

TI-Nspire™ CX

Referentiehandleiding

Belangrijke informatie

Tenzij expliciet anders vermeld in de bij een programma meegeleverde licentie, geeft Texas Instruments geen garantie, expliciet dan wel impliciet, met inbegrip van, maar niet beperkt tot willekeurig welke impliciete garanties van verhandelbaarheid en geschiktheid voor een bepaald doel met betrekking tot welke programma's of boekmaterialen dan ook, en stelt dergelijke materialen uitsluitend beschikbaar op een "as-is" basis. Texas Instruments is in geen enkel geval aansprakelijk voor speciale, indirecte, incidentele of voortvloeiende schade in verband met of voortkomend uit de aankoop of het gebruik van deze materialen, en de enige en uitsluitende aansprakelijkheid van Texas Instruments, ongeacht de actievorm, is niet hoger dan het in de licentie voor het programma vermelde bedrag. Voorts is Texas Instruments niet aansprakelijk voor welke eis van welke aard dan ook tegen het gebruik van deze materialen door enige andere partij.

© 2024 Texas Instruments Incorporated

Feitelijke producten kunnen enigszins afwijken van de getoonde afbeeldingen.

Inhoudsopgave

Uitdrukkingstemplates	1
Alfabetische lijst	7
A	7
B	16
C	20
D	37
E	47
F	55
G	62
I	73
L	81
M	96
N	105
O	115
P	118
Q	125
R	128
S	145
T	166
U	179
V	179
W	180
X	183
Z	184
Symbolen	191
TI-Nspire™ CX II - Tekenopdrachten	216
Programmeren van grafische weergaven	216
Grafisch scherm	216
Standaardweergave en instellingen	217
Foutmeldingen op het grafische scherm	218
Ongeldige opdrachten in de grafische modus	218
C	220
D	221
F	224
G	226
P	227
S	229
U	231

Lege elementen	232
Snelkoppelingen voor het invoeren van wiskundige uitdrukkingen	234
EOS (Equation Operating System)-hiërarchie	236
TI-Nspire CX II - Functies van de TI-Basic-programmering	238
Auto-inspringen in de programmeereditor	238
Verbeterde foutmeldingen voor TI-Basic	238
Constanten en waarden	241
Foutcodes en meldingen	242
Waarschuwingcodes en berichten	251
Algemene informatie	253
Index	254

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. 193.

Voorbeeld:

12

8·2

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 (**►**).

Voorbeeld:

2³

8

Opmerking: zie ook ^ (macht), pag. 194.

Worteltemplate

[ctrl][x²]-toetsen


 **Opmerking:** zie ook √() (wortel), pag. 204.

Voorbeeld:

√4

√{9,16,4}

2
{3,4,2}

Uitdrukkingstemplates 1

N-de wortel-template

ctrl **^** -toetsen

$\sqrt{\quad}$

Opmerking: zie ook **wortel()**, pag. 141.

Voorbeeld:

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

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

e-macht-template

e^x -toetsen

$e^{\quad}$

Het getal e verheven tot een macht

Opmerking: zie ook **e^()**, pag. 47.

Voorbeeld:

$$e^1 \quad 2.71828182846$$

Log-template

ctrl **10^x** -toets

$\log_{\quad}(\quad)$

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

Opmerking: zie ook **log()**, pag. 92.

Voorbeeld:

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

Template voor stuksgewijs gedefinieerde functies (2-stuks)

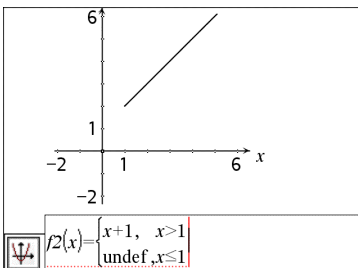
Catalogus > **log²**

$\left\{ \begin{array}{l} \square, \square \\ \square, \square \end{array} \right.$

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. 120.

Voorbeeld:



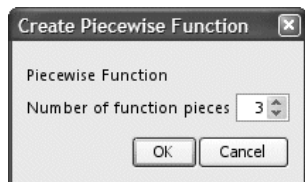
Template voor stuksgewijs gedefinieerde functies (N-stuks)

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. 120.

Stelsel van 2 vergelijkingen-template

Catalogus > 



Creëert een stelsel van twee lineaire 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. 166.

Voorbeeld:

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

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

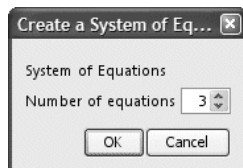
Stelsel van N vergelijkingen-template

Catalogus > 

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

Voorbeeld:

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



Opmerking: zie ook `system()`, pag. 166.

Absolute waarde-template

Catalogus >



$| \square |$ **Opmerking:** zie ook `abs()`, pag. 7.

Voorbeeld:

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

dd°mm'ss.ss'' template

Catalogus >



$\square^\circ \square' \square''$

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.

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

Matrixtemplate (2 x 2)

Catalogus >



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

Voorbeeld:

Creëert een 2 x 2 matrix.

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

Matrixtemplate (1 x 2)

Catalogus >



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

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 >



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

Voorbeeld:

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

Matrixtemplate (m x n)

Catalogus >

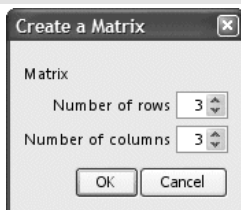


De template verschijnt nadat u het aantal rijen en kolommen heeft ingevoerd.

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. 205.

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. 204.

Eerste afgeleide-template

Catalogus >



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

Voorbeeld:

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

Eerste afgeleide-template

Catalogus > 

De eerste afgeleide-template kan worden gebruikt om de eerste afgeleide in een punt numeriek te berekenen, met behulp van automatische differentiatiemethoden.

Opmerking: zie ook `d()` (afgeleide), pag. 202.

Tweede afgeleide-template

Catalogus > 

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

De tweede afgeleide-template kan worden gebruikt om de tweede afgeleide in een punt numeriek te berekenen, met behulp van automatische differentiatiemethoden.

Opmerking: zie ook `d()` (afgeleide), pag. 202.

Voorbeeld:

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

Bepaalde integraal-template

Catalogus > 

$$\int_a^b \square \, d\square$$

De bepaalde integraal-template kan worden gebruikt om de bepaalde integraal numeriek te berekenen, met dezelfde methode als `nInt()`.

Opmerking: zie ook `nInt()`, pag. 109.

Voorbeeld:

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

Alfabetische lijst

Elementen waarvan de namen niet alfabetische zijn (zoals +, ! en >) staan aan het eind van dit hoofdstuk, pag. 191. 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(*WaardeI*) $\Rightarrow$ *waarde*

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

$\{1.5708, 1.0472\}$

abs(*LijstI*) $\Rightarrow$ *lijst*

$|2-3 \cdot i|$

3.60555

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

amortTbl()

Catalogus >

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

amortTbl(12,60,10,5000,,,12,12)

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

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. 176.

- 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 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 **ΣInt()** en **ΣPrn()**, pag. 206, en **bal()**, pag. 16.

and

BooleaanseUitdr1 **and**
BooleaanseUitdr2 ⇒ *Booleaanse*
uitdrukking

BooleaanseLijst1 **and**
BooleaanseLijst2 ⇒ *Booleaanse lijst*

BooleaanseMatrix1 **and**
BooleaanseMatrix2 ⇒ *Booleaanse*
matrix

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

Geheel getal1 **and** *Geheel*
getal2 ⇒ *geheel getal*

In de Hex-grondtalmodus:

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

Belangrijk: nul, niet de letter O.

In de Bin-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 geretourneerde 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.

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

In de Dec-grondtalmodus:

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(*WaardeI*) ⇒ *waarde*

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

In de hoekmodus Graden:

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

In de hoekmodus Decimale graden:

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

In de hoekmodus Radialen:

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

angle(*LijstI*) ⇒ *lijst*

angle(MatrixI)⇒matrix

Geeft een lijst of matrix met de hoeken van de elementen in *LijstI* of *MatrixI*, waarbij elk element geïnterpreteerd wordt als een complex getal dat een punt in een rechthoekig tweedimensionaal assenstelsel voorstelt.

ANOVA

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. 160).

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

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. 160).

NivRij=0 voor blok

NivRij=2,3,...,*Len*-1, voor tweeweg, waarbij
Len=*lengte(Lijst1)*=*lengte(Lijst2)* = ... = *lengte(Lijst10)* en *Len / NivRij* $\in \mathbb{N}\{2,3,\dots\}$

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

Uitvoervariabele	Beschrijving
stat.dfCol	Vrijheidsgraden van de kolomfactor
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(*Waarde1*)⇒*getal*

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 **ctrl** **enter** drukken.

approx(*Lijst1*)⇒*lijst*

approx(*Matrix1*)⇒*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}

$\text{approx}\left(\left[\sqrt{2}, \sqrt{3}\right]\right)$ [1.41421 1.73205]

$\text{approx}\left(\left[\frac{1}{3}, \frac{1}{9}\right]\right)$ [0.333333 0.111111]

$\text{approx}\{\sin(\pi), \cos(\pi)\}$ {0,-1}

$\text{approx}\left(\left[\sqrt{2}, \sqrt{3}\right]\right)$ [1.41421 1.73205]

►approxFraction()

Catalogus >

Waarde ►**approxFraction**

(*[Tol]*)⇒*waarde*

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 ► $\text{approxFraction}(5.E-14)$
 $\frac{5}{6}$

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

approxRational()Catalogus > **approxRational**(*Waarde*[,
Tol]) $\Rightarrow$ *waarde*

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

approxRational(*Lijst*[, *tol*]) $\Rightarrow$ *lijst*

$$\text{approxRational}(\{0.2, 0.33, 4.125\}, 5 \cdot 10^{-14})$$

approxRational(*Matrix*[, *Tol*]) $\Rightarrow$ *matrix*

$$\left\{ \frac{1}{5}, \frac{33}{100}, \frac{33}{8} \right\}$$

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

arccos()Zie $\cos^{-1}()$, pag. 28.**arccosh()**Zie $\cosh^{-1}()$, pag. 29.**arccot()**Zie $\cot^{-1}()$, pag. 31.**arcoth()**Zie $\coth^{-1}()$, pag. 31.**arccsc()**Zie $\csc^{-1}()$, pag. 34.**arccsch()**Zie $\operatorname{csch}^{-1}()$, pag. 35.**arcsec()**Zie $\sec^{-1}()$, pag. 145.**arcsech()**Zie $\operatorname{sech}^{-1}()$, pag. 146.

arcsin()Zie $\sin^{-1}()$, pag. 155.**arcsinh()**Zie $\sinh^{-1}()$, pag. 156.**arctan()**Zie $\tan^{-1}()$, pag. 167.**arctanh()**Zie $\tanh^{-1}()$, pag. 168.**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* toegevoegd 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.

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

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

avgRC()Catalogus > **avgRC(Uitdr1, Var [=Waarde] [, Stap])⇒uitdrukking****avgRC(Uitdr1, Var [=Waarde] [, Lijst1])⇒lijst****avgRC(Lijst1, Var [=Waarde] [, Stap])⇒lijst****avgRC(Matrix1, Var [=Waarde] [, Stap])⇒matrix**

$x:=2$	2
$\text{avgRC}(x^2 - x + 2, x)$	3.001
$\text{avgRC}(x^2 - x + 2, x, 1)$	3.1
$\text{avgRC}(x^2 - x + 2, x, 3)$	6

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()

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

bal(*NPmt*,*amortTable*)⇒*waarde*

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. 176.

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. 176.

- 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
standaardwaarde *FV=0* gebruikt.

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

- 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. 7.

Opmerking: zie ook $\Sigma Int()$ en $\Sigma Prn()$, pag. 206.

►Base2 ►Grondtal2)

Geheel getal ►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 getal* 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

–2⁶³ 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.

2⁶³ wordt –2⁶³ en wordt weergegeven
als 0h8000000000000000 in de
hexadecimale modus 0b100...000 (63
nullen) in de binaire modus

2⁶⁴ wordt 0 en wordt weergegeven als
0h0 in de hexadecimale modus 0b0 in
de binaire modus

–2⁶³ – 1 wordt 2⁶³ – 1 en wordt
weergegeven als 0h7FFFFFFFFFFFFFFF in
de hexadecimale modus 0b111...111
(64 enen) in de binaire modus

►Base10 (►Grondtal10)

Geheel getal ►Base10 ⇒ *geheel getal*

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

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

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 *binair* *Getal*

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.

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. 17.

256►Base16	0h100
0b111100001111►Base16	0hFOF

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

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(*WaardeI*) ⇒ *waarde*

$\text{ceiling}(.456)$	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(*LijstI*) ⇒ *lijst*

$\text{ceiling}(\{-3.1, 1, 2.5\})$	$\{-3., 1, 3.\}$
------------------------------------	------------------

ceiling(*MatrixI*) ⇒ *matrix*

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

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

centralDiff()Catalogus > **centralDiff**(*Uitdr1*, *Var* [= *Waarde*]
[, *Stap*]) ⇒ *uitdrukking*

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

centralDiff(*Uitdr1*, *Var*
[, *Stap*]) | *Var* = *Waarde* ⇒ *uitdrukking***centralDiff**(*Uitdr1*, *Var* [= *Waarde*]
[, *Lijst*]) ⇒ *lijst***centralDiff**(*Lijst1*, *Var* [= *Waarde*]
[, *Stap*]) ⇒ *lijst***centralDiff**(*Matrix1*, *Var* [= *Waarde*]
[, *Stap*]) ⇒ *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 .

char()Catalogus > **char**(*Geheel getal*) ⇒ *teken*

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

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.

χ²2wayCatalogus > **χ²2way** *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. 160).

Zie voor informatie over het effect van lege elementen in een matrix “Lege elementen” (pag. 232).

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)⇒getal als *ondergrens* en *bovengrens* getallen zijn, *lijst* als *ondergrens* en *bovengrens* lijsten zijn

chi2Cdf(ondergrens,bovengrens,df)⇒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. 232.)

 χ^2 GOF

χ^2 GOF *obsLijst,expLijst,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. 160).

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

Uitvoervariabele	Beschrijving
stat.χ²	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.CompList	Chi-kwadraat-statistiekbijdragen van elementen

χ²Pdf()

χ²Pdf(*XWaarde*,*df*)⇒*getal* als *XWaarde* een getal is,
lijst als *XWaarde* een lijst is

chi2Pdf(*XWaarde*,*df*)⇒*getal* als *XWaarde* een getal is,
lijst als *XWaarde* een lijst is

Berekent de kansdichtheidsfunctie (pdf) voor de χ²-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. 232).

ClearAZ

ClearAZ

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

5 → <i>b</i>	5
<i>b</i>	5
ClearAZ	Done
<i>b</i>	"Error: Variable is not defined"

Als een aantal van de variabelen vergrendeld is, veroorzaakt deze opdracht een foutmelding en worden alleen de niet-vergrendelde variabelen gewist. Zie **unLock**, pag. 179.

ClrErr

ClrErr

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

Zie voor een voorbeeld van **ClrErr** Voorbeeld 2 onder het commando **Try** (pag. 172).

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 foutendialogvenster weergegeven zoals normaal is.

Opmerking: zie ook **PassErr**, pag. 119 en **Try**, pag. 172.

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)⇒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)⇒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*

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()**.

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

conj()

conj(*Waarde1*) $\Rightarrow$ *waarde*

conj(*Lijst1*) $\Rightarrow$ *lijst*

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

Geeft de complex geconjugeerde van het argument.

Opmerking: alle onbepaalde variabelen worden behandeld als reële variabelen.

<code>conj(1+2·i)</code>	$1-2 \cdot i$
<code>conj</code> $\left(\begin{bmatrix} 2 & 1-3 \cdot i \\ -i & -7 \end{bmatrix}\right)$	$\begin{bmatrix} 2 & 1+3 \cdot i \\ i & -7 \end{bmatrix}$

constructMat()

constructMat

(
Uitdr

,
Var1

,
Var2

,*aantalRijen,aantalKolommen*) $\Rightarrow$ *matrix*

Geeft een matrix op basis van de argumenten.

Uitdr is een uitdrukking in de variabelen *Var1* en *Var2*. Elementen in de resulterende matrix worden gevormd door *Uitdr* uit te werken voor elke opgehoogde waarde van *Var1* en *Var2*.

