt uitgewerkt. Om “_” te typen op de rekenmachine drukt u op  .

Berekeningen met lege elementen

Het merendeel van de berekeningen met een lege invoer levert een leeg resultaat op. Zie de speciale gevallen hieronder.

_	-
gcd(100,_)	-
3+_	-
{5,_,10}-{3,6,9}	{2,_,1}

Lijstargumenten met lege elementen

De volgende functies en opdrachten negeren (slaan over) lege elementen die worden aangetroffen in lijstargumenten.

count, **countIf**, **cumulativeSum**, **freqTable**→**list**, **frequency**, **max**, **mean**, **median**, **product**, **stDevPop**, **stDevSamp**, **sum**, **sumIf**, **varPop** en **varSamp**, evenals regressieberekeningen, **OneVar**, **TwoVar** en **FiveNumSummary** statistieken, betrouwbaarheidsintervallen en statistische toetsen

sum({2,_,3,5,6,6})	16.6
median({1,2,_,_,3})	2
cumulativeSum({1,2,_,4,5})	{1,3,_,7,12}
cumulativeSum($\begin{bmatrix} 1 & 2 \\ 3 & - \\ 5 & 6 \end{bmatrix}$)	$\begin{bmatrix} 1 & 2 \\ 4 & - \\ 9 & 8 \end{bmatrix}$

Lijstargumenten met lege elementen

SortA en **SortD** verplaatsen alle lege elementen binnen het eerste argument naar het eind.

$\{5,4,3,_,1\} \rightarrow list1$	$\{5,4,3,_,1\}$
$\{5,4,3,2,1\} \rightarrow list2$	$\{5,4,3,2,1\}$
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	$\{5,3,2,1,4\}$

In regressies zorgt een leeg element in een X- of Y-lijst voor een lege plaats voor het overeenkomstige element van het residu.

$l1:=\{1,2,3,4,5\}; l2:=\{2,_,3,5,6,6\}$	$\{2,_,3,5,6,6\}$
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	$\{1,_,1,1,1\}$

Een weggelaten categorie in regressies zorgt voor een lege plaats voor het overeenkomstige element van het residu.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
$cat:=\{ "M", "M", "F", "F" \}; incl:=\{ "F" \}$	$\{ "F" \}$
LinRegMx l1,l2,1,cat,incl	Done
stat.Resid	$\{_,_,0,0,0\}$
stat.XReg	$\{_,_,4,5\}$
stat.YReg	$\{_,_,5,6,6\}$
stat.FreqReg	$\{_,_,1,1,1\}$

Een frequentie van 0 in regressies zorgt voor een lege plaats voor het overeenkomstige element van het residu.

$l1:=\{1,3,4,5\}; l2:=\{2,3,5,6,6\}$	$\{2,3,5,6,6\}$
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	$\{1,_,1,1,1\}$

Snelkoppelingen voor het invoeren van wiskundige uitdrukkingen

Via snelkoppelingen kunt u elementen van wiskundige uitdrukkingen invoeren door ze in te typen in plaats van de Catalogus of het symboolpalet te gebruiken. Bijvoorbeeld: om de uitdrukking $\sqrt{6}$ in te voeren kunt u `sqrt(6)` typen op de invoerregel. Wanneer u op `enter` drukt, verandert de uitdrukking `sqrt(6)` in $\sqrt{6}$. Bepaalde snelkoppelingen zijn handig vanaf zowel de rekenmachine als het toetsenbord van de computer. Andere snelkoppelingen zijn voornamelijk handig vanaf het toetsenbord van de computer.

Vanaf de rekenmachine of het toetsenbord van de computer

Om dit in te voeren:	Typ deze snelkoppeling:
π	<code>pi</code>
θ	<code>theta</code>
∞	<code>infinity</code>
$\leq$	<code><=</code>
$\geq$	<code>>=</code>
$\neq$	<code>/=</code>
$\Rightarrow$ (logische implicatie)	<code>=></code>
$\Leftrightarrow$ (logische dubbele implicatie, XNOR)	<code><=></code>
$\rightarrow$ (opslag-operator)	<code>=:</code>
$ $ (absolute waarde)	<code>abs(...)</code>
$\sqrt{}$	<code>sqrt(...)</code>
$d()$	<code>derivative(...)</code>
$\int()$	<code>integral(...)</code>
$\Sigma()$ (Som-template)	<code>sumSeq(...)</code>
$\Pi()$ (Product-template)	<code>prodSeq(...)</code>
$\sin^{-1}()$, $\cos^{-1}()$, ...	<code>arcsin(...)</code> , <code>arccos(...)</code> , ...
$\Delta\text{List}()$	<code>deltaList(...)</code>
$\Delta\text{tmpCnv}()$	<code>deltaTmpCnv(...)</code>

Vanaf het toetsenbord van de computer

Om dit in te voeren:	Typ deze snelkoppeling:
<code>c1</code> , <code>c2</code> ... (constanten)	<code>@c1</code> , <code>@c2</code> , ...

Om dit in te voeren:	Typ deze snelkoppeling:
$n1, n2...$ (gehele constanten)	@n1, @n2, ...
i (imaginaire constante)	@i
e (natuurlijke logaritme grondtal e)	@e
E (wetenschappelijke notatie)	@E
T (transponeren)	@t
r (radialen)	@r
$^\circ$ (graden)	@d
$^\circ$ (decimale graden)	@g
$\angle$ (hoek)	@<
$\triangleright$ (conversie)	@>
$\triangleright$ Decimal, $\triangleright$ approxFraction (), en zo verder.	@>Decimal, @>approxFraction () , en zo verder.

Constanten en waarden

De volgende tabel vermeldt de constanten en hun waarden die beschikbaar zijn bij het uitvoeren van eenheidsconversies. Ze kunnen met de hand worden ingetypt of geselecteerd uit de lijst met **Constanten in Utilities > Unit Conversions** (Rekenmachine: druk op  3).

constante	Naam	Waarde
_c	lichtsnelheid	299792458 _m/_s
_Cc	Coulomb-constante	8987551787.3682 _m/_F
_Fc	Constante van Faraday	96485.33289 _coul/_mol
g	zwaartekrachtversnelling	9.80665 _m/_s ²
_Gc	zwaartekrachtconstante	6.67408E-11 _m ³ /_kg/_s ²
_h	Constante van Planck	6.626070040E-34 _J _s
_k	constante van Boltzmann	1.38064852E-23 _J/_°K
_μ0	permeabiliteit van een vacuüm	1.2566370614359E-6 _N/_A ²
_μb	Bohr magneton	9.274009994E-24 _J _m ² /_Wb
_Me	elektronenrustmassa	9.10938356E-31 _kg
_Mμ	muon massa	1.883531594E-28 _kg
_Mn	neutronenrustmassa	1.674927471E-27 _kg
_Mp	protonenrustmassa	1.672621898E-27 _kg
_Na	getal van Avogadro	6.022140857E23 /_mol
_q	elektronenlading	1.6021766208E-19 _coul
_Rb	Bohr-straal	5.2917721067E-11 _m
_Rc	Molaire gasconstante	8.3144598 _J/_mol/_°K
_Rdb	Rydbergconstante	10973731.568508/_m
_Re	elektronstraal	2.8179403227E-15 _m
_u	atomaire massa	1.660539040E-27 _kg
_Vm	molair volume	2.2413962E-2 _m ³ /_mol
_ε0	diëlektrische constante van een vacuüm	8.8541878176204E-12 _F/_m
_σ	constante van Stefan-Boltzmann	5.670367E-8 _W/_m ² /_°K ⁴
_φ0	magnetisch flux-quantum	2.067833831E-15 _Wb

EOS (Equation Operating System)-hiërarchie

In deze paragraaf wordt het Equation Operating System (EOS™) beschreven, dat gebruikt wordt door de TI-Nspire™ CAS-technologie voor wiskunde en exacte vakken. Getallen, variabelen en functies worden ingevoerd in een eenvoudige, duidelijke volgorde. De EOS™-software werkt uitdrukkingen en vergelijkingen uit met behulp van groepering met haakjes en volgens de hieronder beschreven voorrangsregels.