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

CopyVar *Var1, Var2*

CopyVar *Var1*, *Var2* kopieert de waarde van variabele *Var1* naar variabele *Var2*, waarbij *Var2* indien nodig gecreëerd wordt. Variabele *Var1* moet een waarde hebben.

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

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.

Define $a(x) = \frac{1}{x}$ Done

Define $b(x)=x^2$ Done

CopyVar a, c : $c(4)$	$\frac{1}{4}$
-------------------------	---------------

CopyVar b, c : $c(4)$
16

$aa.a:=45$	45
------------	----

$aa.b:=6.78$	6.78
--------------	------

CopyVar aa.,bb.Done

```
getVarInfo()
```

<i>aa.a</i>	"NUM"	"0"	0
<i>aa.b</i>	"NUM"	"0"	0
<i>bb.a</i>	"NUM"	"0"	0
<i>bb.b</i>	"NUM"	"0"	0

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

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

cos()

 -toets

cos(*Waarde1*) $\Rightarrow$ *waarde*

In de hoekmodus Graden:

cos(*Lijst1*) $\Rightarrow$ *lijst*

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

cos(*Waarde1*) geeft de cosinus van het argument als een waarde.

$\cos(45)$	0.707107
------------	----------

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

$\cos(\{0,60,90\})$	$\{1,0.5,0\}$
---------------------	---------------

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

In de hoekmodus Decimale graden:

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

In de hoekmodus Radialen:

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

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

cos

(vierkanteMatrix1) $\Rightarrow$ *vierkanteMatrix*

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

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.

In de hoekmodus Radialen:

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

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(WaardeI) ⇒ *waarde*

In de hoekmodus Graden:

cos⁻¹(LijstI) ⇒ *lijst*

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

cos⁻¹(WaardeI) geeft de hoek waarvan de cosinus *WaardeI* is.

In de hoekmodus Decimale graden:

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

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

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

In de hoekmodus Radialen:

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

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

cos⁻¹
(vierkanteMatrixI) ⇒ *vierkanteMatrix*

In de hoekmodus Radialen en rechthoekige complexe opmaak:

$\cos^{-1}()$

-toets

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.

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

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

cosh()

Catalogus > 

cosh(*Waarde1*) $\Rightarrow$ *waarde*

In de hoekmodus Graden:

cosh(*Lijst1*) $\Rightarrow$ *lijst*

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

cosh(*Waarde1*) geeft de cosinus hyperbolicus van het argument.

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

cosh
(*vierkanteMatrix1*) $\Rightarrow$ *vierkanteMatrix*

In de hoekmodus Radialen:

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()**.

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

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

cosh⁻¹()

Catalogus > 

cosh⁻¹(*Waarde1*) $\Rightarrow$ *waarde*

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

cosh⁻¹(*Lijst1*) $\Rightarrow$ *lijst*

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

cosh⁻¹(*Waarde1*) geeft de inverse cosinus hyperbolicus van het argument.

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:

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

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

cot(*Waarde1*) ⇒
waarde

In de hoekmodus Graden:

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

cot(*Lijst1*) ⇒ *lijst*

Geeft de cotangens van *Waarde1* of geeft een lijst van de cotangensen van alle elementen *Lijst1*.

In de hoekmodus Decimale graden:

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

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 Radialen:

$$\cot(\{1, 2.1, 3\}) \quad \{0.642093, -0.584848, -7.01525\}$$

cot⁻¹()**cot⁻¹(Waarde1)⇒waarde**

In de hoekmodus Graden:

cot⁻¹(Lijst1)⇒lijstcot⁻¹(1) 45.

Geeft de hoek waarvan de cotangens *Waarde1* is of geeft een lijst met de inverse cotangens van elk element in *Lijst1*.

In de hoekmodus Decimale graden:

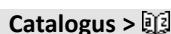
cot⁻¹(1) 50.

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

In de hoekmodus Radialen:

cot⁻¹(1) 0.785398

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

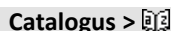
coth()**coth(Waarde1)⇒waarde**

coth(1.2) 1.19954

coth(Lijst1)⇒lijst

coth({1,3,2}) {1.31304,1.00333}

Geeft de cotangens hyperbolicus van *Waarde1* of geeft een lijst van de cotangensen hyperbolicus van alle elementen in *Lijst1*.

coth⁻¹()**coth⁻¹(Waarde1)⇒waarde**coth⁻¹(3.5) 0.293893**coth⁻¹(Lijst1)⇒lijst**coth⁻¹({-2,1,6})
{-0.549306,0.518046,0.168236}

Geeft de inverse cotangens hyperbolicus van *Waarde1* of geeft een lijst met de inverse cotangensen hyperbolicus van elk element in *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*

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

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

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

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. 232.

<code>count(2,4,6)</code>	3
<code>count({2,4,6})</code>	3
<code>count(2,{4,6},{8 10 12 14})</code>	7

countif()Catalogus > 

countif(*Lijst, Criteria*) ⇒ *waarde*

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

Criteria kan zijn:

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

<code>countif({1,3,"abc",undef,3,1},3)</code>	2
---	---

Telt het aantal elementen dat gelijk is aan 3.

<code>countif({"abc","def","abc",3},"def")</code>	1
---	---

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

<code>countif({1,3,5,7,9},?<5)</code>	2
--	---

Telt 1 en 3.

<code>countif({1,3,5,7,9},2<?<8)</code>	3
---	---

Telt 3, 5 en 7.

countIf()Catalogus > 

Lege elementen in de lijst worden genegeerd. Voor meer informatie over lege elementen, zie pag. 232.

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

Telt 1, 3, 7 en 9.

Opmerking: zie ook **sumIf()**, pag. 165 en **frequency()**, pag. 60.

cPolyRoots()Catalogus > 

cPolyRoots(Poly, Var) $\Rightarrow$ lijst

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

cPolyRoots(LijstVanCoëff) $\Rightarrow$ lijst

$\text{cPolyRoots}(y^3+1, y)$	$\{-1, 0.5-0.866025i, 0.5+0.866025i\}$
-------------------------------	--

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

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

Poly moet een veelterm in uitgewerkte vorm met één variabele zijn. Gebruik geen niet-uitgewerkte vormen zoals $y^2 \cdot y + 1$ of $x \cdot x + 2 \cdot x + 1$

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

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

Opmerking: zie ook **polyRoots()**, pag. 122.

crossP()Catalogus > 

crossP(Lijst1, Lijst2) $\Rightarrow$ lijst

$\text{crossP}(\{0.1, 2.2, -5\}, \{1, -0.5, 0\})$	$\{-2.5, -5, -2.25\}$
---	-----------------------

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) $\Rightarrow$ vector

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

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.

csc() -toets

csc(*Waarde1*) $\Rightarrow$ *waarde*

In de hoekmodus Graden:

csc(*Lijst1*) $\Rightarrow$ *lijst*

$\text{csc}(45)$	1.41421
------------------	---------

Geeft de cosecans van *Waarde1* of geeft een lijst met de cosencansen van alle elementen in *Lijst1*.

In de hoekmodus Decimale graden:

$\text{csc}(50)$	1.41421
------------------	---------

In de hoekmodus Radialen:

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

csc⁻¹() -toets

csc⁻¹(*Waarde1*) $\Rightarrow$ *waarde*

In de hoekmodus Graden:

csc⁻¹(*Lijst1*) $\Rightarrow$ *lijst*

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

Geeft de hoek waarvan de cosecans *Waarde1* is of geeft een lijst met de inverse cosecans van elk element in *Lijst1*.

In de hoekmodus Decimale graden:

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

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

In de hoekmodus Radialen:

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

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

csch()Catalogus > **csch(Waarde1)** ⇒ waarde

csch(3) 0.099822

csch(Lijst1) ⇒ lijstcsch({1,2,1,4})
{0.850918,0.248641,0.036644}

Geeft de cosecans hyperbolicus van *Waarde1* of geeft een lijst van de cosecansen hyperbolicus van alle elementen in *Lijst1*.

csch⁻¹()Catalogus > **csch⁻¹(Waarde)** ⇒ waardecsch⁻¹(1) 0.881374**csch⁻¹(Lijst1)** ⇒ lijstcsch⁻¹({1,2,1,3})
{0.881374,0.459815,0.32745}

Geeft de inverse cosecans hyperbolicus van *Waarde1* 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.

CubicRegCatalogus > **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. 160).

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 numerieke of tekst-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. 232).

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. 232.

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.

Geeft 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.

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

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.

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

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

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)

►DD

Catalogus >

Uitdr1 ►DD⇒waarde

Lijst1 ►DD⇒lijst

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 Graden:

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

In de hoekmodus Decimale graden:

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

In de hoekmodus Radialen:

(1.5) ►DD	85.9437°
-------------	----------

►Decimal

Catalogus >

Getal1 ►Decimal⇒waarde

Lijst1 ►Decimal⇒waarde

Matrix1 ►Decimal⇒waarde

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

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

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

Define (Definiëren)

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

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

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.

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. 40 en **Define LibPub**, pag. 40.

```
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

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. 39 en **Define LibPub**, pag. 40.

Define LibPub

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. 39 en **Define LibPriv**, pag. 40.

deltaList()Zie Δ List(), pag. 88.**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
-----------------	------

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

Als een aantal van de variabelen vergrendeld is, veroorzaakt deze opdracht een foutmelding en worden alleen de niet-vergrendelde variabelen gewist. Zie **unLock**, pag. 179.

DelVar

Catalogus > 

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									
<i>getVarInfo()</i>	<table><tr><td><i>aa.a</i></td><td>"NUM"</td><td>"0"</td></tr><tr><td><i>aa.b</i></td><td>"NUM"</td><td>"0"</td></tr><tr><td><i>aa.c</i></td><td>"NUM"</td><td>"0"</td></tr></table>	<i>aa.a</i>	"NUM"	"0"	<i>aa.b</i>	"NUM"	"0"	<i>aa.c</i>	"NUM"	"0"
<i>aa.a</i>	"NUM"	"0"								
<i>aa.b</i>	"NUM"	"0"								
<i>aa.c</i>	"NUM"	"0"								
<i>DelVar aa.</i>	<i>Done</i>									
<i>getVarInfo()</i>	"NONE"									

delVoid()

Catalogus > 

delVoid(*Lijst1*)⇒*lijst*

Geeft een lijst met de inhoud van *Lijst1* waaruit alle lege elementen verwijderd zijn.

Zie voor meer informatie over lege elementen pag. 232.

<i>delVoid</i> ({1,void,3})	{1,3}
-----------------------------	-------

det()

Catalogus > 

det(*vierkanteMatrix*[,
Tolerantie])⇒*uitdrukking*

Geeft de determinant van *vierkanteMatrix*.

Optioneel wordt elk matricelement 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 wordt gebruikt, dan wordt de

<i>det</i> ($\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$)	-2
$\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix} \rightarrow \text{mat1}$	$\begin{bmatrix} 1. \text{E}20 & 1 \\ 0 & 1 \end{bmatrix}$
<i>det</i> (<i>mat1</i>)	0
<i>det</i> (<i>mat1</i> ,1)	1. E20

standaardtolerantie berekend als:

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

diag()

diag(Lijst) ⇒ matrix

diag([2 4 6])

$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$

diag(rijMatrix) ⇒ matrix

$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$

diag(kolomMatrix) ⇒ matrix

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

diag(vierkanteMatrix) ⇒ rijMatrix

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

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

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

vierkanteMatrix moet vierkant zijn.

diag(Ans)

$\begin{bmatrix} 4 & 2 & 9 \end{bmatrix}$

dim()

dim(Lijst) ⇒ geheel getal

dim({0,1,2})

3

Geeft de afmeting van *Lijst*.

dim(Matrix) ⇒ lijst

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

{3,2}

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

dim(String) ⇒ geheel getal

dim("Hello")

5

Geeft het aantal tekens in de tekenreeks *String*.

dim("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* ...] ...

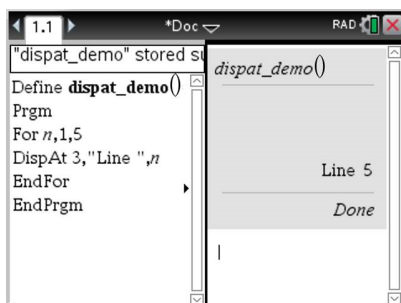
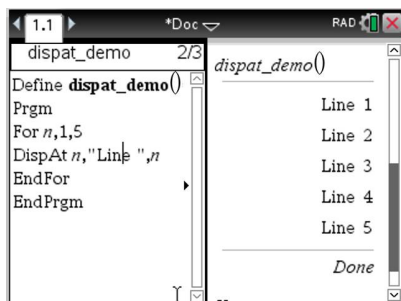
DispAt stelt u in staat de regel op te geven waarin de gespecificeerde uitdrukking of string op het scherm zal worden weergegeven.

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 rekenmachinescherf - 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 Disp.
Ongeldig gegevenstype	Eerste argument moet een getal zijn.

Foutbericht Leeg: DispAt leeg	Beschrijving De "Hello World"-gegevenstypefout wordt gegeven voor "void" (indien 'terugbellen' is gedefinieerd)
---	---

►DMS	Catalogus >
-------------	---

Waarde ►DMS

In de hoekmodus Graden:

Lijst ►DMS

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

Matrix ►DMS

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

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. 210) 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.

dotP()	Catalogus >
---------------	---

dotP(Lijst1, Lijst2)⇒uitdrukking

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

Geeft het inwendige product van twee lijsten.

dotP(Vector1, Vector2)⇒uitdrukking

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

Geeft het inwendige product van twee vectoren.

Beide moeten rijvectoren zijn, of beide moeten kolomvectoren zijn.

e^()		e^x -toets																		
e^(<i>Waarde1</i>)⇒ <i>waarde</i>	e ¹	2.71828																		
Geeft e tot de macht <i>Waarde1</i> .	e ^{3²}	8103.08																		
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 reï θ polaire vorm invoeren. Gebruik deze vorm echter alleen in de hoekmodus Radialen; hij veroorzaakt een domeinfout in de hoekmodi Graden en Decimale graden.																				
e^(<i>Lijst1</i>)⇒ <i>lijst</i>	e ^{1,1,.0.5}	{2.71828,2.71828,1.64872}																		
Geeft e tot de macht van elk element in <i>Lijst1</i> .																				
e^(<i>vierkanteMatrix1</i>)⇒ <i>vierkanteMatrix</i>	<table><tr><td>1</td><td>5</td><td>3</td></tr><tr><td>4</td><td>2</td><td>1</td></tr><tr><td>6</td><td>-2</td><td>1</td></tr></table>	1	5	3	4	2	1	6	-2	1	<table><tr><td>782.209</td><td>559.617</td><td>456.509</td></tr><tr><td>680.546</td><td>488.795</td><td>396.521</td></tr><tr><td>524.929</td><td>371.222</td><td>307.879</td></tr></table>	782.209	559.617	456.509	680.546	488.795	396.521	524.929	371.222	307.879
1	5	3																		
4	2	1																		
6	-2	1																		
782.209	559.617	456.509																		
680.546	488.795	396.521																		
524.929	371.222	307.879																		
Geeft de 'e tot de macht van <i>vierkanteMatrix1</i> '. Dit is niet hetzelfde als het berekenen van e tot de macht van elk element. Zie voor informatie over de berekeningsmethode cos() .																				
<i>vierkanteMatrix1</i> moet diagonaliseerbaar zijn. Het resultaat bevat altijd getallen met een drijvende komma.																				

eff()	Catalogus > 	
eff(<i>nominaalPercentage</i> , <i>CpY</i>)⇒ <i>waarde</i>	eff(5.75,12)	5.90398
Financiële functie die het nominale rentepercentage <i>nominaalPercentage</i> converteert naar een jaarlijks effectief percentage, waarbij <i>CpY</i> het aantal samengestelde periodes per jaar is.		
<i>nominaalPercentage</i> moet een reëel getal zijn, en <i>CpY</i> moet een reëel getal > 0 zijn.		

Opmerking: zie ook **nom()**, pag. 110.

eigVc()

eigVc(*vierkanteMatrix*) $\Rightarrow$ *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:

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

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.

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

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

eigVc(*mI*)

-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.

eigVl()

eigVl(*vierkanteMatrix*) $\Rightarrow$ *lijst*

In de rechthoekige complexe opmaak-modus:

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.

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

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

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.

ElseIf

Catalogus > 

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 $\neg x$

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

Return x

ElseIf $x = 10$ Then

Return 3

EndIf

EndFunc

Done

EndFor

Zie For, pag. 57.

EndFunc

Zie Func, pag. 62.

EndIf

Zie If, pag. 73.

EndLoop

Zie Loop, pag. 95.

EndPrgm

Zie Prgm, pag. 123.

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

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

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*).

Differentiaalvergelijking:

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

$$\text{euler}\left\{0,001 \cdot y \cdot (100 - y), t, y, \{0, 100\}, 10, 1\right\}$$

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.

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

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

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

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 $VarStap / eulerStap$.

eval ()

Hub Menu

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.

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

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

<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"

Opmerking: Zie ook **Get** (pag. 63), **GetStr** (pag. 70), en **Send** (pag. 146).

Exit

Exit

Sluit het huidige **For**, **While** of **Loop**-blok af.

Exit 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.

Functielijst:

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

exp()

e^x-toets

exp(*Waarde1*)⇒*waarde*

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

Geeft **e** tot de macht *Waarde1*.

e ^{3²}	8103.08
----------------------------	---------

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

U kunt een complex getal in rei θ 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 }	{ 2.71828,2.71828,1.64872 }
---------------------------	-----------------------------

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

exp(*vierkanteMatrix1*)⇒*vierkanteMatrix*

e ^{<table><tr><td>1</td><td>5</td><td>3</td></tr><tr><td>4</td><td>2</td><td>1</td></tr><tr><td>6</td><td>-2</td><td>1</td></tr></table>}	1	5	3	4	2	1	6	-2	1	<table><tr><td>782.209</td><td>559.617</td><td>456.509</td></tr><tr><td>680.546</td><td>488.795</td><td>396.521</td></tr><tr><td>524.929</td><td>371.222</td><td>307.879</td></tr></table>	782.209	559.617	456.509	680.546	488.795	396.521	524.929	371.222	307.879
1	5	3																	
4	2	1																	
6	-2	1																	
782.209	559.617	456.509																	
680.546	488.795	396.521																	
524.929	371.222	307.879																	

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.

expr()

Catalogus >

expr(*String*)⇒*uitdrukking*

"Define cube(x)=x^3" → <i>funcstr</i>
"Define cube(x)=x^3"

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

expr(<i>funcstr</i>)	<i>Done</i>
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. 160).

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 numerieke of tekst-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. 232).

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>

factor()Catalogus > 

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.

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

F Cdf()Catalogus > **F Cdf**

(*ondergrens*,*bovengrens*,*dfTeller*,*dfNoemer*) ⇒ *getal*
als *ondergrens* en *bovengrens* getallen zijn, *lijst* als
ondergrens en *bovengrens* lijsten zijn

FCdf

(*ondergrens*,*bovengrens*,*dfTeller*,*dfNoemer*) ⇒ *getal*
als *ondergrens* en *bovengrens* getallen zijn, *lijst* als
ondergrens en *bovengrens* lijsten zijn

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

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

Fill

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

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

matrixVar moet al bestaan.

1 2	$\rightarrow$ <i>amatrix</i>	1 2
3 4		3 4

Fill 1.01, <i>amatrix</i>	Done
---------------------------	------

<i>amatrix</i>	1.01 1.01
	1.01 1.01

Fill *Waarde*, *lijstVar* $\Rightarrow$ *lijst*

Vervang ieder element in variabele *lijstVar* door *Waarde*.

lijstVar moet al bestaan.

{1,2,3,4,5} $\rightarrow$ <i>alist</i>	{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. 160).

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. 232.

Uitvoervariabele	Beschrijving
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

floor()

floor(*Waarde1*) $\Rightarrow$ geheel getal

$\text{floor}(-2.14)$ -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*) $\Rightarrow$ lijst

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

floor(*Matrix1*) $\Rightarrow$ matrix

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

Geeft een lijst of matrix van de floor-waarde van elk element.

Opmerking: zie ook **ceiling()** en **int()**.

For

For *Var*, *Laag*, *Hoog* [, *Stap*]

Blok

EndFor

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

Voert de beweringen in *Blok* iteratief uit voor elke waarde van *Var*, van *Laag* naar *Hoog*, in stappen van *Stap*.

Var mag geen systeemvariabele zijn.

Stap 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()

**format(*Waarde*[,
opmaakString])** ⇒ *string*

Geeft *Waarde* als een tekenreeks op basis van de *opmaaktemplate*.

opmaakString is een string die de volgende vorm moet hebben: “F[n]”, “S[n]”, “E[n]”, “G[n][c]”, waarbij [] optionele gedeeltes aangeeft.

F[n]: Vaste opmaak. n is het aantal cijfers dat weergegeven moet worden achter de decimale punt.

S[n]: Wetenschappelijke opmaak. n is het aantal cijfers dat weergegeven moet worden achter de decimale punt.

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 scheidingsteken) tevens in groepen van drie. c specificeert het groep-scheidingsteken; de standaardinstelling is een komma. Als c een punt is, wordt de radix weergegeven als een komma.

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

[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()**Catalogus >**

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

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

fPart(*Lijst1*) $\Rightarrow$ *lijst*

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

fPart(*Matrix1*) $\Rightarrow$ *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()**Catalogus >**

F**Pdf**(*XWaarde*, *dfTeller*, *dfNoemer*) $\Rightarrow$ *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►lijst()**Catalogus >**

freqTable►list

(*Lijst1*, *freqGeheelGetalLijst*) $\Rightarrow$ *lijst*

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

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.

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

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. 232.

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, $b(n) > ?$) }.

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. 232.

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

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.

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

Opmerking: zie ook `countif()`, pag. 32.

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. 160).

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. 232).

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 = $n_2 - 1$
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*Blok***EndFunc**

Template voor het creëren van een door de gebruiker gedefinieerde functie.

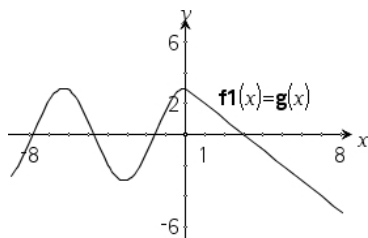
Blok kan een enkele bewering of een serie beweringen zijn die gescheiden worden door het teken “:”, of een serie beweringen op aparte regels. De functie kan de instructie **Return** gebruiken om een specifiek resultaat te retourneren.

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()**

gcd(*Waarde1*, *Waarde2*) $\Rightarrow$ *uitdrukking*

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(*Lijst1*, *Lijst2*) $\Rightarrow$ *lijst*

Geeft de grootste gemene delers van de overeenkomstige elementen in *Lijst1* en *Lijst2*.

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

Geeft de grootste gemene delers van de overeenkomstige elementen in *Matrix1* en *Matrix2*.

gcd(18,33)	3
------------	---

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

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

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

geomCdf(*p*, *bovengrens*) voor $P(1 \leq X \leq \text{bovengrens}) \Rightarrow$ *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()

geomPdf(*p*, *XWaarde*) $\Rightarrow$ *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*.

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.

Voorbeeld: Vraag de huidige waarde van de 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*.

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

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. 70 en **Send**, pag. 146.

getDenom()

Catalogus > 

getDenom(*Breuk1*) ⇒ waarde

Transformeert het argument in een uitdrukking met een vereenvoudigde gemeenschappelijke noemer, en geeft vervolgens de noemer ervan.

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

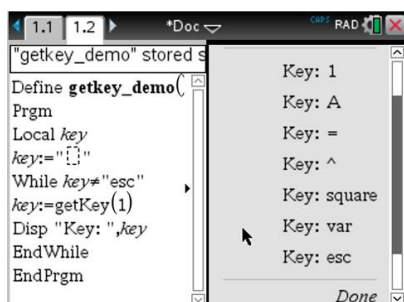
get(Key[01]) ⇒ returnString

Beschrijving: **getKey()** - stelt een TI-Basic programma in staat om toetsenbordinput 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"
Doc	N.v.t.	"doc"
Tabblad	Tabblad	"tabblad"
Touchpad - onderaan klikken	Pijl omlaag	"omlaag"
Menu	N.v.t.	"menu"

rekenmachine/emulatortoets	Desktop	Teruggeven waarde
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)
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

rekenmachine/emulatortoets ?!	Desktop N.v.t.	Teruggeven waarde "?!"
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 desктоptoetsen 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™ 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	Beëindig programma, handel gebeurtenis af	Hetzelfde als de rekenmachine. (TI-Nspire™ Student Software, TI-Nspire™ Navigator™ NC Teacher Software-alleen)

Gebeurtenis	Rekenmachine	Desktop - TI-Nspire™- leerlingensoftware
rekenmachine of desktop- rekenmachine)		
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. 91 en **unLock**, pag. 179.

<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.

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. 149.

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

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
Vectoropmaak	6	1=Rechthoekig, 2=Cilindrisch, 3=Bolvormig
Grondtal	7	1=Decimaal, 2=Hexadecimaal, 3=Binair

getNum()

Catalogus > 

getNum(*BreukI*) ⇒ *waarde*

Transformeert het argument naar een uitdrukking met een vereenvoudigde gemeenschappelijke noemer, en geeft vervolgens de teller ervan.

$x:=5; y:=6$	6
$\text{getNum}\left(\frac{x+2}{y-3}\right)$	7
$\text{getNum}\left(\frac{2}{7}\right)$	2
$\text{getNum}\left(\frac{1}{x} + \frac{1}{y}\right)$	11

GetStr

Hub Menu

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. 63 en **Send**, pag. 146.

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()**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>1</td></tr><tr><td>y</td><td>"LIST"</td><td>"LibPriv "</td><td>0</td></tr><tr><td>z</td><td>"FUNC"</td><td>"LibPub "</td><td>0</td></tr></table>	x	"NUM"	"{"	1	y	"LIST"	"LibPriv "	0	z	"FUNC"	"LibPub "	0
x	"NUM"	"{"	1										
y	"LIST"	"LibPriv "	0										
z	"FUNC"	"LibPub "	0										
getVarInfo(tmp3)	"Error: Argument must be a string"												
getVarInfo("tmp3")	<table><tr><td>[volcy12</td><td>"NONE"</td><td>"LibPub "</td><td>0]</td></tr></table>	[volcy12	"NONE"	"LibPub "	0]								
[volcy12	"NONE"	"LibPub "	0]										

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

Goto

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

Uitdr1 ► **Grad**⇒*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:

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

In de hoekmodus Radialen:

I

identity()

identity(*Geheel getal*) ⇒ *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

If

If *BooleaanseUitdr*
Bewering

If *BooleaanseUitdr* **Then**
Blok

EndIf

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

$g(-2)$ 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 $g(x) = \text{Func}$

Done

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

Else

Return x

EndIf

EndFunc

 $g(12)$

12

 $g(-12)$

12

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

Return 5

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

Return 3

EndIf

EndFunc

Done

 $g(-4)$

4

 $g(10)$

3

ifFn()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*

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

wordt uitgewerkt naar waar, wordt het 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(*WaardeI*) $\Rightarrow$ *waarde*

Geeft het imaginaire deel van het argument.

imag(*LijstI*) $\Rightarrow$ *lijst*

Geeft een lijst met de imaginaire delen van de elementen.

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

Geeft een matrix met de imaginaire delen van de elementen.

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

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

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

inString()Catalogus > 

inString(*bronString*, *subString*[, *Start*])
 $\Rightarrow$ *geheel getal*

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

int (*waarde*) $\Rightarrow$ *geheel getal*
int(*Lijst1*) $\Rightarrow$ *lijst*
int(*Matrix1*) $\Rightarrow$ *matrix*

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

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()Catalogus > 

intDiv (*Getal1*, *getal2*) $\Rightarrow$ *geheel getal*
intDiv(*Lijst1*, *Lijst2*) $\Rightarrow$ *lijst*
intDiv(*Matrix1*, *Matrix2*) $\Rightarrow$ *matrix*

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

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.

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 **onedef**.

xLijst, *yLijst* en *yAccentLijst* moeten dezelfde dimensie ≥ 2 hebben en uitdrukkingen bevatten die vereenvoudigd worden tot getallen.

xWaarde kan een getal of een lijst met getallen zijn.

Differentiaalvergelijking:

$y' = -3 \cdot y + 6 \cdot t + 5$ en $y(0) = 5$

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

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

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.,6.5,7.,7.5,8.,8.5,9.,9.5,10.}
xlist:=mat▶list(rk[1])
{0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.}
ylist:=mat▶list(rk[2])
{5.,3.19499,5.00394,6.99957,9.00593,10.9978}
yprimelist:=-3*y+6*t+5|y=ylist and t=xlist
{-10.,1.41503,1.98819,2.00129,1.98221,2.0061}
interpolate(xvaluelist,xlist,ylist,yprimelist)
{5.,2.67062,3.19499,4.02782,5.00394,6.00011,7.00011,8.00011,9.00011,10.00011}
```

inv χ^2 ()

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(*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()

invBinom

(
CumulatieveKans
AantalPogingen,Kans,
UitvoerForm) $\Rightarrow$ *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*)..

UitvoerForm=0, resultaat wordt weergegeven als een scalair (standaard).

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

invBinomN()

invBinomN(*CumulatieveKans,Kans*,
AantalSucces,UitvoerForm) $\Rightarrow$ *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*).

UitvoerForm=0, resultaat wordt weergegeven als een scalair (standaard).

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
$\text{invBinomN}(0.01, 0.7, 49, 1)$	$\begin{bmatrix} 85 & 0.010451 \\ 86 & 0.00709 \end{bmatrix}$

invBinomN()Catalogus > 

UitvoerForm=1, resultaat wordt weergegeven als een matrix.

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()Catalogus > 

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()Catalogus > 

iPart(*Getal*) $\Rightarrow$ *geheel getal*

iPart(*Lijst1*) $\Rightarrow$ *lijst*

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

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

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()Catalogus > 

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.

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

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.

Opmerking: Zie ook *mirr()*, pag. 101.

isPrime()

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.

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

isVoid()

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.

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

Zie voor meer informatie over lege elementen pag. 232.

L

Lbl *labelNaam*

Definieert een label met de naam *labelNaam* binnen een functie.

U kunt een **Goto** *labelNaam*-instructie gebruiken om de besturing naar de instructie onmiddellijk na het label te brengen.

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

lcm(*Getal1*, *Getal2*) $\Rightarrow$ *uitdrukking*

lcm(*Lijst1*, *Lijst2*) $\Rightarrow$ *lijst*

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

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 veelvouden van de overeenkomstige elementen.

$\text{lcm}(6,9)$	18
$\text{lcm}\left(\left\{\frac{1}{3}, -14, 16\right\}, \left\{\frac{2}{15}, 7, 5\right\}\right)$	$\left\{\frac{2}{3}, 14, 80\right\}$

left(*bronString*[, *Aantal*]) $\Rightarrow$ *string*

left("Hello",2)

"He"

Geeft het meest linkse *Aantal* tekens in tekenreeks *bronString*.

Als u *Aantal* weglaat, wordt de hele *bronString* gegeven.

left(*Lijst1*[, *Aantal*]) $\Rightarrow$ *lijst*

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

{1,3,-2}

Geeft het meest linkse *Aantal* elementen in *Lijst1*.

Als u *Aantal* weglaat, wordt de hele *Lijst1* gegeven.

left(*Vergelijken*) $\Rightarrow$ *uitdrukking*

Geeft het linkerlid van een vergelijking of ongelijkheid.

libShortcut()

libShortcut(*BibliotheekNaamString*, *SnelNaamString* [, *LibPrivVlag*]) $\Rightarrow$ *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. 26).

Zie voor het wissen van een variabelegroep **DelVar** (pag. 41).

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")

<i>clearmat</i>	"FUNC"	"LibPub "
<i>gauss1</i>	"PRGM"	"LibPriv "
<i>gauss2</i>	"FUNC"	"LibPub "

libShortcut("linalg2","la")

{*la.clearmat*,*la.gauss2*}

libShortcut("linalg2","la",1)

{*la.clearmat*,*la.gauss1*,*la.gauss2*}

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. 160).

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 numerieke of tekst-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. 232).

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. 160).

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 numerieke of tekst-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. 232).

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. 160).

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. 232).

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
stat.s	Standaardfout van de lijn

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

LinRegtTestCatalogus > **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. 160).