Volgorde van uitwerking

Niveau	Operator
1	Haakjes (), vierkante haken [], accolades { }
2	Indirectie (#)
3	Functieaanroepen
4	Navolgende operatoren: graden-minuten-seconden (°,'"), faculteit (!), percentage (%), radialen (°), subscript ([]), transponeren (T)
5	Machtsverheffen, macht-operator (^)
6	Negatie (-)
7	Aaneenvoeging van tekenreeksen (&)
8	Vermenigvuldigen (•), delen (/)
9	Optellen (+), aftrekken (-)
10	Gelijkheidsrelaties: is gelijk aan (=), is niet gelijk aan (≠ of /=), kleiner dan (<), kleiner dan of gelijk aan (≤ of <=), groter dan (>), groter dan of gelijk aan (≥ of >=)
11	Logisch niet
12	Logisch en
13	Logisch or
14	xof, noch, niet en
15	Logische implicatie (⇒)
16	Logische dubbele implicatie, XNOR (⇔)
17	Beperkende operator (" ")
18	Opslaan (→)

Haakjes, vierkante haken en accolades

Alle berekeningen binnen haakjes, vierkante haken of accolades worden het eerst uitgewerkt. Bijvoorbeeld: in de uitdrukking $4(1+2)$, werkt de EOS™-software eerst het gedeelte van de uitdrukking binnen de haakjes uit, $1+2$, en vermenigvuldigt vervolgens het resultaat, 3, met 4.

Het aantal openings- en sluithaakjes, vierkante haken en accolades moet hetzelfde zijn binnen een uitdrukking of vergelijking. Als dit niet het geval is, dan verschijnt er een foutmelding met het ontbrekende element. Bijvoorbeeld: bij $(1+2)/(3+4)$ wordt de foutmelding "Ontbrekende)" weergegeven.

Opmerking: omdat u met de TI-Nspire™ CAS-software uw eigen functies kunt definiëren, wordt een variabelenaam die wordt gevolgd door een uitdrukking tussen haakjes, beschouwd als een "functieaanroep" in plaats van als een impliciete vermenigvuldiging. Bijvoorbeeld: $a(b+c)$ is de functie a uitgewerkt voor $b+c$. Om de uitdrukking $b+c$ te vermenigvuldigen met de variabele a moet u expliciete vermenigvuldiging gebruiken: $a*(b+c)$.

indirectie

De indirectie-operator (#) converteert een string naar een variabele- of functienaam. Bijvoorbeeld: `#("x"&"y"&"z")` creëert de variabelenaam `xyz`. Met indirectie kunt u ook variabelen binnen een programma creëren en wijzigen. Bijvoorbeeld: als `10→r` en `"r"→s1`, dan `#s1=10`.

Navolgende operatoren

Navolgende operatoren zijn operatoren die direct na het argument komen, zoals $5!$, 25% of $60^{\circ}15'45''$. Argumenten gevolgd door een navolgende operator worden uitgewerkt op het vierde prioriteitsniveau. Bijvoorbeeld: in de uitdrukking $4^3!$, wordt eerst $3!$ uitgewerkt. Het resultaat, 6, wordt vervolgens de exponent van 4, en dit levert 4096 op.

Machtsverheffen

Machtsverheffen (^) en element-voor-element-machtsverheffen (.^) worden uitgewerkt van rechts naar links. Bijvoorbeeld: de uitdrukking 2^3^2 wordt op dezelfde manier uitgewerkt als 2^3^2 , en heeft als resultaat 512. Dit verschilt van $(2^3)^2$, wat 64 oplevert.

Negatie

Om een negatief getal in te voeren drukt u op $\boxed{-}$ gevolgd door het getal. Navolgende bewerkingen en machtsverheffen worden uitgevoerd vóór negatie. Bijvoorbeeld: het resultaat van $-x^2$ is een negatief getal, en $-9^2 = -81$. Gebruik haakjes om het kwadraat van een negatief getal te berekenen, zoals $(-9)^2$, wat 81 als resultaat heeft.

Beperking ("|")

Het argument dat volgt op de beperkende operator ("|") biedt een serie beperkingen die van invloed zijn op de uitwerking van het argument dat voorafgaat aan de operator.

TI-Nspire CX II - Functies van de TI-Basic-programmering

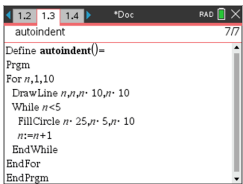
Auto-inspringen in de programmeereditor

De TI-Nspire™-programma-editor slaat nu automatisch instructies binnen een blokopdracht op.

Blokopdrachten zijn If/EndIf, For/EndFor, While/EndWhile, Loop/EndLoop, Try/EndTry

De editor zal automatisch spaties toevoegen aan programmaopdrachten binnen een blokopdracht. De opdracht om het blok te sluiten zal worden afgestemd met de opdracht om het blok te openen.

Het onderstaande voorbeeld toont auto-inspringing in geneste blokopdrachten.



```
autoindent
Define autoindent()=
Prgm
For n,1,10
  DrawLine n,n,n-10,n-10
  While n<5
    FillCircle n-25,n-5,n-10
    n=n+1
  EndWhile
EndFor
EndPrgm
```

Codefragmenten die worden gekopieerd en geplakt, behouden de oorspronkelijke inspringing.

Als u een programma opent dat is gemaakt in een eerdere versie van de software, blijft de oorspronkelijke inspringing behouden.

Verbeterde foutmeldingen voor TI-Basic

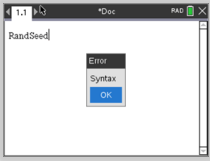
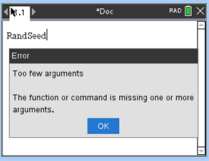
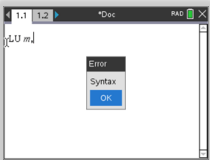
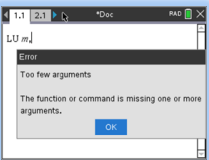
Fouten

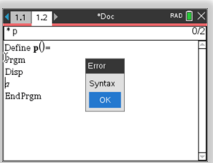
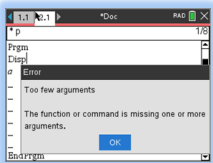
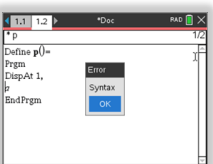
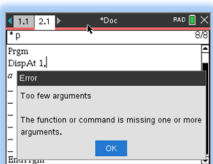
Fouttoestand	Nieuwe melding
Fout in conditieoverzicht (If/While)	Een conditionele instructie gaf geen oplossing voor TRUE of FALSE OPMERKING: Met de wijziging om de cursor op de regel met de fout te plaatsen, hoeven we niet langer te specificeren of de fout in de instructie "If" of "While" staat.
Ontbrekende EndIf	Verwachtte EndIf , maar vond een ander eindresultaat
Ontbrekende EndFor	Verwachtte EndFor , maar vond een ander eindresultaat
Ontbrekende EndWhile	Verwachtte EndWhile , maar vond een ander eindresultaat

Fouttoestand	Nieuwe melding
Ontbrekende EndLoop	Verwachtte EndLoop , maar vond een ander eindresultaat
Ontbrekende EndTry	Verwachtte EndTry , maar vond een ander eindresultaat
"Then" weggelaten na If <condition>	Ontbrekende If..Then
"Then" weggelaten na Elseif <condition>	Then ontbreekt in blok: Elseif .
Wanneer "Then", "Else" en "Elseif" werden aangetroffen buiten controleblokken	Else ongeldig buiten de blokken: If..Then..EndIf of Try..EndTry
"Elseif" verschijnt buiten het blok "If..Then..EndIf"	Elseif ongeldig buiten het blok: If..Then..EndIf
"Then" verschijnt buiten het blok "If.... EndIf"	Then ongeldig buiten het blok: If..EndIf