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $a + b \cdot x$
stat.t	t -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*, ...) $\Rightarrow$ lijst

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

linSolve(*LineaireVgl1* en *LineaireVgl2* en ..., *Var1*, *Var2*, ...) $\Rightarrow$ lijst

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

linSolve(*{LineaireVgl1, LineaireVgl2, ...}*, *Var1*, *Var2*, ...) $\Rightarrow$ lijst

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

linSolve(*StelselLineaireVgl*, *{Var1, Var2, ...}*) $\Rightarrow$ lijst

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

linSolve(*LineaireVgl1* en *LineaireVgl2* en ..., *{Var1, Var2, ...}*) $\Rightarrow$ lijst

linSolve(*{LineaireVgl1, LineaireVgl2, ...}*, *{Var1, Var2, ...}*) $\Rightarrow$ 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.

ln(*WaardeI*)⇒*waarde*

ln(2.) 0.693147

ln(*LijstI*)⇒*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.

$$\ln\{-3, 1.2, 5\}$$

"Error: Non-real calculation"

Als de complexe opmaak-modus Rechthoekig is:

$$\ln\{-3, 1.2, 5\}$$

$$\{1.09861+3.14159 \cdot i, 0.182322, 1.60944\}$$
ln(*vierkanteMatrixI*)⇒*vierkanteMatrix*

Geeft de natuurlijke logaritme van *vierkanteMatrixI*. Dit is niet hetzelfde als het berekenen van de natuurlijke logaritme van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

In de hoekmodus Radialen en rechthoekige complexe opmaak:

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

$$\begin{bmatrix} 1.83145+1.73485 \cdot i & 0.009193-1.49086 \\ 0.448761-0.725533 \cdot i & 1.06491+0.623491 \cdot i \\ -0.266891-2.08316 \cdot i & 1.12436+1.79018 \cdot i \end{bmatrix}$$

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

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

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. 160).

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 numerieke of tekst-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. 232).

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>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>

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. 179 en **getLockInfo()**, pag. 69.

<i>a</i> :=65	65
---------------	----

Lock <i>a</i>	<i>Done</i>
---------------	-------------

getLockInfo(<i>a</i>)	1
-------------------------	---

<i>a</i> :=75	"Error: Variable is locked."
---------------	------------------------------

DelVar <i>a</i>	"Error: Variable is locked."
-----------------	------------------------------

Unlock <i>a</i>	<i>Done</i>
-----------------	-------------

<i>a</i> :=75	75
---------------	----

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

log()

ctrl **10^x** -toetsen

log(*Waarde1* [, *Waarde2*]) ⇒ *waarde*

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

log(*Lijst1* [, *Waarde2*]) ⇒ *lijst*

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

Geeft de logaritme met grondtal-*Waarde2*- van het eerste argument.

$$\log_3 (10) - \log_3 (5) \quad 0.63093$$

Opmerking: zie ook **Log-template**, pag. 2.

Geeft bij een lijst de logaritme met grondtal-*Waarde2*- van de elementen.

Als de complexe opmaak-modus Reëel is:

$$\log_{10} (\{-3, 1.2, 5\})$$

"Error: Non-real calculation"

Als het tweede argument wordt weggelaten, dan wordt 10 als grondtal gebruikt.

Als de complexe opmaak-modus Rechthoekig is:

$$\log_{10} (\{-3, 1.2, 5\})$$
$$\{0.477121 + 1.36438 \cdot i, 0.079181, 0.69897\}$$

log(*vierkanteMatrix1* [, *Waarde*]) ⇒ *vierkanteMatrix*

In de hoekmodus Radialen en rechthoekige complexe opmaak:

$$\log_{10} \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 0.795387 + 0.753438 \cdot i & 0.003993 - 0.6474i \\ 0.194895 - 0.315095 \cdot i & 0.462485 + 0.2707i \\ -0.115909 - 0.904706 \cdot i & 0.488304 + 0.7774i \end{bmatrix}$$

Geeft de logaritme met grondtal-*Waarde*- van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van de logaritme met grondtal-*Waarde*- 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.

Logistic

Catalogus >

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. 160).

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 numerieke of tekst-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. 232).

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>
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>

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. 160).

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 numerieke of tekst-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. 232).

Uitvoervariabele	Beschrijving
stat.RegEqn	Regressievergelijking: $c/(1+a \cdot e^{-bx})+d$
stat.a, stat.b, stat.c, stat.d	Regressiecoëfficiënten
stat.Resid	Residuen uit de regressie
stat.XReg	Lijst van 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>

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

```

	Done
rollcount()	16
rollcount()	3

LU
Catalogus >

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*.

$lMatrix \cdot uMatrix = pMatrix \cdot matrix$

$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$
LU <i>m1</i> , <i>lower</i> , <i>upper</i> , <i>perm</i>	Done
<i>lower</i>	$\begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{6} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$
<i>upper</i>	$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$
<i>perm</i>	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

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.

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*.

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

mat▶list ($\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$)	$\{1, 2, 3\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
mat▶list (<i>m1</i>)	$\{1, 2, 3, 4, 5, 6\}$

max()

max(Waarde1, Waarde2) ⇒ *uitdrukking*

max(Lijst1, Lijst2) ⇒ *lijst*

max(Matrix1, Matrix2) ⇒ *matrix*

max (2.3, 1.4)	2.3
max ($\{1, 2\}, \{-4, 3\}$)	$\{1, 3\}$

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*

$$\text{max}(\{0,1,-7,1.3,0.5\}) \quad 1.3$$

Geeft het maximumelement in *lijst*.

max(Matrix1)⇒*matrix*

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

Geeft een rijvector met het maximumelement van elke kolom in *Matrix1*.

Lege elementen worden genegeerd.
Voor meer informatie over lege elementen, zie pag. 232.

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

mean(Lijst[,freqLijst])⇒*uitdrukking*

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

Geeft het gemiddelde van de elementen in *Lijst*.

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

Elk element uit *freqLijst* telt het aantal malen dat het overeenkomstige element in *Lijst* achter elkaar voorkomt.

mean(Matrix1[,freqMatrix])⇒*matrix*

Geeft een rijvector van de gemiddelden van alle kolommen in *Matrix1*.

In de rechthoekige vectoropmaak:

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

Elk element uit *freqMatrix* telt het aantal malen dat het overeenkomstige element in *Matrix1* achter elkaar voorkomt.

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

Lege elementen worden genegeerd.
Voor meer informatie over lege elementen, zie pag. 232.

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

median(Lijst[,freqLijst])⇒uitdrukking

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

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

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

Geeft een rijvector met de medianen van de kolommen in *Matrix1*.

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. 232.

MedMed

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. 160).

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 numerieke of tekst-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. 232).

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*]) ⇒ *string*

Geeft *Aantal* tekens uit de tekenreeks *bronString*, beginnend met teken nummer *Start*.

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

mid()	Catalogus > 
mid(<i>bronLijst</i>, <i>Start</i> [, <i>Aantal</i>]) ⇒ <i>lijst</i>	$\text{mid}(\{ \{ 9, 8, 7, 6 \}, 3 \}) \quad \{ 7, 6 \}$
Geeft <i>Aantal</i> elementen uit <i>bronLijst</i> , beginnend met element nummer <i>Start</i> .	$\text{mid}(\{ \{ 9, 8, 7, 6 \}, 2, 2 \}) \quad \{ 8, 7 \}$
Als <i>Aantal</i> wordt weggelaten of groter is dan de afmeting van <i>bronLijst</i> , dan worden alle elementen uit <i>bronLijst</i> gegeven, beginnend met element nummer <i>Start</i> .	$\text{mid}(\{ \{ 9, 8, 7, 6 \}, 1, 2 \}) \quad \{ 9, 8 \}$
	$\text{mid}(\{ \{ 9, 8, 7, 6 \}, 1, 0 \}) \quad \{ \}$
mid(<i>bronStringLijst</i>, <i>Start</i> [, <i>Aantal</i>]) ⇒ <i>lijst</i>	$\text{mid}(\{ \{ "A", "B", "C", "D" \}, 2, 2 \}) \quad \{ "B", "C" \}$
Geeft <i>Aantal</i> strings uit de lijst met strings <i>bronStringLijst</i> , beginnend met element nummer <i>Start</i> .	

min()	Catalogus > 
min(<i>Waarde1</i>, <i>Waarde2</i>) ⇒ <i>uitdrukking</i>	$\text{min}(2, 3, 1, 4) \quad 1, 4$
min(<i>Lijst1</i>, <i>Lijst2</i>) ⇒ <i>lijst</i>	$\text{min}(\{ \{ 1, 2 \}, \{ -4, 3 \} \}) \quad \{ -4, 2 \}$
min(<i>Matrix1</i>, <i>Matrix2</i>) ⇒ <i>matrix</i>	
Geeft het minimum van de twee argumenten. Als de argumenten twee lijsten of matrices zijn, dan wordt een lijst of matrix met de minimumwaarde van elk paar corresponderende gegevens gegeven.	
min(<i>Lijst</i>) ⇒ <i>uitdrukking</i>	$\text{min}(\{ \{ 0, 1, -7, 1, 3, 0, 5 \} \}) \quad -7$
Geeft het minimelement van <i>Lijst</i> .	
min(<i>Matrix1</i>) ⇒ <i>matrix</i>	$\text{min} \left(\begin{bmatrix} 1 & -3 & 7 \\ -4 & 0 & 0,3 \end{bmatrix} \right) \quad \begin{bmatrix} -4 & -3 & 0,3 \end{bmatrix}$
Geeft een rijvector met het minimelement van elke kolom in <i>Matrix1</i> .	
Opmerking: zie ook max() .	

mirr
(
financPercentage
,herinvestPercentage,CF0,CFLijst
[,CFFreq])

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

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. 79.

mod(*Waarde1, Waarde2*)⇒*uitdrukking*

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

mod(*Lijst1, Lijst2*)⇒*lijst*

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

mod(*Matrix1, Matrix2*)⇒*matrix*

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

Geeft het eerste argument modulus het tweede argument zoals gedefinieerd wordt door de identiteiten:

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

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

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

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

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

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. 135

mRow()

mRow(*Waarde*, *Matrix1*, *Index*) $\Rightarrow$ *matrix*

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

Geeft een kopie van *Matrix1* met elk element in rij *Index* van *Matrix1* vermenigvuldigd met *Waarde*.

mRowAdd()

mRowAdd(*Waarde*, *Matrix1*, *Index1*, *Index2*) $\Rightarrow$ *matrix*

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

Geeft een kopie van *Matrix1* met elk element in rij *Index2* van *Matrix1* vervangen door:

$$\text{Waarde} \cdot \text{rij } \text{Index1} + \text{rij } \text{Index2}$$

MultReg

MultReg *Y*, *X1*[*X2*[*X3*,...[,*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. 160).

Alle lijsten moeten gelijke afmetingen hebben.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 232).

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.ŷLijst	$\hat{y}$ 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. 160).

Alle lijsten moeten gelijke afmetingen hebben.

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 232).

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,...}
stat.Resid	Residuen uit de regressie

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. 160).

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 232).

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 bLijst
stat.PList	Lijst van P -waarden voor elke t -statistiek
stat.SEList	Lijst van standaardfouten voor coëfficiënten in bLijst
stat.yLijst	$\hat{y}$ Lijst = $b_0 + b_1 \cdot x_1 + \dots$
stat.Resid	Residuen uit de regressie

Uitvoervariabele	Beschrijving
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 residue en de invloed
stat.Leverage	Maat voor hoever de waarden van de onafhankelijke variabelen van hun gemiddelde waarden af liggen

N

nand (niet en)

ctrl **=** **toetsen**

BooleaanseUitdr1 **nand**

BooleaanseUitdr2 levert *Booleaanse uitdrukking*

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(Waarde1, Waarde2) ⇒ uitdrukking

Voor geheel getal *Waarde1* en *Waarde2* met $Waarde1 \geq Waarde2 \geq 0$, is **nCr()** het aantal combinaties van *Waarde1* dingen die met *Waarde2* keer tegelijk genomen zijn. (Dit is ook bekend als een binomiale coëfficiënt.)

$nCr(z, 3) _{z=5}$	10
$nCr(z, 3) _{z=6}$	20

nCr(Waarde, 0) ⇒ 1

nCr(Waarde, negGeheel getal) ⇒ 0

nCr(Waarde, posGeheel getal) ⇒
 $Waarde \cdot (Waarde - 1) \cdot \dots$
 $(Waarde - posGeheel\ getal + 1) /$
 $posGeheel\ getal!$

nCr(Waarde, nietGeheel getal) ⇒ uitdrukking!
 $((Waarde - nietGeheel\ getal)! \cdot nietGeheel\ getal!)$

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

Geeft een matrix met combinaties op basis van de overeenkomstige elementparen in de twee matrices. De argumenten moeten matrices van dezelfde afmeting zijn.

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

nDerivative()

nDerivative(*Uitdr1*, *Var*=*Waarde* [, *Orde*]) $\Rightarrow$ *waarde*

nDerivative(*Uitdr1*, *Var* [, *Orde*]) | *Var*=*Waarde* $\Rightarrow$ *waarde*

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.

Als de variabele *Var* geen numerieke waarde bevat, moet u *Waarde* opgeven.

Orde van de afgeleide moet 1 of 2 zijn.

Opmerking: het algoritme **nDerivative()** kent een beperking: het werkt recursief door de niet-vereenvoudigde uitdrukking heen, door de numerieke waarde van de eerste afgeleide (en indien van toepassing de tweede) en de uitwerking van iedere subuitdrukking te berekenen, wat tot een onverwachte uitkomst kan leiden.

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

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

Zie het voorbeeld rechts. De eerste afgeleide van $x \cdot (x^2 + x)^{1/3}$ voor $x=0$ is gelijk aan 0. Omdat de eerste afgeleide van de subuitdrukking $(x^2 + x)^{1/3}$ echter onbepaald is voor $x=0$, en deze waarde gebruikt wordt om de afgeleide van de gehele uitdrukking te berekenen, geeft **nDerivative()** de uitkomst als onbepaald en toont een waarschuwingsbericht.

Als u deze beperking tegenkomt, verifieer de oplossing dan grafisch. U kunt ook proberen **centralDiff()** te gebruiken.

newList()

newList(aantalElementen) $\Rightarrow$ lijst

newList(4) $\{0, 0, 0, 0\}$

Geeft een lijst met de afmeting *aantalElementen*. Elk element is nul.

newMat()

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()

nfMax(Uitdr, Var) $\Rightarrow$ waarde

nfMax($x^2 - 2 \cdot x - 1, x$) -1.

nfMax(Uitdr, Var, ondergrens) $\Rightarrow$ waarde

nfMax($0.5 \cdot x^3 - x - 2, x, -5, 5$) 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.

nfMin(Uitdr, Var)⇒waarde

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

nfMin(Uitdr, Var, ondergrens)⇒waarde

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

nfMin(Uitdr, Var, ondergrens, bovangrens)⇒waarde

nfMin(Uitdr, Var) | ondergrens ≤ Var ≤ bovangrens⇒waarde

Geeft een mogelijke numerieke waarde van variabele *Var* waarvoor het lokale minimum van *Uitdr* optreedt.

Als u *ondergrens* en *bovangrens* opgeeft, zoekt de functie binnen het gesloten interval $[ondergrens,bovangrens]$ naar het lokale minimum.

nInt(Uitdr1, Var, Onder, Boven)⇒uitdrukking

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

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 $Onder < Var < Boven$.

nInt()Catalogus > 

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.

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

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

nom()Catalogus > **nom**

$(\text{effectiefPercentage}, CpY) \Rightarrow \text{waarde}$

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

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. 47.

nor (noch)  **toetsen**

BooleaanseUitdr1 **nor**

BooleaanseUitdr2 levert *Booleaanse uitdrukking*

BooleaanseLijst1 **nor** *BooleaanseLijst2* levert *Booleaanse lijst*

BooleaanseMatrix1

nor *BooleaanseMatrix2* levert *Booleaanse matrix*

nor (noch)

  **toetsen**

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 getal1***nor***Geheel 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*

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

norm(*Vector*)⇒*uitdrukking*

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

Geeft de Frobenius-norm.

$\text{norm}\left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$	2.23607
--	---------

normCdf()

Catalogus > 

normCdf(*ondergrens*,*bovengrens*, μ , σ)⇒*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* = -9E999.

normPdf()

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)

not *BooleaanseUitdr* $\Rightarrow$ *Booleaanse uitdrukking*

Geeft waar, onwaar of een vereenvoudigde vorm van het argument.

not *Geheel getal1* $\Rightarrow$ *geheel getal*

Geeft het één-complement van een reëel geheel getal. Intern wordt *Geheel getal1* 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).

not (2 $\geq$ 3)	true
not 0hB0 $\blacktriangleright$ Base16	0hFFFFFFFFFFFFFF4F
not not 2	2

In de Hex-grondtalmodus:

Belangrijk: nul, niet de letter O.

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

In de Bin-grondtalmodus:

0b100101 $\blacktriangleright$ Base10	37
not 0b100101	0b11111111111111111111111111111111 $\blacktriangleright$
not 0b100101 $\blacktriangleright$ Base10	-38

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

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. 17.

nPr()

nPr(Waarde1, Waarde2)⇒uitdrukking

Voor geheel getal *Waarde1* en *Waarde2* met $Waarde1 \geq Waarde2 \geq 0$, is **nPr()** het aantal permutaties van *Waarde1* dingen die met *Waarde2* tegelijk genomen zijn.

$nPr(z,3) z=5$	60
$nPr(z,3) z=6$	120
$nPr(\{5,4,3\},\{2,4,2\})$	$\{20,24,6\}$
$nPr\left(\begin{bmatrix} 6 & 5 \\ 4 & 3 \end{bmatrix}, \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}\right)$	$\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$

nPr(Waarde, 0)⇒1

nPr(Waarde, negGeheel getal)⇒ 1/ ((Waarde+1) · (Waarde+2)... (Waarde-negGeheel getal))

nPr(Waarde, posGeheel getal)⇒ Waarde · (Waarde-1)... (Waarde-posGeheel getal+1)

nPr(Waarde, nietGeheel getal)⇒Waarde! / (Waarde-nietGeheel getal)!

nPr(Lijst1, Lijst2)⇒lijst

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

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)⇒matrix

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

Geeft een matrix met permutaties op basis van de overeenkomstige elementparen in de twee matrices. De argumenten moeten matrices van dezelfde afmeting zijn.

npv()Catalogus > 

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.

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

nSolve()Catalogus > 

nSolve(*Vergelijking*,*Var*[=*Gok*]) $\Rightarrow$ *getal of fout_string*

nSolve(*Vergelijking*,*Var*
[=*Gok*],*ondergrens*) $\Rightarrow$ *getal of fout_string*

nSolve(*Vergelijking*,*Var*
[=*Gok*],*ondergrens*,*bovengrens*) $\Rightarrow$ *getal of fout_string*

nSolve(*Vergelijking*,*Var*[=*Gok*]) |
ondergrens ≤ *Var* ≤ *bovengrens* $\Rightarrow$ *getal of fout_string*

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

Opmerking: als er meerdere oplossingen zijn, dan kunt u een gok gebruiken om een bepaalde oplossing te helpen vinden.

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() 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.

$\text{nSolve}(x^2+5 \cdot x-25=9, x) x < 0$	-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 [*n*], *X* [, *Freq*] [, *Categorie*, *Opnemen*]]

OneVar [*n*], *X*1, *X*2 [*X*3 [, ... [, *X*20]]]

Berekent statistieken voor één variabele op maximaal 20 lijsten. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 160).

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. 232.

Uitvoervariabele	Beschrijving
stat. $\bar{X}$	Gemiddelde van de x-waarden
stat. Σx	Som van de x-waarden
stat. Σx^2	Som van de x^2 -waarden
stat.sx	Steekproef-standaarddeviatie van x
stat. x	Populatie-standaarddeviatie van x
stat.n	Aantal gegevens
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.SSX	Som van de kwadraten van de afwijkingen ten opzichte van het gemiddelde van x

or (of)

BooleaanseUitdr1 **or** *BooleaanseUitdr2*
levert *Booleaanse uitdrukking*

BooleaanseLijst1 **or** *BooleaanseLijst2*
levert *Booleaanse lijst*

BooleaanseMatrix1
or *BooleaanseMatrix2* levert
Booleaanse matrix

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

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

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

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 minteken. 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.

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).

In de Hex-grondtalmodus:

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

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.

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. 17.

Opmerking: zie **xor**.

ord()

ord(*String*)⇒*geheel getal*

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

ord(*Lijst1*)⇒*lijst*

char{104}	"h"
-----------	-----

ord(char{24})	24
---------------	----

Geeft de numerieke code van het eerste teken in tekenreeks *String*, of een lijst van de eerste tekens van elk lijstelement.

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

P**P►Rx()**

P►Rx(*rUitdr*, *θUitdr*)⇒*uitdrukking*

In de hoekmodus Radialen:

P►Rx(*rLijst*, *θLijst*)⇒*lijst*

P►Rx(4,60°)	2.
-------------	----

P►Rx(*rMatrix*, *θMatrix*)⇒*matrix*

P►Rx($\left\{ -3, 10, 1.3 \right\}, \left\{ \frac{\pi}{3}, \frac{\pi}{4}, 0 \right\}$)	$\left\{ -1.5, 7.07107, 1.3 \right\}$
--	---------------------------------------

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.

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

P►Ry(*rWaarde*, *θWaarde*)⇒*waarde*

In de hoekmodus Radialen:

P►Ry(*rLijst*, *θLijst*)⇒*lijst*

P►Ry(*4,60°*) 3.4641

P►Ry(*rMatrix*, *θMatrix*)⇒*matrix*

P►Ry $\left\{ \left\{ -3, 10, 1.3 \right\}, \left\{ \frac{\pi}{3}, \frac{-\pi}{4}, 0 \right\} \right\}$
 $\left\{ -2.59808, -7.07107, 0 \right\}$

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.

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

PassErr

PassErr

Zie voor een voorbeeld van **PassErr** Voorbeeld 2 onder het commando **Try** (pag. 172).

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. 24 en **Try**, pag. 172.

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()

piecewise(*Uitdr1* [, *Cond1* [, *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 $p(x) = \begin{cases} x, & x > 0 \\ \text{undef}, & x \leq 0 \end{cases}$ Done

$p(1)$	1
$p(-1)$	undef

Opmerking: zie ook **Stuksgewijs gedefinieerde functie-template**, pag. 3.

poissCdf()

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()

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 .

Vector ► Polar

$[1 \ 3.] \blacktriangleright \text{Polar}$	$[3.16228 \ \angle 71.5651]$
---	------------------------------

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. 132.

complexeWaarde ►Polar

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door $\text{e} \blacktriangleright \text{Polar}$ 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.

In de hoekmodus Radialen:

$(3+4 \cdot i) \blacktriangleright \text{Polar}$	$e^{0.927295 \cdot i} \cdot 5$
$\left(4 \angle \frac{\pi}{3}\right) \blacktriangleright \text{Polar}$	$e^{1.0472 \cdot i} \cdot 4$

In de hoekmodus Decimale graden:

$(4 \cdot i) \blacktriangleright \text{Polar}$	$(4 \angle 100.)$
--	-------------------

In de hoekmodus Graden

$(3+4 \cdot i) \blacktriangleright \text{Polar}$	$(5 \angle 53.1301)$
--	----------------------

polyEval()

polyEval(Lijst1, Uitdr1) ⇒ uitdrukking

$\text{polyEval}(\{1, 2, 3, 4\}, 2)$	26
--------------------------------------	----

polyEval(Lijst1, Lijst2) ⇒ uitdrukking

$\text{polyEval}(\{1, 2, 3, 4\}, \{2, -7\})$	$\{26, -262\}$
--	----------------

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.

polyRoots(Poly, Var) ⇒ lijst

polyRoots(LijstVanCoëff) ⇒ 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 in uitgewerkte vorm met één variabele zijn. Gebruik geen niet-uitgewerkte vormen zoals $y^2 \cdot y + 1$ of $x \cdot x + 2 \cdot x + 1$

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

Opmerking: zie ook **cPolyRoots()**, pag. 33.

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

PowerReg

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. 160).

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 numerieke of tekst-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. 232).

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

Prgm

Blok

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.

Bereken GCD (grootste gemene deler) en geef tussenresultaten weer.

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

$$\text{proggcd}(4560, 450)$$

450 60

60 30

30 0

GCD=30

Done

prodSeq()Zie $\Pi()$, pag. 204.**Product (Pi)**Zie $\Pi()$, pag. 204.**product()**Catalogus > 

product(*Lijst*[, *Start*[,
Eind]]) $\Rightarrow$ uitdrukking

Geeft het product van de elementen in *Lijst*. *Start* en *Eind* zijn optioneel. Ze specificeren een bereik van elementen.

product(*Matrix1*[, *Start*[,
Eind]]) $\Rightarrow$ matrix

Geeft een rijvector met de producten van de elementen in de kolommen van *Matrix1*. *Start* en *Eind* zijn optioneel. Ze specificeren een bereik van rijen.

Lege elementen worden genegeerd.
Voor meer informatie over lege elementen, zie pag. 232.

$\text{product}(\{1, 2, 3, 4\})$	24
----------------------------------	----

$\text{product}(\{4, 5, 8, 9\}, 2, 3)$	40
--	----

$\text{product}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}\right)$	$[28 \ 80 \ 162]$
--	-------------------

$\text{product}\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, 1, 2\right)$	$[4 \ 10 \ 18]$
--	-----------------

propFrac()Catalogus > 

propFrac(*Waarde1*[, *Var*]) $\Rightarrow$ waarde

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.

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

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

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.

U kunt de functie **propFrac()** gebruiken om gemengde breuken te representeren en om het optellen en aftrekken van gemengde breuken te demonstreren.

$\text{propFrac}\left(\frac{11}{7}\right)$	$1 + \frac{4}{7}$
$\text{propFrac}\left(3 + \frac{1}{11} + 5 + \frac{3}{4}\right)$	$8 + \frac{37}{44}$
$\text{propFrac}\left(3 + \frac{1}{11} - \left(5 + \frac{3}{4}\right)\right)$	$-2 - \frac{29}{44}$

Q

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 bovendriehoeks.

Optioneel wordt elk matricelement 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.

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 m1,qm,rm	
qm	$\begin{bmatrix} 0.123091 & 0.904534 & 0.408248 \\ 0.492366 & 0.301511 & -0.816497 \\ 0.86164 & -0.301511 & 0.408248 \end{bmatrix}$
rm	$\begin{bmatrix} 8.12404 & 9.60114 & 11.0782 \\ 0. & 0.904534 & 1.80907 \\ 0. & 0. & 0. \end{bmatrix}$

- 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:

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

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.

QuadReg

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. 160).

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 numerieke of tekst-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. 232).

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

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. 160).

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 numerieke of tekst-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. 232).

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

R ► Pθ()

R ► Pθ (*xWaarde*, *yWaarde*) ⇒ *waarde*

R ► Pθ (*xLijst*, *yLijst*) ⇒ *lijst*

R ► Pθ (*xMatrix*, *yMatrix*) ⇒ *matrix*

In de hoekmodus Graden:

$$\text{R ► P}\theta(2,2) \quad 45.$$

Geeft de equivalente θ-coördinaat van het (x,y)-paar argumenten.

In de hoekmodus Decimale graden:

$$\text{R ► P}\theta(2,2) \quad 50.$$

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

In de hoekmodus Radialen:

$$\text{R ► P}\theta(3,2) \quad 0.588003$$

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

$$\text{R ► P}\theta\left(\begin{bmatrix} 3 & -4 & 2 \end{bmatrix}, \begin{bmatrix} 0 & \frac{\pi}{4} & 1.5 \end{bmatrix}\right) \quad \begin{bmatrix} 0. & 2.94771 & 0.643501 \end{bmatrix}$$

R► Pr()Catalogus > **R► Pr** (*xWaarde*, *yWaarde*) ⇒ *waarde***R► Pr** (*xLijst*, *yLijst*) ⇒ *lijst***R► Pr** (*xMatrix*, *yMatrix*) ⇒ *matrix*

Geeft de equivalente r-coördinaat van het (*x*,*y*)-paar argumenten.

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

In de hoekmodus Radialen:

R►Pr(3,2)	3.60555
R►Pr($\left[\begin{matrix} 3 & -4 & 2 \end{matrix} \right], \left[\begin{matrix} 0 & \frac{\pi}{4} & 1.5 \end{matrix} \right]$)	$\left[\begin{matrix} 3 & 4.07638 & \frac{5}{2} \end{matrix} \right]$

► RadCatalogus > *Waarde*►Rad ⇒ *waarde*

Converteert het argument naar radialen.

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

In de hoekmodus Graden:

(1.5)►Rad	(0.02618) ^r
-----------	------------------------

In de hoekmodus Decimale graden:

(1.5)►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)	46.
randBin(80,0.5,3)	{ 43., 39., 41. }

randInt
(
ondergrens
,bovgrens) ⇒
uitdrukking

randInt(3,10)	3.
randInt(3,10,4)	{9.,3.,4.,7.}

randInt
(
ondergrens
,bovgrens
,AantalPogingen)
⇒ *lijst*

randInt
(
ondergrens
,bovgrens) geeft
een willekeurig
geheel getal binnen
het bereik dat wordt
bepaald door de
gehele
getallen *ondergrens*
en *bovgrens*.

randInt
(
ondergrens
,bovgrens
,AantalPogingen)
geeft een lijst met
AantalPogingen van
willekeurige gehele
getallen binnen het
opgegeven bereik.

randMat()

randMat(*aantalRijen*,
aantalKolommen) ⇒ *matrix*

Geeft een matrix met gehele getallen
tussen -9 en 9 met de gespecificeerde
afmeting.

Beide argumenten moeten
vereenvoudigen tot gehele getallen.

RandSeed 1147	<i>Done</i>		
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()	Catalogus > 							
randNorm(μ, σ) $\Rightarrow$ <i>uitdrukking</i> randNorm(μ, σ, <i>AantalPogingen</i>) $\Rightarrow$ <i>lijst</i>								
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 $[\mu-3\cdot\sigma, \mu+3\cdot\sigma]$.								
randNorm(μ, σ, <i>AantalPogingen</i>) geeft een lijst met <i>AantalPogingen</i> decimale getallen uit de gespecificeerde normale verdeling.								
	<table><tr><td>RandSeed 1147</td><td>Done</td></tr><tr><td>randNorm(0,1)</td><td>0.492541</td></tr><tr><td>randNorm(3,4.5)</td><td>-3.54356</td></tr></table>		RandSeed 1147	Done	randNorm(0,1)	0.492541	randNorm(3,4.5)	-3.54356
RandSeed 1147	Done							
randNorm(0,1)	0.492541							
randNorm(3,4.5)	-3.54356							

randPoly()	Catalogus > 					
randPoly (<i>Var</i> , <i>Orde</i>) $\Rightarrow$ <i>uitdrukking</i>						
Geeft een veelterm in <i>Var</i> van de gespecificeerde <i>Orde</i> . De coëfficiënten zijn willekeurige gehele getallen in het bereik -9 t/m 9 . De leidende coëfficiënt is niet nul.						
<i>Orde</i> moet 0–99 zijn.						
	<table><tr><td>RandSeed 1147</td><td>Done</td></tr><tr><td>randPoly(<i>x</i>,5)</td><td>$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$</td></tr></table>		RandSeed 1147	Done	randPoly(<i>x</i> ,5)	$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$
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()		Catalogus > 									
randSamp (<i>Lijst</i> , <i>AantalPogingen</i> [, <i>geenTerugl</i>]) $\Rightarrow$ <i>lijst</i>											
Geeft een lijst met een willekeurige steekproef van <i>AantalPogingen</i> uit <i>Lijst</i> met een optie voor met teruglegging (<i>geenTerugl</i> =0), of zonder teruglegging (<i>geenTerugl</i> =1). De standaardinstelling is met teruglegging.											
		<table><tr><td>Define <i>list3</i>=</td><td>{1,2,3,4,5}</td><td>Done</td></tr><tr><td>Define <i>list4</i>=randSamp(<i>list3</i>,6)</td><td></td><td>Done</td></tr><tr><td><i>list4</i></td><td>{1.,3.,3.,1.,3.,1.}</td><td></td></tr></table>	Define <i>list3</i> =	{1,2,3,4,5}	Done	Define <i>list4</i> =randSamp(<i>list3</i> ,6)		Done	<i>list4</i>	{1.,3.,3.,1.,3.,1.}	
Define <i>list3</i> =	{1,2,3,4,5}	Done									
Define <i>list4</i> =randSamp(<i>list3</i> ,6)		Done									
<i>list4</i>	{1.,3.,3.,1.,3.,1.}										

RandSeed		Catalogus > 	
RandSeed <i>Getal</i>			
		RandSeed 1147	Done
		rand()	0.158206

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.

real()Catalogus >

real(*Waarde1*) $\Rightarrow$ *waarde*

$\text{real}(2+3 \cdot i)$ 2

Geeft het reële deel van het argument.

real(*Lijst1*) $\Rightarrow$ *lijst*

$\text{real}(\{1+3 \cdot i, 3, i\})$ {1,3,0}

Geeft de reële delen van alle elementen.

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

$\text{real}\left(\begin{bmatrix} 1+3 \cdot i & 3 \\ 2 & i \end{bmatrix}\right)$ $\begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix}$

Geeft de reële delen van alle elementen.

► RectCatalogus >

Vector ► **Rect**

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

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

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. 121.

complexeWaarde ► **Rect**

In de hoekmodus Radialen:

Geeft *complexeWaarde* weer in rechthoekige vorm $a+bi$. De *complexeWaarde* kan elke complexe vorm hebben. Een $re^{i\theta}$ -invoer veroorzaakt echter een fout in de hoekmodus Graden.

$$\left(4 \cdot e^{\frac{\pi}{3}}\right) \blacktriangleright \text{Rect} \quad 11.3986$$

$$\left(\left(4 \angle \frac{\pi}{3}\right)\right) \blacktriangleright \text{Rect} \quad 2.+3.4641 \cdot i$$

Opmerking: u moet haakjes gebruiken voor een invoer ($r \angle \theta$) in poolcoördinaten.

In de hoekmodus Decimale graden:

$$\left((1 \angle 100)\right) \blacktriangleright \text{Rect} \quad i$$

In de hoekmodus Graden:

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

Opmerking: om $\angle$ te typen selecteert u dit uit de symbolenlijst in de Catalogus.

ref()

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

Geeft de rij-echelonvorm van *MatrixI*.

Optioneel wordt elk matrixelement 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.

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

- 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 *a*niet 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 $a=0$.

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.

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

Opmerking: zie ook `rref()`, page 144.

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

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
```