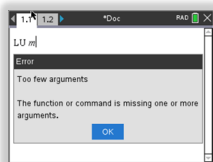
Syntaxfouten

In het geval dat opdrachten die verwachten dat een of meer argumenten worden aangeroepen met een onvolledige lijst met argumenten, wordt de foutmelding "**Te weinig argumenten**" weergegeven in plaats van "**syntaxisfout**"

Huidige werking	Nieuwe werking van CX II
 The screenshot shows the TI-84 Plus CE calculator interface. The command 'RandSeed' is entered. An error dialog box is displayed with the title 'Error' and the message 'Syntax'. There is an 'OK' button at the bottom.	 The screenshot shows the TI-84 Plus CE calculator interface. The command 'RandSeed' is entered. An error dialog box is displayed with the title 'Error' and the message 'Too few arguments'. Below this, it says 'The function or command is missing one or more arguments.' There is an 'OK' button at the bottom.
 The screenshot shows the TI-84 Plus CE calculator interface. The command 'L1 m1' is entered. An error dialog box is displayed with the title 'Error' and the message 'Syntax'. There is an 'OK' button at the bottom.	 The screenshot shows the TI-84 Plus CE calculator interface. The command 'L1 m1' is entered. An error dialog box is displayed with the title 'Error' and the message 'Too few arguments'. Below this, it says 'The function or command is missing one or more arguments.' There is an 'OK' button at the bottom.

Huidige werking	Nieuwe werking van CX II
	
	

Opmerking: Wanneer een onvolledige lijst met argumenten niet door een komma wordt gevolgd, is de foutmelding: "te weinig argumenten". Dit is hetzelfde als in vorige releases.



Foutcodes en meldingen

Als er een fout optreedt, wordt de code ervan toegekend aan de variabele *errCode*. Door de gebruiker gedefinieerde programma's en functies kunnen *errCode* onderzoeken om de oorzaak van een fout vast te stellen. Zie voor een voorbeeld van het gebruik van *errCode* voorbeeld 2 onder het commando **Try**, pag. 203.

Opmerking: sommige foutcondities zijn alleen van toepassing op TI-Nspire™ CAS-producten, en sommige zijn alleen van toepassing op TI-Nspire™-producten.

Foutcode	Beschrijving
10	Een functie heeft geen waarde teruggegeven
20	Een test heeft niet geresulteerd in WAAR of ONWAAR. Over het algemeen kunnen ongedefinieerde variabelen niet worden vergeleken. Bijvoorbeeld: de test $If\ a < b$ veroorzaakt deze fout als a of b ongedefinieerd is wanneer de If -bewering wordt uitgevoerd.
30	Argument kan niet de naam zijn van een map.
40	Argumentfout
50	Argumenten komen niet overeen Twee of meer argumenten moeten van hetzelfde type zijn.
60	Argument moet een Booleanse uitdrukking of geheel getal zijn
70	Argument moet een decimaal getal zijn
90	Argument moet een lijst zijn
100	Argument moet een matrix zijn
130	Argument moet een string zijn
140	Argument moet een variabelenaam zijn. Zorg ervoor dat de naam: • niet met een cijfer begint • geen spaties of speciale tekens bevat • geen onderstrepings tekens of punt op een ongeldige manier gebruikt • de lengtebeperkingen niet overschrijdt Zie voor meer informatie het hoofdstuk Rekenmachine in de documentatie.
160	Argument moet een uitdrukking zijn
165	Batterijen zijn te zwak om te verzenden of te ontvangen Installeer nieuwe batterijen voordat u verzendt of ontvangt.
170	Grens

Foutcode	Beschrijving
	De ondergrens moet lager zijn dan de bovengrens om het zoekinterval te definiëren.
180	Afbreken De  - of  -toets is ingedrukt tijdens een lange berekening of tijdens de uitvoering van een programma.
190	Cirkeldefinitie Deze foutmelding wordt weergegeven om te voorkomen dat het geheugen volraakt tijdens oneindige vervanging van variabelewaarden tijdens een vereenvoudiging. Bijvoorbeeld: $a+1 \rightarrow a$, waar a een niet-gedefinieerde variabele is, zal deze fout veroorzaken.
200	Beperkingsuitdrukking ongeldig Bijvoorbeeld: $\text{solve}(3x^2-4=0, x) \mid x < 0 \text{ or } x > 5$ zou deze foutmelding geven, omdat de beperkende voorwaarde gescheiden wordt door “or” en niet door “and”.
210	Ongeldig gegevenstype Een argument is van het verkeerde gegevenstype.
220	Afhankelijke grenswaarde
230	Afmeting Een lijst- of matrixindex is niet geldig. Bijvoorbeeld: als de lijst {1,2,3,4} is opgeslagen in L1, dan is L1[5] een afmetingsfout omdat L1 slechts vier elementen heeft.
235	Dimensies komen niet overeen. Niet genoeg elementen in de lijsten.
240	Dimensies komen niet overeen Twee of meer argumenten moeten dezelfde dimensies hebben. Bijvoorbeeld: in [1,2] + [1,2,3] komen de dimensies niet overeen omdat de matrices een verschillend aantal elementen bevatten.
250	Delen door nul
260	Domeinfout Een argument moet in een gespecificeerd domein liggen. Bijvoorbeeld rand(0) is niet geldig.
270	Dubbele variabelenaam
280	Else en Elseif zijn ongeldig buiten het If..EndIf-blok
290	EndTry mist het overeenkomende Else-voorschrift
295	Te grote iteratie
300	Lijst of matrix verwacht met 2 of 3 elementen

Foutcode	Beschrijving
310	Het eerste argument van nSolve moet een vergelijking met één variabele zijn. Deze kan geen variabele zonder waarde bevatten anders dan de relevante variabele.
320	Eerste argument van solve of cSolve moet een vergelijking of ongelijkheid zijn. Bijvoorbeeld: solve($3x^2-4$, x) is ongeldig omdat het eerste argument geen vergelijking is.
345	Inconsistente eenheden
350	Index buiten het bereik
360	Indirectiestring is geen geldige variabelenaam
380	Ongedefinieerd Ans Of de eerdere berekening heeft geen Ans gecreëerd, of er is geen eerdere berekening ingevoerd.
390	Ongeldige toewijzing
400	Ongeldige toewijzingswaarde
410	Ongeldig commando
430	Ongeldig voor de huidige modusinstellingen
435	Ongeldige gok
440	Ongeldige impliciete vermenigvuldiging $x(x+1)$ is bijvoorbeeld ongeldig; terwijl $x*(x+1)$ de correcte syntax is. Dit is om verwarring tussen impliciete vermenigvuldiging en functienotatie te voorkomen.
450	Ongeldig in een functie of de huidige uitdrukking Alleen bepaalde opdrachten zijn geldig in een door de gebruiker gedefinieerde functie.
490	Ongeldig in het Try..EndTry-blok
510	Ongeldige lijst of matrix
550	Ongeldig buiten functie of programma Een aantal commando's is niet geldig buiten een functie of programma. Bijvoorbeeld Local kan niet gebruikt worden tenzij in een functie of programma.
560	Ongeldig buiten de blokken Loop..EndLoop, For..EndFor of While..EndWhile Bijvoorbeeld: de opdracht Exit is alleen geldig binnen deze lus-blokken.
565	Ongeldig buiten programma
570	Ongeldige padnaam Bijvoorbeeld: \var is ongeldig.