**StatusVar
waarde****Status***statusVar*
=2meetmodus staat. 
De toepassing Vernier
DataQuest™ is niet gestart.*statusVar*
=3De toepassing Vernier
DataQuest™ is gestart, maar
u hebt geen sondes
aangesloten.

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()**Catalogus > **remain(Waarde1, Waarde2) ⇒ waarde****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:

remain(x,0) x

remain(x,y) $x - y \cdot \text{iPart}(x/y)$ 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. 101.

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

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

RequestCatalogus > **Request** *promptString*, *var*[, *ToonVlag*
[, *statusVar*]]

Definieer een programma:

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*.

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.

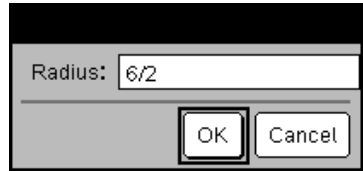
Het optionele argument *statusVar* geeft het programma een manier om te vast te stellen hoe de gebruiker het dialoogvenster heeft afgesloten. Merk op dat voor *statusVar* het argument *ToonVlag* vereist is.

- 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**.

```
Definieer request_demo()=Prgm
  Request "Straal: ",r
  Disp "Oppervlakte = ",pi*r^2
EndPrgm
```

Voer het programma uit en typ een antwoord:

request_demo()



Resultaat na selectie van **OK**:

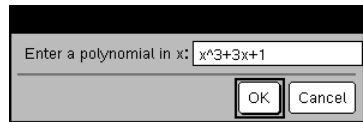
```
Straal: 6/2
Area= 28.2743
```

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**:

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.

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 137.

Echte oplossingen zijn: {-0.322185}

RequestStr

RequestStr *promptString*, var[, *ToonVlag*]

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:

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 enkele malen op **Enter**.
- **iPad®:** De app toont een instructie. U kunt blijven wachten of annuleren.

Opmerking: zie ook **Request**, page 135.

```
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()
```

Antwoord heeft 5 tekens.

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.

```
Define factorial (nn)=
Func
Local answer, counter
1 → answer
For counter, 1, nn
answer · counter → answer
EndFor
Return answer
EndFunc
```

```
factorial (3)
```

6

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.

right()

right(Lijst1[, Aantal]) $\Rightarrow$ *lijst*

<code>right({1,3,-2,4},3)</code>	<code>{3,-2,4}</code>
----------------------------------	-----------------------

Geeft het meest rechtse *Aantal* elementen in *Lijst1*.

Als u *Aantal* weglaat, wordt de hele *Lijst1* gegeven.

right(bronString[, Aantal]) $\Rightarrow$ *string*

<code>right("Hello",2)</code>	<code>"lo"</code>
-------------------------------	-------------------

Geeft het meest rechtse *Aantal* tekens in de tekenreeks *bronString*.

Als u *Aantal* weglaat, wordt de hele *bronString* gegeven.

right(Vergelijken) $\Rightarrow$ *uitdrukking*

Geeft het rechterlid van een vergelijking of ongelijkheid.

rk23 ()

rk23(Uitdr, Var, afhVar, {Var0, VarMax}, afhVar0, VarStap [,fouttol])
 $\Rightarrow$ *matrix*

Differentiaalvergelijking:

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

rk23(StelselUitdr, Var, LijstVanAfhVars, {Var0, VarMax}, LijstVanAfhVars0, VarStap[, fouttol])
 $\Rightarrow$ *matrix*

<code>rk23(0.001·y·(100-y),t,y,{0,100},10,1)</code>				
<code>0.</code>	<code>1.</code>	<code>2.</code>	<code>3.</code>	<code>4.</code>
<code>10.</code>	<code>10.9367</code>	<code>11.9493</code>	<code>13.042</code>	<code>14.2</code>

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

rk23(LijstVanUitdr, Var, LijstVanAfhVars, {Var0, VarMax}, LijstVanAfhVars0, VarStap[, fouttol])
 $\Rightarrow$ *matrix*

Dezelfde vergelijking met *fouttol* ingesteld op **1.E-6**

<code>rk23(0.001·y·(100-y),t,y,{0,100},10,1,1.E-6)</code>				
<code>0.</code>	<code>1.</code>	<code>2.</code>	<code>3.</code>	<code>4.</code>
<code>10.</code>	<code>10.9367</code>	<code>11.9495</code>	<code>13.0423</code>	<code>14.2189</code>

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 $\text{afhVar}(\text{Var0}) = \text{afhVar0}$ in het interval $[\text{Var0}, \text{VarMax}]$. 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.

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

LijstVanAfhVars0 is een lijst met beginwaarden voor afhankelijke variabelen.

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 $y2(0) = 5$

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	

Als *VarStap* wordt uitgewerkt tot een getal dat niet nul is, geldt: $\text{sign}(\text{VarStap}) = \text{sign}(\text{VarMax} - \text{Var0})$ en oplossingen worden gegeven bij $\text{Var0} + i * \text{VarStap}$ voor alle $i=0,1,2,\dots$ zodanig dat $\text{Var0} + i * \text{VarStap}$ valt binnen $[\text{var0}, \text{VarMax}]$ (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(*Waarde*) $\Rightarrow$ *wortel*

root(*Waarde1*, *Waarde2*) $\Rightarrow$ wortel

`root(Waarde)` geeft de wortel van *Waarde*.

root(*Waarde1*, *Waarde2*) geeft de *Waarde2*-wortel van *Waarde1*.

Waarde1 kan bestaan uit een reële of complexe constante met drijvende komma, een geheel getal of een complexe rationale constante.

Opmerking: zie ook N-de wortelsjabloon, pag. 2.

$\sqrt[3]{8}$	2
$\sqrt[3]{3}$	1.44225

rotate()

```
rotate(Geheel getal[,AantalRotaties]) =>
geheel getal
```

In de Bin-grondtalmodus:

Roteert de bits in een binair geheel getal. U kunt *Geheel getal* 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 getal* 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. 17.

[illegible]

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

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(*LijstI*[,*AantalRotaties*]) $\Rightarrow$ *lijst*

Geeft een kopie van *LijstI* die met *AantalRotaties* elementen naar rechts of links is gerooteerd. Verandert *LijstI* 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(*StringI*[,*AantalRotations*]) $\Rightarrow$ *string*

Geeft een kopie van *StringI* die met *AantalRotaties* tekens naar rechts of links is gerooteerd. Verandert *StringI* 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 teken naar rechts roteren).

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"

round(*WaardeI*[,*cijfers*]) $\Rightarrow$ *waarde*

round(1.234567,3)	1.235
-------------------	-------

round()

Catalogus > 

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*

Geeft een lijst van elementen, afgerond op het gespecificeerde aantal cijfers.

round(Matrix1[, cijfers]) $\Rightarrow$ *matrix*

Geeft een matrix met de elementen, afgerond op het gespecificeerde aantal cijfers.

$$\text{round}(\{\pi, \sqrt{2}, \ln(2)\}, 4) \\ \{3.1416, 1.4142, 0.6931\}$$

$$\text{round}\left(\begin{pmatrix} \ln(5) & \ln(3) \\ \pi & e^1 \end{pmatrix}, 1\right) \quad \begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$$

rowAdd()

Catalogus > 

rowAdd(Matrix1, rIndex1, rIndex2) $\Rightarrow$ *matrix*

Geeft een kopie van *Matrix1* met rij *rIndex2* vervangen door de som van de rijen *rIndex1* en *rIndex2*.

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

rowDim()

Catalogus > 

rowDim(Matrix) $\Rightarrow$ *uitdrukking*

Geeft het aantal rijen in *Matrix*.

Opmerking: zie ook **colDim()**, pag. 24.

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

rowNorm()

Catalogus > 

rowNorm(Matrix) $\Rightarrow$ *uitdrukking*

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

$$\text{rowNorm}\left(\begin{bmatrix} -5 & 6 & -7 \\ 3 & 4 & 9 \\ 9 & -9 & -7 \end{bmatrix}\right) \quad 25$$

Opmerking: alle matricelementen moeten vereenvoudigen tot getallen. Zie ook **colNorm()**, pag. 25.

rowSwap()

rowSwap(*Matrix1*, *rIndex1*, *rIndex2*)
 $\Rightarrow$ *matrix*

Geeft *Matrix1* met rijen *rIndex1* en *rIndex2* verwisseld.

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

rref()

rref(*Matrix1*[, *Tol*]) $\Rightarrow$ *matrix*

Geeft 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.

- 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:
 $5E-14 \cdot \max(\dim(Matrix1)) \cdot rowNorm(Matrix1)$

Opmerking: zie ook **ref()**, page 133.

sec()**sec(Waarde1)** $\Rightarrow$ waarde

In de hoekmodus Graden:

sec(Lijst1) $\Rightarrow$ lijstGeeft de secans van *Waarde1* of geeft een lijst met de secansen van alle elementen in *Lijst1*.**Opmerking:** het argument wordt geïnterpreteerd als een hoek in graden, decimale graden of radialen volgens de huidige hoekmodus-instelling. U kunt $^{\circ}$, G of $^{\circ}$ gebruiken om de hoekmodus tijdelijk te onderdrukken.

$\sec(45)$	1.41421
$\sec(\{1,2,3,4\})$	$\{1.00015, 1.00081, 1.00244\}$

sec⁻¹()**sec⁻¹(Waarde1)** $\Rightarrow$ waarde

In de hoekmodus Graden:

sec⁻¹(Lijst1) $\Rightarrow$ lijstGeeft de hoek waarvan de secans *Waarde1* is of geeft een lijst met de inverse secans 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 **arcsec(...)** in te typen.

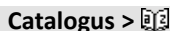
$\sec^{-1}(1)$	0.
----------------	----

In de hoekmodus Decimale graden:

$\sec^{-1}(\sqrt{2})$	50.
-----------------------	-----

In de hoekmodus Radialen:

$\sec^{-1}(\{1,2,5\})$	$\{0, 1.0472, 1.36944\}$
------------------------	--------------------------

sech()**sech(Waarde1)** $\Rightarrow$ waarde**sech(Lijst1)** $\Rightarrow$ lijstGeeft de secans hyperbolicus van *Waarde1* of geeft een lijst met de secansen hyperbolicus van de elementen in *Lijst1*.

$\operatorname{sech}(3)$	0.099328
$\operatorname{sech}(\{1,2,3,4\})$	$\{0.648054, 0.198522, 0.036619\}$

sech⁻¹(*Waarde1*) ⇒ *waarde*

sech⁻¹ (*Lijst1*) ⇒ *lijst*

Geeft de inverse secans hyperbolicus van *Waarde1* 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:

sech ⁻¹ (1)	0
sech ⁻¹ { { 1, -2, 2.1 } }	
	{ 0, 2.0944·i, 8·E-15+1.07448·i }

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.

Opmerking: U kunt de opdracht **Send** gebruiken binnen een door de gebruiker gedefinieerd programma, maar niet binnen een functie.

Opmerking: Zie ook **Get** (pag. 63), **GetStr** (pag. 70), en **eval()** (pag. 51).

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 <i>lightval</i>	Done
<i>lightval</i>	0.347922

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.

<i>n</i> :=50	50
<i>m</i> :=4	4
Send "SET SOUND eval(m·n)"	Done
<i>iostr.SendAns</i>	"SET SOUND 200"

seq()

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*

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]]) ⇒ *matrix*

Genereer de eerste 5 termen van de rij $u(n) = u(n-1)^2/2$, waarbij $u(1)=2$ en $VarStap=1$.

$\text{seqGen}\left(\frac{(u(n-1))^2}{n}, n, u, \{1, 5\}, \{2\}\right)$
$\left\{2, 2, \frac{4}{3}, \frac{4}{9}, \frac{16}{405}\right\}$

Voorbeeld waarin $Var0=2$:

$\text{seqGen}\left(\frac{u(n-1)+1}{n}, n, u, \{2, 5\}, \{3\}\right)$
$\left\{3, \frac{4}{3}, \frac{7}{12}, \frac{19}{60}\right\}$

Stelsel van twee rijen:

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.

$$\text{seqGen}\left(\left\{\frac{1}{n}, \frac{u_1^2(n-1)}{2} + u_1(n-1)\right\}, n, \{u_1, u_2\}, \{1, 5\}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 2 & 2 & \frac{3}{2} & \frac{13}{2} & \frac{19}{2} \end{bmatrix}$$

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 $u(n)=\text{Uitdr}(u, n)$, 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

Genereert een lijst met termen voor een niet-recursieve rij $u(n)=\text{Uitdr}(n)$, als volgt: Verhoogt *n* van 1 tot *nMax* met 1, werkt $u(n)$ uit voor overeenkomstige waarden van *n* met behulp van de formule *Uitdr*(*n*) en geeft de resultaten terug in een lijst.

Als *nMax* ontbreekt, dan wordt *nMax* ingesteld op 2500

Als *nMax*=0, dan wordt *nMax* ingesteld op 2500

Genereer de eerste 6 termen van de rij $u(n) = u(n-1)2/2$, waarbij $u(1)=2$.

$$\text{seqn}\left(\frac{u(n-1)}{n}, \{2\}, 6\right)$$

$$\left\{2, 1, \frac{1}{3}, \frac{1}{12}, \frac{1}{60}, \frac{1}{360}\right\}$$

$$\text{seqn}\left(\frac{1}{n^2}, 6\right)$$

$$\left\{1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \frac{1}{36}\right\}$$

Opmerking: `seqn()` roept `seqGen()` aan met $n0=1$ en $nstep=1$

setMode()

setMode(modeNaamGeheel getal, instellingGeheel getal) $\Rightarrow$ geheel getal

setMode(lijst) $\Rightarrow$ lijst met gehele getallen

Alleen geldig binnen een functie of programma.

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(lijst) stelt u in staat meerdere instellingen te veranderen. *lijst* bevat paren van gehele getallen voor modi en instellingen. **setMode(lijst)** geeft een vergelijkbare lijst waarvan de paren gehele getallen de oorspronkelijke modi en instellingen representeren.