Foutcode	Beschrijving
575	Ongeldig polair complex
580	Ongeldige programmaverwijzing Er kan niet verwezen worden naar programma's binnen functies of uitdrukkingen, zoals $1+p(x)$, waarbij p een programma is.
600	Ongeldige tabel
605	Ongeldig gebruik van eenheden
610	Ongeldige variabelenaam in een Local-bewering
620	Ongeldige variabele of functienaam
630	Ongeldige variabelereferentie
640	Ongeldige vectorsyntax
650	Gegevensoverdracht Link Een gegevensoverdracht tussen twee eenheden is niet uitgevoerd. Controleer of de verbindingskabel tussen de twee eenheden aan beide zijden goed is aangesloten.
665	Matrix niet diagonaliseerbaar
670	Geheugen bijna vol 1. Wis enkele gegevens in dit document 2. Sla dit document op en sluit het Als 1 en 2 mislukken, haal de batterijen er dan uit en plaats ze weer terug.
672	Bron uitgeput
673	Bron uitgeput
680	Ontbrekend (
690	Ontbrekend)
700	Ontbrekend “
710	Ontbrekend]
720	Ontbrekend }
730	Ontbrekend begin of eind van bloksyntax
740	Ontbrekend Then in het blok If..EndIf
750	Naam is geen functie of programma
765	Geen functies geselecteerd
780	Geen oplossing gevonden

Foutcode	Beschrijving
800	Niet-reëel resultaat Als de software bijvoorbeeld in de instelling Reëel staat, dan is $\sqrt{(-1)}$ ongeldig. Om complexe resultaten toe te staan verandert u de modusinstelling "Reëel of complex" in RECHTHOEKIG of POLAIR.
830	Overschrijding
850	Programma niet gevonden Een programmareferentie binnen een ander programma kon niet gevonden worden op het aangegeven pad gedurende de uitvoering.
855	Random-functies zijn niet toegestaan bij het tekenen van grafieken
860	Recurisie te diep
870	Gereserveerde naam of systeemvariabele
900	Argumentfout Mediaan-mediaan-model kon niet worden toegepast op gegevensset.
910	Syntaxfout
920	Tekst niet gevonden
930	Te weinig argumenten In de functie of het commando ontbreken één of meer argumenten.
940	Te veel argumenten De uitdrukking of vergelijking bevat een te groot aantal argumenten en kan niet geëvalueerd worden.
950	Te veel subscripts
955	Te veel ongedefinieerde variabelen
960	Variabele is niet gedefinieerd Er is geen waarde toegewezen aan de variabele. Gebruik een van de volgende commando's: • <code>sto</code> → • <code>:=</code> • Define om waarden aan variabelen toe te kennen.
965	Niet-gelicenseerd OS

Foutcode	Beschrijving
970	Variabele is in gebruik, hierdoor zijn verwijzingen of veranderingen niet toegestaan.
980	Variabele is beschermd
990	Ongeldige variabelenaam Zorg ervoor dat de naam de lengtebeperkingen niet overschrijdt.
1000	Domein van de venstervariabelen
1010	Zoomen
1020	Interne fout
1030	Schending van het beveiligde geheugen
1040	Niet-ondersteunde functie. Voor deze functie is het Computer Algebra System nodig. Probeer TI-Nspire™ CAS.
1045	Niet-ondersteunde operator. Voor deze operator is het Computer Algebra System nodig. Probeer TI-Nspire™ CAS.
1050	Niet-ondersteunde functie. Voor deze operator is het Computer Algebra System nodig. Probeer TI-Nspire™ CAS.
1060	Invoerargument moet numeriek zijn. Alleen invoer met numerieke waarden is toegestaan.
1070	Goniofunctie-argument is te groot voor nauwkeurige verkleining
1080	Niet-ondersteund gebruik van Ans. Deze toepassing ondersteunt Ans niet.
1090	Functie is niet gedefinieerd. Gebruik een van de volgende commando's: • Define • := • sto → om een functie te definiëren.
1100	Niet-reële berekening Als de software bijvoorbeeld in de instelling Reëel staat, dan is $\sqrt{-1}$ ongeldig. Om complexe resultaten toe te staan verandert u de modusinstelling "Reëel of complex" in RECHTHOEKIG of POLAIR.
1110	Ongeldige grenzen
1120	Geen tekenverandering
1130	Argument kan geen lijst of matrix zijn
1140	Argumentfout

Foutcode	Beschrijving
	Het eerste argument moet een polynomiale uitdrukking in het tweede argument zijn. Als het tweede argument wordt weggelaten, probeert de software om een standaardinstelling te selecteren.
1150	Argumentfout De eerste twee argumenten moeten polynomiale uitdrukkingen in het derde argument zijn. Als het derde argument wordt weggelaten, probeert de software om een standaardinstelling te selecteren.
1160	Ongeldige bibliotheekpadnaam Een padnaam moet de vorm <code>xxx\yyy</code> hebben, waarbij: • Het <code>xxx</code>-gedeelte tussen de 1 en 16 tekens kan hebben. • Het <code>yyy</code>-gedeelte 1 tot 15 tekens kan hebben. Zie voor meer informatie de paragraaf Bibliotheek in de documentatie.
1170	Ongeldig gebruik van bibliotheekpadnaam • Er kan geen waarde aan een padnaam worden toegekend met Define, <code>:=</code> of <code>sto</code> →. • Een padnaam kan niet gedeclareerd worden als een Local-variabele of gebruikt worden als een parameter in een functie- of programmadefinitie.
1180	Ongeldige bibliotheekvariabelenaam. Zorg ervoor dat de naam: • geen punt bevat • niet met een onderstrepingsteken begint • niet meer dan 15 tekens heeft Zie voor meer informatie de paragraaf Bibliotheek in de documentatie.
1190	Bibliotheekdocument niet gevonden: • Controleer of de bibliotheek zich in de map MyLib bevindt. • Bibliotheeken vernieuwen. Zie voor meer informatie de paragraaf Bibliotheek in de documentatie.
1200	Bibliotheekvariabele niet gevonden: • Controleer of de bibliotheekvariabele bestaat in de eerste opgave in de bibliotheek. • Zorg ervoor dat de bibliotheekvariabele gedefinieerd is als LibPub of LibPriv. • Bibliotheeken vernieuwen. Zie voor meer informatie de paragraaf Bibliotheek in de documentatie.
1210	Ongeldige naam voor sneltoets bibliotheek.

Foutcode	Beschrijving
	Zorg ervoor dat de naam: • geen punt bevat • niet met een onderstrepingsteken begint • niet meer dan 16 tekens heeft • geen gereserveerde naam is Zie voor meer informatie de paragraaf Bibliotheek in de documentatie.
1220	Domeinfout: TangentLine en normalLine ondersteunen alleen functies met reële waarden.
1230	Domeinfout. Goniometrische conversie-operatoren worden niet ondersteund in de hoekmodi Graden en Decimale graden.
1250	Argumentfout Gebruik een stelsel lineaire vergelijkingen. Voorbeeld van een stelsel van twee lineaire vergelijkingen met variabelen x en y: $3x+7y=5$ $2y-5x=-1$
1260	Argumentfout: Het eerste argument van nfMin of nfMax moet een uitdrukking in 1 variabele zijn. Deze kan geen variabele zonder waarde bevatten anders dan de relevante variabele.
1270	Argumentfout De orde van de afgeleide moet gelijk zijn aan 1 of 2.
1280	Argumentfout Gebruik een polynoom in uitgewerkte vorm met 1 variabele.
1290	Argumentfout Gebruik een polynoom in één variabele.
1300	Argumentfout De coëfficiënten van de polynoom moeten te herleiden zijn tot numerieke waarden.
1310	Domeinfout: Een functie kon niet worden uitgewerkt voor één of meer van de argumenten.
1380	Argumentfout: Geneste oproepen aan domein() functie zijn niet toegestaan.

Waarschuwingcodes en berichten

U kunt de **warnCodes()**-functie gebruiken om de codes van waarschuwingen die gegenereerd zijn door het uitwerken van een uitdrukking op te slaan. Deze tabel geeft een overzicht van de numerieke waarschuwingcodes en de bijbehorende berichten.

Zie voor een voorbeeld van het opslaan van waarschuwingcodes **warnCodes()**, pag. 213.

Waarschuwingcode	Bericht
10000	Bewerking geeft mogelijk ongeldige oplossingen.
10001	Het differentiëren van een vergelijking kan een valse vergelijking opleveren.
10002	Twijfelachtige oplossing
10003	Twijfelachtige nauwkeurigheid
10004	Bewerking veroorzaakt mogelijk verlies van oplossingen.
10005	cSolve specificeert mogelijk meer nulpunten.
10006	Solve specificeert mogelijk meer nulpunten.
10007	Er kunnen meer oplossingen bestaan. Probeer geschikte onder- en bovengrenzen te specificeren en/of doe een schatting. Voorbeelden met solve(): • solve(Vergelijking,Var=Gok) ondergrens<Var<bovengrens • solve(Vergelijking,Var) ondergrens<Var<bovengrens • solve(Vergelijking,Var=Gok)
10008	Het domein van het resultaat is mogelijk kleiner dan het domein van de invoer.
10009	Het domein van het resultaat is mogelijk groter dan het domein van de invoer.
10012	Niet-reële berekening
10013	∞^0 of undef^0 is vervangen door 1
10014	undef^0 is vervangen door 1
10015	1^∞ of 1^undef is vervangen door 1
10016	1^undef is vervangen door 1
10017	Overloop is vervangen door ∞ of $-\infty$
10018	De bewerking vraagt om en retourneert een 64-bits waarde.
10019	Bron uitgeput, vereenvoudiging is mogelijk onvolledig.

Waarschuwingcode	Bericht
10020	Goniofunctie-argument is te groot voor nauwkeurige reductie.
10021	De invoer bevat een ongedefinieerde parameter. Het resultaat is mogelijk niet geldig voor alle mogelijke parameterwaarden.
10022	Het specificeren van de juiste onder- en bovengrenzen kan een oplossing opleveren.
10023	Scalair is vermenigvuldigd met de eenheidsmatrix.
10024	Resultaat is verkregen met behulp van benaderende berekeningen.
10025	Equivalentie kan niet worden geverifieerd in de EXACT-modus.
10026	Beperking kan genegeerd worden. Specificeer een beperking in de vorm "\" 'Variabele WiskundeTestSymbool Constante' of een combinatie van deze vormen, bijvoorbeeld 'x<3 en x>-12'

Algemene informatie

Online Help

education.ti.com/eguide

Selecteer uw land voor meer productinformatie.

Neem contact op met TI Ondersteuning

education.ti.com/ti-cares

Selecteer uw land voor technische en andere ondersteuningsbronnen.

Service- en garantie-informatie

education.ti.com/warranty

Selecteer uw land voor informatie over de duur en de voorwaarden van de garantie of over productservice.

Beperkte garantie. Deze garantie heeft geen invloed op uw wettelijke rechten.

Index

'	
', minutennotatie	244
', prime	245
-	
-, aftrekken[*]	224
!	
!, faculteit	235
"	
", secondennotatie	244
#	
#, indirectie	241
#, indirectie-operator	273
%	
%, percentage	230
&	
&, toevoegen	235
*	
*, vermenigvuldigen	225
.	
., punt aftrekken	228
., puntvermenigvuldiging	229
./, punt delen	229
., punt machtsverheffen	229
., punt optellen	228
:	
:=, toewijzen	249

^	
$\wedge^{-1}$, omgekeerde	247
$\wedge$, macht	227
-	
_, eenheidsaanduiding	245
, beperkende operator	247
+	
+, optellen	224
/	
/, delen[*]	226
=	
=, is gelijk	230
≠	
≠, is niet gelijk aan[*]	231
>	
>, groter dan	233
Π	
Π, product[*]	238
Σ	
Σ(), som[*]	239
ΣInt()	239
ΣPrn()	240
√	
√, wortel[*]	238
∫	
∫, integraal[*]	236

$\leq$		$\rightarrow$	
$\leq$, kleiner dan of gelijk aan	232	$\rightarrow$, opslaan	249
$\geq$		$\Leftrightarrow$	
$\geq$, groter dan of gelijk aan	233	$\Leftrightarrow$, logische dubbele implicatie[*]	234
$\blacktriangleright$		$\textcircled{C}$	
$\blacktriangleright$, converteren naar hoek in decimale graden[Grad]	91	$\textcircled{C}$, commentaar	250
$\blacktriangleright$, eenheden converteren[*]	246	$^{\circ}$	
$\blacktriangleright$ approxFraction()	14	$^{\circ}$, graden/minuten/seconden[*]	244
$\blacktriangleright$ Cilind, weergeven als cilindrische vector[Cilind]	45	$^{\circ}$, gradennotatie[*]	243
$\blacktriangleright$ cos, weergeven in termen van cosinus[cos]	31	0	
$\blacktriangleright$ DD, weergeven als decimale hoek [DD]	48	Ob, binaire indicator	250
$\blacktriangleright$ Decimal, resultaat weergeven als decimaal[Decimaal]	48	Oh, hexadecimale indicator	250
$\blacktriangleright$ DMS, weergeven als graden/minuten/seconden [DMS]	58	1	
$\blacktriangleright$ exp, weerg. in termen v. e[exp]	67	$10^{\wedge}()$, macht van tien	247
$\blacktriangleright$ Grondtal10, weergegeven als decimaal geheel getal [Grondtal10]	19	2	
$\blacktriangleright$ Grondtal16, weergeven als hexadecimaal[Grondtal16]	20	2-stuks stuksgewijs gedefinieerde functie	
$\blacktriangleright$ Grondtal2, weergeven als binair [Grondtal2]	18	template voor	2
$\blacktriangleright$ Polar, weergeven als polaire vector [Polar]	141	A	
$\blacktriangleright$ Rad, converteren naar radialen	151	aantal dagen tussen datums, dbd()	47
$\blacktriangleright$ Rect, weergeven als rechthoekige vector	155	abs(), absolute waarde	8
$\blacktriangleright$ sin, weergeven in termen van sinus [sin]	177	absolute waarde	
$\blacktriangleright$ Sphere, weergeven als bolvormige vector[Sphere]	187	template voor	4
$\Rightarrow$		afgeleide of n-de afgeleide	
$\Rightarrow$, logische implicatie[*]	234, 269	template voor	6
		afgeleiden	
		eerste afgeleide, d()	235
		numerieke afgeleide, nDeriv()	129
		numerieke afgeleide, nDerivative()	128
		aflossingstabel, amortTbl()	8, 17
		afmeting, dim()	55
		afronden, round()	165
		afsluiten, Exit	67

aftrekken, -	224	bibliotheek	
als, If	92	snelkoppelingen naar objecten	
amortTbl(), aflossingstabel	8, 17	creëren	102
and, Booleaanse operator	9	binair	
anders, Else	92	indicator, Ob	250
angle(), hoek	10	weergeven, ▶Grondtal2	18
ANOVA, eenwegs variantieanalyse	11	binnen string, inString()	95
ANOVA2way, tweewegs		binomCdf()	21, 98
variantieanalyse	11	binomPdf()	21
Ans, laatste antwoord	13	bodem, floor()	75
antwoord (laatste), Ans	13	bolvormige vectorweergave,	
approx(), benaderend	14	▶Sphere	187
approxRational()	14	booglengte, arcLen()	15
arccos()	15	Booleaanse operatoren	
arcosh()	15	⇒	234, 269
arccosinus, $\cos^{-1}()$	33	⇔	234
arccot()	15	and	9
arccoth()	15	niet	132
arccsc()	15	niet en	126
arccsch()	15	noch	130
arcLen(), booglengte	15	of	137
arcsec()	15	xof	214
arcsech()	16	breuken	
arcsin()	16	propFrac	147
arcsinh()	16	template voor	1
arcsinus, $\sin^{-1}()$	179		
arctan()	16	C	
arctangens, $\tan^{-1}()$	195	Cdf()	73
arctanh()	16	ceiling(), plafond	21
argumenten in TVM-functies	208	centralDiff()	22
augment(),		cFactor(), complexe factor	23
uitbreiden/aaneenvoegen	16	char(), tekenreeks	24
avgRC(), gemiddelde		charPoly()	24
veranderingssnelheid	17	χ^2 2way	25
		χ^2 Cdf()	25
B		χ^2 GOF	25
beëindigen		χ^2 Pdf()	26
terwijl, EndWhile	214	cilindrische vectorweergave, ▶Cylind	45
benaderend, approx()	14	ClearAZ	26
bepaalde integraal		ClrErr, fout wissen	27
template voor	6	colAugment	27
beperkende operator " "	247	colDim(), matrixkolomafmeting	27
beperkende operator, volgorde van		colNorm(), matrixkolomnorm	27
uitwerking	272	combinaties, nCr()	127