Als u alle modusinstellingen hebt opgeslagen met **getMode(0)** $\rightarrow$ *var*, dan kunt u **setMode(var)** gebruiken om die instellingen te herstellen tot de functie of het programma wordt afgesloten. Zie **getMode()**, pag. 69.

Geef de benaderde waarde van π weer met behulp van de standaardinstelling voor Cijfers weergegeven, en geef π vervolgens weer met een instelling van Vast2. Controleer om te zien of de standaardinstelling hersteld wordt nadat het programma is uitgevoerd.

Define <i>prog1()</i> =Prgm	Done
Disp π	
setMode(1,16)	
Disp π	
EndPrgm	
<i>prog1()</i>	
	3.14159
	3.14
	Done

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.

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
Vectoropmaak	6	1=Rechthoekig, 2=Cilindrisch, 3=Bolvormig
Grondtal	7	1=Decimaal, 2=Hexadecimaal, 3=Binair

shift(*Geheel getal*
[,*#Verschuivingen*])⇒*geheel getal*

In de Bin-grondtalmodus:

```

shift(0b11111010110000110101)
                                0b1111101011000011010
shift(256,1)                    0b10000000000

```

Verschuift de bits in een binair geheel getal. U kunt *Geheel getal1* in elk grondtal invoeren; het wordt automatisch geconverteerd naar een 64-bits binaire vorm met een 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. Zie voor meer informatie ►**Base2**, pag. 17.

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 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.

0b0000000000000111101011000011010

Voegt 0 in als de meest linkse bit 0 is, of 1 als de meest linkse bit 1 is.

Dit levert op:

0b0000000000000011110101100001101-0

Het resultaat wordt weergegeven volgens de grondtal-modus. Nullen aan het begin worden niet weergegeven.

shift(*Lijst1* [,*#Verschuivingen*])⇒*lijst*

Geeft een kopie van *Lijst1* die met *#Verschuivingen* elementen naar rechts of links is verschoven. Verandert *Lijst1* niet.

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 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}

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(*String1* [,*#Verschuivingen*])⇒*string*

Geeft een kopie van *String1* die met *#Verschuivingen* tekens naar rechts of links is verschoven. Verandert *String1* 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 teken naar rechts verschuiven).

Tekens die aan het begin of eind van *string* ingevoegd worden door de verschuiving, worden ingesteld op een spatie.

shift("abcd")	" abc"
shift("abcd",-2)	" ab"
shift("abcd",1)	"bcd "

sign()

sign(*Waarde1*)⇒*waarde*

sign(*Lijst1*)⇒*lijst*

sign(*Matrix1*)⇒*matrix*

Geeft, bij reële en complexe *Waarde1*, *Waarde1* / **abs(*Waarde1*)** wanneer *Waarde1* ≠ 0.

Geeft 1 als *Waarde1* positief is.

Geeft -1 als *Waarde1* negatief is.

sign(0) geeft ±1 als de complexe opmaak-modus Reëel is; anders geeft hij zichzelf.

Als de complexe opmaak-modus Reëel is:

sign([-3 0 3])	[-1 undef 1]
----------------	--------------

sign(-3.2)	-1
sign({2,3,4,-5})	{1,1,1,-1}

sign(0) representeert de eenheidscirkel in het complexe vlak.

Geeft bij een lijst of matrix de tekens van alle elementen.

simult()

simult(*coeffMatrix*, *constVector*[, *Tol*]) $\Rightarrow$ *matrix*

Geeft een kolomvector die de oplossingen voor een stelsel lineaire vergelijkingen bevat.

Opmerking: zie ook **linSolve()**, pag. 87.

coeffMatrix moet een vierkante matrix zijn die de coëfficiënten van de vergelijkingen bevat.

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*

Los op naar x en y:

$$x + 2y = 1$$

$$3x + 4y = -1$$

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

De oplossing is $x=-3$ en $y=2$.

Los op:

$$ax + by = 1$$

$$cx + dy = 2$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \text{matx1} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\text{simult}\left(\text{matx1}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) \quad \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

Los op:

$$x + 2y = 1$$

$$3x + 4y = -1$$

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.

$$x + 2y = 2$$

$$3x + 4y = -3$$

$$\text{simult}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ -1 & -3 \end{bmatrix}\right) \quad \begin{bmatrix} -3 & -7 \\ 2 & \frac{9}{2} \end{bmatrix}$$

Voor het eerste stelsel: $x=-3$ en $y=2$. Voor het tweede stelsel: $x=-7$ en $y=9/2$.

sin()

 -toets

sin(WaardeI) ⇒ waarde

sin(LijstI) ⇒ lijst

sin(WaardeI) geeft de sinus van het argument.

sin(LijstI) 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 °, G of R gebruiken om de hoekmodusinstelling tijdelijk te onderdrukken.

sin

(vierkanteMatrixI) ⇒ vierkanteMatrix

Geeft de matrixsinus van *vierkanteMatrixI*. Dit is niet hetzelfde als het berekenen van de sinus 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 Graden:

$\sin\left(\left(\frac{\pi}{4}\right)^{\circ}\right)$	0.707107
$\sin(45)$	0.707107
$\sin(\{0,60,90\})$	$\{0,0.866025,1.\}$

In de hoekmodus Decimale graden:

$\sin(50)$	0.707107
------------	----------

In de hoekmodus Radialen:

$\sin\left(\frac{\pi}{4}\right)$	0.707107
$\sin(45^{\circ})$	0.707107

In de hoekmodus Radialen:

$\sin\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right)$	$\begin{bmatrix} 0.9424 & -0.04542 & -0.031999 \\ -0.045492 & 0.949254 & -0.020274 \\ -0.048739 & -0.00523 & 0.961051 \end{bmatrix}$
---	--

$\sin^{-1}()$



$\sin^{-1}(\text{WaardeI}) \Rightarrow \text{waarde}$

In de hoekmodus Graden:

$\sin^{-1}(\text{LijstI}) \Rightarrow \text{lijst}$

$\sin^{-1}(1)$ 90.

$\sin^{-1}(\text{WaardeI})$ geeft de hoek waarvan de waarde *WaardeI* is.

In de hoekmodus Decimale graden:

$\sin^{-1}(\text{LijstI})$ geeft een lijst van de inverse sinussen van elk element in *LijstI*.

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

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

In de hoekmodus Radialen:

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

$\sin^{-1}(\{0,0.2,0.5\})$ $\{0.,0.201358,0.523599\}$

$\sin^{-1}(\text{vierkanteMatrixI}) \Rightarrow \text{vierkanteMatrix}$

In de hoekmodus Radialen en rechthoekige complexe opmaak-modus:

Geeft de inverse matrixsinus van *vierkanteMatrixI*. Dit is niet hetzelfde als het berekenen van de inverse sinus van elk element. Zie voor informatie over de berekeningsmethode **cos()**.

$\sin^{-1}\left(\begin{bmatrix} 1 & 5 \\ 4 & 2 \end{bmatrix}\right)$
 $\begin{bmatrix} -0.174533-0.12198 \cdot i & 1.74533-2.35591 \cdot i \\ 1.39626-1.88473 \cdot i & 0.174533-0.593162 \cdot i \end{bmatrix}$

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

$\sinh()$

Catalogus >

$\sinh(\text{GetalI}) \Rightarrow \text{waarde}$

$\sinh(1.2)$ 1.50946

$\sinh(\text{LijstI}) \Rightarrow \text{lijst}$

$\sinh(\{0,1.2,3.\})$ $\{0,1.50946,10.0179\}$

$\sinh(\text{WaardeI})$ geeft de sinus hyperbolicus van het argument.

$\sinh(\text{LijstI})$ geeft een lijst met de sinus hyperbolicus van elk element in *LijstI*.

In de hoekmodus Radialen:

$\sinh(\text{vierkanteMatrixI}) \Rightarrow \text{vierkanteMatrix}$

sinh()

Catalogus > 

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

$$\sinh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$

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

sinh⁻¹()

Catalogus > 

sinh⁻¹(Waarde1)⇒waarde

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

sinh⁻¹(Lijst1)⇒lijst

$$\sinh^{-1}(\{0,2,1,3\}) \quad \{0,1.48748,1.81845\}$$

sinh⁻¹(Waarde1) geeft de inverse sinus hyperbolicus van het argument.

sinh⁻¹(Lijst1) geeft een lijst met de inverse sinus hyperbolicus van elk element in *Lijst1*.

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

sinh⁻¹(vierkanteMatrix1)⇒vierkanteMatrix

In de hoekmodus Radialen:

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

$$\sinh^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} 0.041751 & 2.15557 & 1.1582 \\ 1.46382 & 0.926568 & 0.112557 \\ 2.75079 & -1.5283 & 0.57268 \end{bmatrix}$$

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

SinReg

Catalogus > 

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. 160).

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 numerieke of tekst-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. 232).

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
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>

SortA

Catalogus >

SortA *Lijst1*[, *Lijst2*] [, *Lijst3*] ...

$\{2,1,4,3\} \rightarrow list1$

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

SortA *Vector1*[, *Vector2*] [, *Vector3*] ...

SortA *list1*

Done

Sorteert de elementen van het eerste argument in oplopende volgorde.

list1

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

Sorteert de elementen van het eerste argument in oplopende volgorde.

$\{4,3,2,1\} \rightarrow list2$

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

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.

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.

list1

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

Voor meer informatie over lege elementen, zie pag. 232.

SortD

Catalogus >

SortD *Lijst1*[, *Lijst2*] [, *Lijst3*] ...

$\{2,1,4,3\} \rightarrow list1$

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

SortD *Vector1*[, *Vector2*] [, *Vector3*] ...

$\{1,2,3,4\} \rightarrow list2$

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

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.

list1

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

Voor meer informatie over lege elementen, zie pag. 232.

list2

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

Vector ►Sphere

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

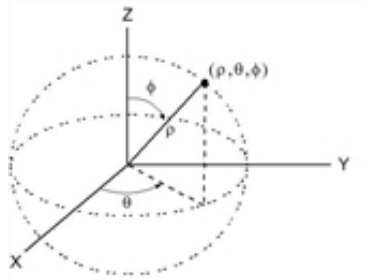
Geeft de rij- of de kolomvector in bolvorm weer [ρ $\angle \theta$ $\angle \phi$].

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.

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \text{►Sphere} \\ \left[3.74166 \quad \angle 1.10715 \quad \angle 0.640522 \right]$$

$$\begin{pmatrix} 2 & \angle \frac{\pi}{4} & 3 \end{pmatrix} \text{►Sphere} \\ \left[3.60555 \quad \angle 0.785398 \quad \angle 0.588003 \right]$$



sqrt()

sqrt(Waarde1)⇒waarde

$$\sqrt{4} \qquad 2$$

sqrt(Lijst1)⇒lijst

$$\sqrt{\{9,2,4\}} \qquad \{3,1.41421,2\}$$

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

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 $xlist,ylist,1$: stat.results	
"Title"	"Linear Regression (mx+b)"
"RegEqn"	"m*x+b"
"m"	3.2
"b"	1.2
"r ² "	0.996109
"r"	0.998053
"Resid"	"{...}"
stat.values	"Linear Regression (mx+b)"
	"m*x+b"
	3.2
	1.2
	0.996109
	0.998053
	"{-0.4,0.4,0.2,0.,-0.2}"

stat.a	stat.dfDenom	stat.MedianY	stat.Q3X	stat.SSBlock
stat.AdjR ²	stat.dfBlock	stat.MEPred	stat.Q3Y	stat.SSCol
stat.b	stat.dfCol	stat.MinX	stat.r	stat.SSX
stat.b0	stat.dfError	stat.MinY	stat.r ²	stat.SSY
stat.b1	stat.dfInteract	stat.MS	stat.RegEqn	stat.SSError
stat.b2	stat.dfReg	stat.MSBlock	stat.Resid	stat.SSInteract
stat.b3	stat.dfNumer	stat.MSCol	stat.ResidTrans	stat.SSReg
stat.b4	stat.dfRow	stat.MSError	stat.ox	stat.SSRow
stat.b5	stat.DW	stat.MSInteract	stat.oy	stat.tList
stat.b6	stat.e	stat.MSReg	stat.ox1	stat.UpperPred
stat.b7	stat.ExpMatrix	stat.MSRow	stat.ox2	stat.UpperVal
stat.b8	stat.F	stat.n	stat.Σx	stat.̄x
stat.b9	stat.FBlock	stat.̂p	stat.Σx ²	stat.̄x1
stat.b10	stat.Fcol	stat.̂p1	stat.Σxy	stat.̄x2
stat.bList	stat.FInteract	stat.̂p2	stat.Σy	stat.̄xDiff

stat. χ^2	stat.FreqReg	stat. $\hat{p}$ Diff	stat. Σy^2	stat. $\bar{X}$ List
stat.c	stat.Frow	stat.PList	stat.s	stat.XReg
stat.CLower	stat.Leverage	stat.PVal	stat.SE	stat.XVal
stat.CLowerList	stat.LowerPred	stat.PValBlock	stat.SEList	stat.XValList
stat.CompList	stat.LowerVal	stat.PValCol	stat.SEPred	stat. $\bar{y}$
stat.CompMatrix	stat.m	stat.PValInteract	stat.sResid	stat. $\hat{y}$
stat.CookDist	stat.MaxX	stat.PValRow	stat.SEslope	stat. $\hat{y}$ List
stat.CUpper	stat.MaxY	stat.Q1X	stat.sp	stat.YReg
stat.CUpperList	stat.ME	stat.Q1Y	stat.SS	
stat.d	stat.MedianX			

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($\{1,2,5,-6,3,-2\}$)	3.59398
stDevPop($\{1.3,2.5,-6.4\},\{3,2,5\}$)	4.11107

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. 232.

stDevPop(*Matrix1*[,
freqMatrix]) $\Rightarrow$ *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. 232.

$$\text{stDevPop} \left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{bmatrix} \right) = \begin{bmatrix} 3.26599 & 2.94392 & 1.63299 \end{bmatrix}$$

$$\text{stDevPop} \left(\begin{bmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{bmatrix}, \begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix} \right) = \begin{bmatrix} 2.52608 & 5.21506 \end{bmatrix}$$

stDevSamp()

stDevSamp(*Lijst*[,
freqLijst]) $\Rightarrow$ *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. 232.

$$\text{stDevSamp}(\{1, 2, 5, -6, 3, -2\}) = 3.937$$

$$\text{stDevSamp}(\{1.3, 2.5, -6.4\}, \{3, 2, 5\}) = 4.33345$$

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. 232.

```
stDevSamp( $\begin{pmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ 5 & 7 & 3 \end{pmatrix}$ )
[4. 3.60555 2.]

stDevSamp( $\begin{pmatrix} -1.2 & 5.3 \\ 2.5 & 7.3 \\ 6 & -4 \end{pmatrix}$ ,  $\begin{bmatrix} 4 & 2 \\ 3 & 3 \\ 1 & 7 \end{bmatrix}$ )
[2.7005 5.44695]
```

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:=0 0
Define prog1() $\Rightarrow$ Prgm Done
  For i,1,10,1
  If i=5
  Stop
  EndFor
  EndPrgm
prog1() Done
i 5
```

Storezie $\rightarrow$ (store), pag. 213.**string()**Catalogus > 

string(*Uitdr*) $\Rightarrow$ *string*

Vereenvoudigt *Uitdr* en geeft het resultaat als een tekenreeks.

```
string(1.2345) "1.2345"
string(1+2) "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.

$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \rightarrow m1$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
<code>subMat(m1,2,1,3,2)</code>	$\begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$
<code>subMat(m1,2,2)</code>	$\begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$

Sum (Sigma)Zie $\Sigma()$, pag. 205.**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. 232.

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. 232.

<code>sum({1,2,3,4,5})</code>	15
<code>sum({a,2*a,3*a})</code>	"Error: Variable is not defined"
<code>sum(seq(n,n,1,10))</code>	55
<code>sum({1,3,5,7,9},3)</code>	21

<code>sum($\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$)</code>	$\begin{bmatrix} 5 & 7 & 9 \end{bmatrix}$
<code>sum($\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$)</code>	$\begin{bmatrix} 12 & 15 & 18 \end{bmatrix}$
<code>sum($\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, 2, 3)$</code>	$\begin{bmatrix} 11 & 13 & 15 \end{bmatrix}$

sumIf(*Lijst*,*Criteria*[,
SomLijst]) $\Rightarrow$ waarde

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. 232.