comDenom(), gemeenschappelijke noemer	28	csch ⁻¹ (), inverse cosecans hyperbolicus	40
commentaar, ©	250	csch(), cosecans hyperbolicus	40
completeSquare(), complete square	29	cSolve(), complexe oplossen	40
complex		CubicReg, derdegraads regressie ...	43
geconjugeerde, conj()	30	cumulatieve som, cumulativeSum()	44
nulpunten, cZeros()	45	cumulativeSum(), cumulatieve som	44
ontbinden, cFactor()	23	cycle, Cycle	45
oplossen, cSolve()	40	Cycle, cycle	45
conj(), complex geconjugeerde	30	cZeros(), complexe nulpunten	45
constante			
in solve()	183		
constanten		D	
in deSolve()	52	d(), eerste afgeleide	235
in solve()	185	dagen tussen datums, dbd()	47
opnemen in cSolve()	42	dbd(), dagen tussen datums	47
opnemen in cZeros()	46	de normaal, normalLine()	132
sneltoetsen voor	269	decimaal	
constructMat(), matrix construeren	30	weergave van geheel getal, 4Grondtal10	19
converteren		decimale	
►Rad	151	hoekweergave, ►DD	48
4Grad	91	Define	49
eenheden	246	Define LibPriv	50
correlatiematrix, corrMat()	31	Define LibPub	50
corrMat(), correlatiematrix	31	define, Define	49
cos ⁻¹ , arccosinus	33	Define, define	49
cos(), cosinus	32	defining	
cosh ⁻¹ (), arccosinus hyperbolicus ..	35	private function or program ...	50
cosh(), cosinus hyperbolicus	34	public function or program	50
cosinus		delen door geheel getal, intDiv() ...	96
uitdrukking weergeven in		delen, /	226
termen van	31	deltaList()	51
cosinus, cos()	32	deltaTmpCnv()	51
cot ⁻¹ (), arccotangens	36	DelVar, variabele wissen	51
cot(), cotangens	35	delVoid(), lege elementen	
cotangens, cot()	35	verwijderen	52
coth ⁻¹ (), arccotangens hyperbolicus	37	derdegraads regressie, CubicReg ...	43
coth(), cotangens hyperbolicus	36	derivative()	52
count(), items tellen in een lijst	37	deSolve(), oplossing	52
countif(), items in een lijst		det(), matrixdeterminant	54
voorwaardelijk tellen	37	diag(), matrixdiagonaal	55
cPolyRoots()	38	dim(), afmeting	55
crossP(), uitwendig product	39	Disp, gegevens weergeven	56, 168
csc ⁻¹ (), inverse cosecans	39	DispAt	56
csc(), cosecans	39		

domein(), domeinfunctie	59	euler(), Euler function	64
domeinfunctie, domein()	59	exact(), exact	67
dominant term, dominantTerm() ..	59	exact, exact()	67
dominantTerm(), dominant term ..	59	Exit, afsluiten	67
dot		exp(), e tot een macht	68
product, dotP()	61	exp►list(), uitdrukking naar lijst	68
dotP(), inwendig product	61	expand(), uitbreiden	69
E		exponent, E	242
e-macht		exponenten	
template voor	2	template voor	1
e tot een macht, e^()	61, 68	exponentiële regressie, ExpReg	70
E, exponent	242	expr(), string naar uitdrukking	70, 113
e, uitdr. weergeven in termen van ..	67	ExpReg, exponentiële regressie	70
e^(), e tot een macht	61	F	
echte breuk, propFrac	147	F-toets met 2 steekproeven	80
eenheden		factor(), ontbinden	71
converteren	246	faculteit, !	235
eenheid(), eenheidsmatrix	92	Fill, matrix vullen	74
eenheidsmatrix, eenheid()	92	financiële functies, tvml()	206
eenheidsvector, unitV()	210	financiële functies, tvml()	206
eerste afgeleide		financiële functies, tvmlN()	207
template voor	5	financiële functies, tvmlPmt()	207
eff), nominaal naar effectief		financiële functies, tvmlPV()	207
percentage converteren ..	62	FiveNumSummary	74
effectief percentage, eff()	62	floor(), bodem	75
eigenvector, eigVc()	62	fMax(), functiemaximum	75
eigenwaarde, eigVl()	63	fMin(), functieminimum	76
eigVc(), eigenvector	62	For	77
eigVl(), eigenwaarde	63	for, For	77
einde		For, for	77
als, EndIf	92	format(), opmaakstring	77
einde als, EndIf	92	fout overbrengen, PassErr	139
else if, Elseif	63	fouten en het oplossen van	
Elseif, else if	63	problemen	
end		fout overbrengen, PassErr	139
for, EndFor	77	fout wissen, ClrErr	27
functie, EndFunc	81	fpart(), functiedeel	78
loop, EndLoop	116	freqTable()	79
end loop, EndLoop	116	frequency()	79
EndTry, proberen beëindigen	203	Frobenius-norm, norm()	131
EndWhile, terwijl beëindigen	214	Func, functie	81
EOS (Equation Operating System) ..	272	Func, functie programmeren	81
Equation Operating System (EOS) ..	272		

Input, invoer	95	lengte van string	55
instellen		LibPriv	50
modus, setMode()	172	LibPub	50
instellingen, huidige ophalen	88	libShortcut(), snelkoppelingen naar	
inString(), binnen string	95	bibliotheekobjecten	
int(), geheel getal	95	creëren	102
intDiv(), delen door geheel getal ...	96	lijst naar matrix, list►mat()	109
integraal, ∫	236	lijst, items tellen in	37
interpolate(), interpoleren	96	lijst, items voorwaardelijk tellen in ..	37
inverse cumulatieve normale		lijsten	
verdeling (invNorm())	98	aflopend sorteren, SortD	186
inverse, $\wedge^{-1}$	247	cumulatieve som,	
invF()	97	cumulativeSum()	44
invNorm(), inverse cumulatieve		inwendig product, dotP()	61
normale verdeling)	98	lege elementen in	267
invoer, Input	95	lijst naar matrix, list►mat()	109
invt()	99	matrix naar lijst, mat►list()	117
Inv χ^2 ()	97	maximum, max()	118
iPart(), geheel deel	99	mid-string, mid()	120
irr(), interne rentabiliteit		minimum, min()	121
interne rentabiliteit, irr()	99	nieuw, newList()	128
is gelijk, =	230	oplopend sorteren, SortA	186
is niet gelijk aan, ≠	231	product, product()	146
isPrime(), priemtoets	100	som, sum()	192
isVoid(), testen op lege elementen ..	100	uitbreiden/aaneenvoegen,	
items in een lijst voorwaardelijk		augment()	16
tellen, countif()	37	uitdrukking naar lijst, exp►list() ..	68
items tellen in een lijst, count()	37	uitwendig product, crossP() ...	39
		verschil, Δlist()	109
		verschillen in een lijst, @list() ..	109
K		limiet	
kansdichtheid, normPdf()	132	lim()	103
kleiner dan of gelijk aan, {	232	limit()	103
kleinste gemene veelvoud, lcm	101	template voor	6
kwadratische regressie, QuadReg ...	149	limit() of lim(), limiet	103
		lineaire regressie, LinRegAx	105
L		lineaire regressie, LinRegBx	104, 106
label, Lbl	101	links, left()	101
Lbl, label	101	LinRegBx, lineaire regressie	104
lcm, kleinste gemene veelvoud	101	LinRegMx, lineaire regressie	105
left(), links	101	LinRegtIntervals, lineaire regressie ..	106
lege elementen	267	LinRegtTest	107
lege elementen, testen op	100	linSolve()	108
lege elementen, verwijderen	52	Δlist(), lijstverschil	109

list►mat(), lijst naar matrix	109	matrix naar lijst, mat►list()	117
ln(), natuurlijk logaritme	110	maximum, max()	118
LnReg, logaritmische regressie	110	minimum, min()	121
Local, lokale variabele	112	nieuw, newMat()	128
Lock, variabele of variabelegroep vergrendelen	112	product, product()	146
Log		punt aftrekken, .-	228
template voor	2	punt delen, ./	229
logaritmen	110	punt machtsverheffen, .^	229
logaritmische regressie, LnReg	110	punt optellen, .+	228
logische dubbele implicatie, ⇔	234	punt vermenigvuldigen, .*	229
logische implicatie, ⇒	234, 269	QR-ontbinding, QR	148
Logistic, logistische regressie	114	rij-afmeting, rowDim()	166
LogisticD, logistische regressie	115	rij-echelonvorm, ref()	156
logistische regressie, Logistic	114	rij-omwisseling, rowSwap()	166
logistische regressie, LogisticD	115	rij-optelling, rowAdd()	165
lokaal, Local	112	rijbewerking, mRow()	123
lokale variabele, Local	112	rijnorm, rowNorm()	166
loop, Loop	116	rijvermenigvuldiging en - optelling, mRowAdd()	123
Loop, loop	116	som, sum()	192
LU, matrix beneden-boven- decompositie	117	submatrix, subMat()	192, 194
		transponeren, T	194
		uitbreiden/aaneenvoegen, augment()	16
		vullen, Fill	74
		willekeurig, randMat()	153
M		matrix (1 × 2)	
macht van tien, 10^()	247	template voor	4
macht, ^	227	matrix (2 × 1)	
machtsregressie, PowerReg	144, 158, 160, 199	template voor	4
mat►list(), matrix naar lijst	117	matrix (2 × 2)	
matrices		template voor	4
afmeting, dim()	55	matrix (m × n)	
beneden-boven-decompositie, LU	117	template voor	4
cumulatieve som, cumulativeSum()	44	matrix construeren, constructMat()	30
determinant, det()	54	matrix naar lijst, mat►list()	117
diagonaal, diag()	55	max(), maximum	118
eenheid, eenheid()	92	maximum, max()	118
eigenvector, eigVc()	62	mean(), gemiddelde	118
eigenwaarde, eigVl()	63	mediaan-mediaan-lijnregressie, MedMed	119
gereduceerde rij-echelonvorm, rref()	166	mediaan, median()	119
kolomafmeting, colDim()	27	median(), mediaan	119
kolomnorm, colNorm()	27	MedMed, mediaan-mediaan- lijnregressie	119
lijst naar matrix, list►mat()	109		

Meervoudige lineaire regressie t-toets	125	noch, Booleaanse operator	130
met, 	247	noemer	28
mid-string, mid()	120	nom), effectief naar nominaal percentage converteren ..	130
mid(), mid-string	120	nominaal percentage, nom()	130
min(), minimum	121	norm(), Frobenius-norm	131
minimum, min()	121	normale verdelingskans, normCdf()	132
minutennotatie,	244	normalLine()	132
mirr(), gewijzigde interne rentabiliteit	122	normCdf()	132
mod(), modulo	122	normPdf()	132
modi		notatie in decimale graden, g	242
instelling, setModel()	172	nPr(), permutaties	133
modulo, mod()	122	npv(), netto contante waarde	134
modusinstellingen, getModel()	88	nSolve(), numerieke oplossing	134
mRow(), matrixrij-bewerking	123	nulpunten, zeroes()	215
mRowAdd(), matrixrij-vermenigvuldiging en -optelling	123	numeriek	
MultReg	123	afgeleide, nDeriv()	129
MultRegIntervals()	124	afgeleide, nDerivative()	128
MultRegTests()	125	integraal, nInt()	129
		oplossing, nSolve()	134
		O	
N		objecten	
n-de wortel		snellkoppelingen naar bibliotheek creëren ...	102
template voor	1	of (Booleaans), of	137
N-stuks stuksgewijs gedefinieerde functie		of Booleaanse operator	137
template voor	3	omgekeerde, $\wedge^{-1}$	247
natuurlijk logaritme, ln()	110	onbepaalde integraal	
nCr(), combinaties	127	template voor	6
nDerivative(), numerieke afgeleide ..	128	onderstrepingssteken, _	245
negatie, negatieve getallen invoeren	273	OneVar, statistieken voor één variabele	135
netto contante waarde, npv()	134	ontbinden, factor()	71
newList(), nieuwe lijst	128	operatoren	
newMat(), nieuwe matrix	128	volgorde van uitwerking	272
nfMax(), numeriek functiemaximum	129	ophalen/terugsturen	
nfMin(), numeriek functieminiimum	129	getal, getNum()	89
niet en (nen), Booleaanse operator ..	126	noemer, getDenom()	84
niet, Booleaanse operator	132	variabele-informatie, getVarInfo()	87, 90
nieuw		oplossen, solve()	182
lijst, newList()	128	oplossing, deSolve()	52
matrix, newMat()	128	opmaakstring, format()	77
nInt(), numerieke integraal	129		

opslaan			
symbool, &	249		
optellen, +	224		
ord(), numerieke tekencode	138		
P			
P•Rx(), rechthoekige x-coördinaat ..	138		
P•Ry(), rechthoekig y-coördinaat ..	139		
PassErr, fout overbrengen	139		
Pdf()	78		
percentage, %	230		
permutaties, nPr()	133		
piecewise()	140		
plafond, ceiling()	21-22, 38		
poissCdf()	140		
poissPdf()	140		
polair			
vectorweergave, ►Polar	141		
polyCoeff()	141		
polyDegree()	142		
polyEval(), veelterm uitwerken	143		
polyGcd()	143-144		
PolyRoots()	144		
pool			
coördinaat, R►Pr()	151		
coördinaat, R►Pθ()	151		
PowerReg, machtsregressie	144		
Prgm, programma definiëren	145		
priemgetaltoets, isPrime()	100		
prime,	245		
proberen, Try	203		
prodSeq()	146		
product (P)			
template voor	5		
product(), product	146		
product, product()	146		
product, Π()	238		
programma's en programmeren			
I/O-scherm weergeven, Disp ...	168		
programma's en programmeren			
fout wissen, ClrErr	27		
I/O-scherm weergeven, Disp ...	56		
proberen, Try	203		
programmeren			
fout overbrengen, PassErr	139		
gegevens weergeven, Disp	56, 168		
programma definiëren, Prgm ..	145		
programs			
defining private library	50		
defining public library	50		
propFrac, echte breuk	147		
punt			
aftrekken, .N	228		
delen, .P	229		
macht, .^	229		
optellen, .+	228		
vermenigvuldigen, .*	229		
Q			
QR-ontbinding, QR	148		
QR, QR-ontbinding	148		
QuadReg, kwadratische regressie ...	149		
QuartReg, vierdegraads regressie ...	150		
R			
R, radialen	242		
R►Pr(), poolcoördinaat	151		
R►Pθ(), poolcoördinaat	151		
raaklijn, tangentLine()	196		
radialen, R	242		
rand(), willekeurig getal	152		
randBin, willekeurig getal	152		
randInt(), willekeurig geheel getal ..	152		
randMat(), willekeurige matrix	153		
randNorm(), willekeurige norm	153		
randPoly(), willekeurige veelterm ..	154		
randSamp()	154		
RandSeed, willekeurig getal-seed ...	154		
real(), reëel	155		
rechthoekig y-coördinaat, P►Ry() ..	139		
rechthoekige vectorweergave, ►Rect	155		
rechthoekige x-coördinaat, P►Rx() ..	138		
rechts, right()	96, 161-162		
reëel, real()	155		
reeksen			
gebruiken om variabelenamen	273		

te creëren	24
tekenreeks, char()	24
ref(), rij-echelonvorm	156
RefreshProbeVars	157
regressies	
derdegraads, CubicReg	43
exponentiële, ExpReg	70
kwadratisch, QuadReg	149
lineaire regressie, LinRegAx	105
lineaire regressie, LinRegBx	104, 106
logaritmisch, LnReg	110
Logistic	114
logistisch, Logistic	115
machtsregressie,	
PowerReg .	144, 158, 160, 199
mediaan-mediaan-lijn, MedMed	119
MultReg	123
sinusoïde, SinReg	181
vierdegraads, QuartReg	150
remain(), rest	158
rest, remain()	158
resultaat	
weergave in termen van e	67
weergeven in termen van	
cosinus	31
weergeven in termen van sinus .	177
resultaatwaarden, statistieken	189
resultaten voor twee variabelen,	
TwoVar	208
resultaten, statistieken	188
Return, terugkeren	161
right(), rechts	161
right, right()	29, 64, 213
rij-echelonvorm, ref()	156
rk23(), Runge-Kutta-functie	162
rotate(), roteren	163
roteren, rotate()	163
round(), afronden	165
rowAdd(), matrixrij-optelling	165
rowDim(), matrixrij-afmeting	166
rowNorm(), matrixrij-norm	166
rowSwap(), matrixrij-omwisseling ..	166
rref(), gereduceerde rij-	
echelonvorm	166

S

sec ⁻¹ (), inverse secans	167
sec(), secans	167
sech ⁻¹ (), inverse secans hyperbolicus	168
sech(), secans hyperbolicus	168
secondennotatie, "	244
seq(), getallenrij	169
seqGen()	169
seqn()	170
sequence, seq()	169-170
series(), series	171
series, series()	171
setMode(), instellingsmodus	172
shift(), verschuiven	174
sign(), teken	176
simult(), simultane vergelijkingen ..	176
simultane vergelijkingen, simult() ..	176
sin ⁻¹ (), arcsinus	179
sin(), sinus	178
sinh ⁻¹ (), arcsinus hyperbolicus	180
sinh(), sinus hyperbolicus	179
SinReg, sinusoïde regressie	181
sinus	
uitdrukking weergeven in	
termen van	177
sinus, sin()	178
sinusoïde regressie, SinReg	181
snelkoppelingen, toetsenbord	269
sneltoetsen	269
solve(), oplossen	182
som (G)	
template voor	5
som van hoofdsombetalen	240
som van rentebetalen	239
som, sum()	192
som, Σ()	239
SortA, oplopend sorteren	186
SortD, aflopend sorteren	186
sorteren	
aflopend, SortD	186
oplopend, SortA	186
sqr(), wortel	187
standaarddeviatie, stdDev() ..	189-190, 211

eerste afgeleide	5	Try, foutbehandelingscommando ..	203
exponent	1	Try, proberen	203
limiet	6	tTest, t-toets	204
Log	2	tTest_2Samp, t-toets met twee steekproeven	205
matrix (1 × 2)	4	TVM-argumenten	208
matrix (2 × 1)	4	tvmFV()	206
matrix (2 × 2)	4	tvmI()	206
matrix (m × n)	4	tvmN()	207
n-de wortel	1	tvmPmt()	207
N-stuks stuksgewijs gedefinieerde functie ..	3	tvmPV()	207
onbepaalde integraal	6	tweede afgeleide template voor	6
product (P)	5	TwoVar, resultaten voor twee variabelen	208
som (G)	5		
stelsel van 2 vergelijkingen	3		
stelsel van vergelijkingen (N vergelijkingen)	3	U	
tweede afgeleide	6	uitbreiden, expand()	69
wortel	1	uitbreiden/aaneenvoegen, augment()	16
terugkeren, Return	161	uitdrukkingen string naar uitdrukking, expr() ..	70, 113
terwijl beëindigen, EndWhile	214	uitdrukking naar lijst, exp4lijst() ..	68
terwijl, While	214	uitsluiting met operator "!"	247
Test_2S, F-toets met 2 steekproeven ..	80	uitwendig product, crossP()	39
testen op lege elementen, isVoid() ..	100	uitwerking, volgorde van	272
tExpand(), goniometrische uitwerking	198	unitV(), eenheidsvector	210
tijdwaarde van geld, aantal betalingen	207	unLock, variabele of variabelegroep ontgrendelen	210
tijdwaarde van geld, betalingsbedrag ..	207	user-defined functions	49
tijdwaarde van geld, contante waarde	207	user-defined functions and programs	50
tijdwaarde van geld, Rente	206		
tijdwaarde van geld, toekomstige waarde	206	V	
TInterval, t- betrouwbaarheidsinterval ..	200	variabele naam van een tekenreeks creëren	273
tInterval_2Samp, t- betrouwbaarheidsinterval met 2 steekproeven	200	variabele of functie kopiëren, CopyVar	30
ΔtmpCnv() [tmpCnv]	202	variabelen alle variabelen bestaande uit één letter wissen	26
tmpCnv()	201-202	lokaal, Local	112
toevoegen, &	235	wissen, DelVar	51
tPdf(), studentt-kansdichtheid	203		
trace()	203		
transponeren, T	194		

variabelen en functies		W	
kopiëren	30	waarschuwingscodes en berichten .	285
variabelen en variabelegroepen		Wait-opdracht	212
ontgrendelen	210	wanneer, when()	213
variabelen en variabelegroepen		warnCodes(), Warning codes	213
vergrendelen	112	weergeven als	
variabelen, vergrendelen en		binair, ►Grondtal2	18
ontgrendelen	88, 112, 210	bolvormige vector, ►Sphere ...	187
variantie, variance()	211	cilindrische vector, ►Cylind ...	45
varPop()	211	decimaal geheel getal,	
varSamp(), steekproefvariantie	211	►Grondtal10	19
vectoren		decimale hoek, ►DD	48
cilindrische vectorweergave,		graden/minuten/seconden,	
4Cilind	45	►DMS	58
eenheid, unitV()	210	hexadecimaal, ►Grondtal16	20
inwendig product, dotP()	61	polaire vector, ►Polar	141
uitwendig product, crossP() ...	39	rechthoekige vector, ►Rect	155
veelterm uitwerken, polyEval()	143	when(), wanneer	213
veeltermen		While, terwijl	214
uitwerken, polyEval()	143	willekeurig	
willekeurig, randPoly()	154	getal-seed, RandSeed	154
verdelingsfuncties		matrix, randMat()	153
binomCdf()	21, 98	norm, randNorm()	153
binomPdf()	21	veelterm, randPoly()	154
invNorm()	98	willekeurige steekproef	154
invt()	99	Wiss.	255
Inv χ^2 ()	97	wissen	
normCdf()	132	lege elementen uit een lijst	52
normPdf()	132	variabele, DelVar	51
poissCdf()	140	Wissen	
poissPdf()	140	fout, ClrErr	27
tCdf()	197	wortel	
tPdf()	203	template voor	1
χ^2 2way()	25	wortel, $\sqrt{\quad}$	187, 238
χ^2 Cdf()	25		
χ^2 GOF()	25	X	
χ^2 Pdf()	26	x ² , kwadraat	228
vermenigvuldigen, *	225	XNOR	234
verschuiven, shift()	174	xof, Booleaans exclusief of	214
verwijderen			
lege elementen uit een lijst	52	Z	
Verzoek	158	zeroes(), nulpunten	215
VerzoekStr	160	zInterval, z-	218
vierdegraads regressie (QuartReg) ..	150		
vullen	259-260		

betrouwbaarheidsinterval .	
zInterval_1Prop, z-	
betrouwbaarheidsinterval	
met één proportie	218
zInterval_2Prop, z-	
betrouwbaarheidsinterval	
met twee proporties	219
zInterval_2Samp, z-	
betrouwbaarheidsinterval	
met twee steekproeven ...	219
zTest	220
zTest_1Prop, z-toets met één	
proportie	221
zTest_2Prop, z-toets voor twee	
proporties	222
zTest_2Samp, z-toets met twee	
steekproeven	222