Opmerking: zie ook **countIf()**, pag. 32.

$\text{sumIf}(\{1,2,e,3,\pi,4,5,6\}, 2.5 < ? < 4.5)$	12.859874482
$\text{sumIf}(\{1,2,3,4\}, 2 < ? < 5, \{10,20,30,40\})$	70

system(*Waarde1* [, *Waarde2* [, *Waarde3* [, ...]]])

Geeft een stelsel vergelijkingen, in de vorm van een lijst. U kunt ook een stelsel creëren met behulp van een template.

T

T (transponeren)

Matrix1 **T** $\Rightarrow$ *matrix*

Geeft de complex geconjugeerde transponering van *Matrix1*.

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

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

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

tan()

tan(*Waarde1*) $\Rightarrow$ *waarde*

In de hoekmodus Graden:

tan(*Lijst1*) $\Rightarrow$ *lijst*

tan(*Waarde1*) geeft de tangens van het argument.

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 °, G of r gebruiken om de instelling van de hoekmodus tijdelijk te onderdrukken.

$\tan\left(\left(\frac{\pi}{4}\right)^r\right)$	1.
$\tan(45)$	1.
$\tan(\{0,60,90\})$	$\{0.,1.73205,undef\}$

In de hoekmodus Decimale graden:

$\tan\left(\left(\frac{\pi}{4}\right)^r\right)$	1.
$\tan(50)$	1.
$\tan(\{0,50,100\})$	$\{0.,1.,undef\}$

In de hoekmodus Radialen:

$\tan\left(\frac{\pi}{4}\right)$	1.
$\tan(45^\circ)$	1.
$\tan\left(\left\{\pi, \frac{\pi}{3}, \pi, \frac{\pi}{4}\right\}\right)$	$\{0.,1.73205,0.,1.\}$

tan()



tan

(*vierkanteMatrix1*) $\Rightarrow$ *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(*Waarde1*) $\Rightarrow$ *waarde*

tan⁻¹(*Lijst1*) $\Rightarrow$ *lijst*

tan⁻¹(*Waarde1*) geeft de hoek waarvan de tangens *Waarde1* is.

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*) $\Rightarrow$ *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:

$$\tan^{-1}(1) \quad 45$$

In de hoekmodus Decimale graden:

$$\tan^{-1}(1) \quad 50$$

In de hoekmodus Radialen:

$$\tan^{-1}(\{0,0.2,0.5\}) \quad \{0,0.197396,0.463648\}$$

In de hoekmodus Radialen:

$$\tan^{-1} \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} \begin{bmatrix} -0.083658 & 1.26629 & 0.62263 \\ 0.748539 & 0.630015 & -0.070012 \\ 1.68608 & -1.18244 & 0.455126 \end{bmatrix}$$

tanh(*WaardeI*)⇒*waarde* $\tanh(1.2)$ 0.833655**tanh**(*LijstI*)⇒*lijst* $\tanh(\{0,1\})$ {0.,0.761594}**tanh**(*WaardeI*) geeft de tangens hyperbolicus van het argument.**tanh**(*LijstI*) geeft een lijst met de tangens hyperbolicus van elk element in *LijstI*.**tanh****(vierkanteMatrixI)**⇒*vierkanteMatrix*

Geeft de matrixtangens hyperbolicus van *vierkanteMatrixI*. Dit is niet hetzelfde als het berekenen van de 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:

$$\tanh \begin{pmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{pmatrix} = \begin{bmatrix} -0.097966 & 0.933436 & 0.425972 \\ 0.488147 & 0.538881 & -0.129382 \\ 1.28295 & -1.03425 & 0.428817 \end{bmatrix}$$

tanh⁻¹(*WaardeI*)⇒*waarde*

In rechthoekige complexe opmaak:

tanh⁻¹(*LijstI*)⇒*lijst* $\tanh^{-1}(0)$ 0.**tanh⁻¹**(*WaardeI*) geeft de inverse tangens hyperbolicus van het argument. $\tanh^{-1}(\{1,2,1,3\})$
{undef,0.518046-1.5708*i*,0.346574-1.5708*i*}**tanh⁻¹**(*LijstI*) geeft een lijst van de inverse tangens hyperbolicus van elk element in *LijstI*.Om het hele resultaat te zien drukt u op  en gebruikt u vervolgens  en  om de cursor te verplaatsen.

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

tanh⁻¹**(vierkanteMatrixI)**⇒*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()**.

In de hoekmodus Radialen en rechthoekige complexe opmaak:

tanh⁻¹()

Catalogus > 

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

$$\tanh^{-1}\left(\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}\right) = \begin{bmatrix} -0.099353+0.164058\cdot i & 0.267834-1.4908 \\ -0.087596-0.725533\cdot i & 0.479679-0.94730 \\ 0.511463-2.08316\cdot i & -0.878563+1.7901 \end{bmatrix}$$

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

tCdf()

Catalogus > 

tCdf(*ondergrens*,*bovengrens*,*df*)⇒*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 $P(X \leq \textit{bovengrens})$ stelt u *ondergrens* = -9E999.

Text

Catalogus > 

Text*promptString*[, *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 Rekenmachineschiedenis.
- 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. 135 of **RequestStr**, pag. 137.

Definieer een programma dat pauzeert om vijf verschillende toevalsgetallen in een dialoogvenster weer te geven.

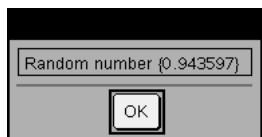
Maak binnen de template `Prgm...EndPrgm` elke regel af door op  in plaats van op **Enter** te drukken. Op het toetsenbord van de computer houdt u **Alt** ingedrukt en drukt u op **Enter**.

```
Define text_demo()=Prgm
  For i,1,5
    strinfo:="Random number "
    & string(rand(i))
    Text strinfo
  EndFor
EndPrgm
```

Opmerking: u kunt dit commando binnen een door de gebruiker gedefinieerd programma gebruiken, maar niet binnen een functie.

Voer het programma uit:
`text_demo()`

Voorbeeld van een
 dialoogvenster:



TInterval *Lijst[,Freq[,CNiveau]]*

(Invoer van een gegevenslijst)

TInterval $\bar{x}, Sx, n[, CNiveau]$

(Invoer van samenvattingsstatistieken)

Berekent een *t*-betrouwbaarheidsinterval. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.resultaten* (pag. 160).

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

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 *Lijst1, Lijst2[, Freq1[, Freq2
[, CNiveau[, Gepoold]]]]*

(Invoer van een gegevenslijst)

tInterval_2Samp $\bar{x}1, sx1, n1, \bar{x}2, sx2, n2[, CNiveau
[, Gepoold]]$

(Invoer van samenvattingsstatistieken)

Berekent een *t*-betrouwbaarheidsinterval met twee steekproeven. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 160).

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. 232).

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.

tPdf(*XWaarde, df*) $\Rightarrow$ *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()

Catalogus > 

trace(vierkanteMatrix)⇒waarde

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
$a:=12$	12
$\text{trace}\left(\begin{pmatrix} a & 0 \\ 1 & a \end{pmatrix}\right)$	24

Try

Catalogus > 

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 fouterstel kan uitvoeren. Zie "Foutcodes en meldingen", pag. 242 voor een lijst met foutcodes.

blok1 en *blok2* kunnen 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.

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.

$$\text{eigenvals}\left(\begin{pmatrix} -3 \\ -41 \\ 5 \end{pmatrix}, [-1 \ 2 \ -3.1]\right)$$

Define *prog1()*=Prgm

Try

z:=z+1

Disp "z incremented."

Else

Disp "Sorry, z undefined."

EndTry

EndPrgm

Done

z:=1:prog1()

z incremented.

Done

DelVar *z:prog1()*

Sorry, z undefined.

Done

Define *eigenvals(a,b)=Prgm*

© Het programmeren van *eigenvals(A,B)* geeft de eigenwaarden van *A·B* weer

Try

Disp "A=" ,a

Disp "B=" ,b

Disp " "

Opmerking: zie ook **ClrErr**, pag. 24 en **PassErr**, pag. 119.

```

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 μ_0 ,*Lijst*[,*Freq*[,*Hypoth*]]

(Invoer van een gegevenslijst)

tTest μ_0 , $\bar{x}$,*sx*,*n*[,*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. 160).

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. 232).

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

Catalogus > 

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. 160).

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. 232).

Uitvoervariabele	Beschrijving
stat.t	Standaard normale waarde berekend voor het verschil tussen de gemiddelden

Uitvoervariabele	Beschrijving
stat.PVal	Kleinste significantieniveau waarbij de nulhypothese verworpen kan worden
stat.df	Vrijheidsgraden voor de t-statistiek
stat. $\bar{x}_1$, stat. $\bar{x}_2$	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 <i>Gepoold</i> =1.

tvmFV() Catalogus >

tvmFV(*N,I,PV,Pmt,[PpY],[CpY],[PmtAt]*)⇒*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. 176. Zie ook **amortTbl()**, pag. 7.

tvml() Catalogus >

tvml(*N,PV,Pmt,FV,[PpY],[CpY],[PmtAt]*)⇒*waarde*

tvml(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. 176. Zie ook **amortTbl()**, pag. 7.

tvmN() Catalogus >

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. 176. Zie ook **amortTbl()**, pag. 7.

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. 176. Zie ook **amortTbl()**, pag. 7.

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. 176. Zie ook **amortTbl()**, pag. 7.

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

TVM-argument*	Beschrijving	Gegevenstype
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. 160).

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 numerieke of tekst-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. 232.

Uitvoervariabele	Beschrijving
stat. $\bar{x}$	Gemiddelde van de x-waarden
stat. x	Som van de x-waarden
stat. x ²	Som van de x ² -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 · 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
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(<i>Vector1</i>) $\Rightarrow$ <i>vector</i>	$\text{unitV}\left(\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}\right)$	$\begin{bmatrix} 0.408248 & 0.816497 & 0.408248 \end{bmatrix}$
Geeft een rij- of kolom-eenheidsvector, afhankelijk van de vorm van <i>Vector1</i> .	$\text{unitV}\left(\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right)$	$\begin{bmatrix} 0.267261 \\ 0.534522 \\ 0.801784 \end{bmatrix}$
<i>Vector1</i> moet een matrix met één rij of een matrix met één kolom zijn.		

unLock		Catalogus > 
unLock <i>Var1</i> [, <i>Var2</i>] [, <i>Var3</i>] ...	<i>a</i> :=65	65
unLock <i>Var</i> .	Lock <i>a</i>	Done
Ontgrendelt de gespecificeerde variabelen of variabelegroep.	getLockInfo(<i>a</i>)	1
Vergrendelde variabelen kunnen niet worden gewijzigd of gewist.	<i>a</i> :=75	"Error: Variable is locked."
	DelVar <i>a</i>	"Error: Variable is locked."
	Unlock <i>a</i>	Done
Zie Lock , pag. 91 en getLockInfo() , pag. 69.	<i>a</i> :=75	75
	DelVar <i>a</i>	Done

V

varPop()		Catalogus > 
varPop (<i>Lijst</i> [, <i>freqLijst</i>]) $\Rightarrow$ <i>uitdrukking</i>	$\text{varPop}(\{5,10,15,20,25,30\})$	72.9167
Geeft de populatievariantie van <i>Lijst</i> .		
Elk element uit <i>freqLijst</i> telt het aantal malen dat het overeenkomstige element in <i>Lijst</i> voorkomt.		
Opmerking: <i>Lijst</i> 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. 232.		

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. 232.

varSamp(MatrixI[,freqMatrix]) ⇒ matrix

Geeft een rijvector met de steekproefvariantie van elke kolom in *MatrixI*.

Elk element van *freqMatrix* telt het aantal opeenvolgende keer dat het overeenkomstige element voorkomt in *MatrixI*.

Opmerking: *MatrixI* 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. 232.

$\text{varSamp}(\{1,2,5,-6,3,-2\})$	$\frac{31}{2}$
$\text{varSamp}(\{1,3,5\},\{4,6,2\})$	$\frac{68}{33}$

$\text{varSamp}\left(\begin{bmatrix} 1 & 2 & 5 \\ -3 & 0 & 1 \\ .5 & .7 & 3 \end{bmatrix}\right)$	$[4.75 \quad 1.03 \quad 4]$
$\text{varSamp}\left(\begin{bmatrix} -1.1 & 2.2 \\ 3.4 & 5.1 \\ -2.3 & 4.3 \end{bmatrix}, \begin{bmatrix} 6 & 3 \\ 2 & 4 \\ 5 & 1 \end{bmatrix}\right)$	$[3.91731 \quad 2.08411]$

W

Wait tijdInSeconden

Stelt uitvoering uit voor de duur van *tijdInSeconden* seconden.

Om 4 seconden te wachten:

Wait 4

Om 1/2 seconde te wachten:

Wait 0.5

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 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 ()

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.

 <code>warnCodes(det([1.23456E-999]),warn)</code>	
	1.23456E-999
<code>warn</code>	{ 10029 }

Zie pag. 251 voor een lijst met waarschuwingscodes en bijbehorende berichten.

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.

when() is nuttig voor het definiëren van recursieve functies.

$\text{when}(x < 0, x + 3) x = 5$	undef
-------------------------------------	-------

$\text{when}(n > 0, n \cdot \text{factorial}(n - 1), 1) \rightarrow \text{factorial}(n)$	Done
$\text{factorial}(3)$	6
3!	6

While *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 $sum_of_recip(n)=Func$	
	Local $i,tempsum$
	$1 \rightarrow i$
	$0 \rightarrow tempsum$
	While $i \leq n$
	$tempsum + \frac{1}{i} \rightarrow tempsum$
	$i + 1 \rightarrow i$
	EndWhile
	Return $tempsum$
	EndFunc
	Done
$sum_of_recip(3)$	$\frac{11}{6}$

X

*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.

Opmerking: zie or, pag. 116.

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

true xor true	false
5>3 xor 3>5	true

In de Hex-grondtalmodus:

Belangrijk: nul, niet de letter O.

0h7AC36 xor 0h3D5F	0h79169
--------------------	---------

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 geretournerde 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. 17.

Opmerking: zie **or**, pag. 116.

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

zInterval

zInterval $\sigma, \text{Lijst[,Freq[,CNiveau]]}$

(Invoer van een gegevenslijst)

zInterval $\sigma, \bar{x}, n [,CNiveau]$

(Invoer van samenvattingsstatistieken)

Berekent een *z*-betrouwbaarheidsinterval. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 160).

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

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_1Prop

zInterval_1Prop x, n [,CNiveau]

Berekent een z -betrouwbaarheidsinterval voor één proportie. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 160).

x is een niet-negatief geheel getal.

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

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

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. 160).

x_1 en x_2 zijn niet-negatieve gehele getallen.

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

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 σ_1, σ_2 , Lijst1, Lijst2[,Freq1[,Freq2,
[CNiveau]]]

(Invoer van een gegevenslijst)

zInterval_2Samp $\sigma_1, \sigma_2, \bar{x}_1, n_1, \bar{x}_2, n_2$ [,CNiveau]

(Invoer van samenvattingsstatistieken)

Berekent een z -betrouwbaarheidsinterval voor twee steekproeven. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 160).

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

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

Uitvoervariabele	Beschrijving
stat. $\bar{x}$ 1, stat. $\bar{x}$ 2	Steekproefgemiddelden van de gegevensverzameling uit de willekeurige normale verdeling
stat. σ x1, stat. σ 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

Catalogus > 

zTest $\mu_0, \sigma, \text{Lijst}, [\text{Freq}, \text{Hypoth}]$

(Invoer van een gegevenslijst)

zTest $\mu_0, \sigma, \bar{x}, n, [\text{Hypoth}]$

(Invoer van samenvattingsstatistieken)

Voer een z-toets uit met frequentie *freqlijst*. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 160).

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. 232).

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 $p0, x, n[, Hypoth]$

Berekent een z -toets voor één proportie. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 160).

x is een niet-negatief geheel getal.

Toets $H_0: p = p0$ tegen één van de volgende alternatieven:

Voor $H_1: p > p0$ stelt u *Hypoth*>0 in

Voor $H_1: p \neq p0$ (standaardinstelling) stelt u *Hypoth*=0 in

Voor $H_1: p < p0$ stelt u *Hypoth*<0 in

Zie voor informatie over het effect van lege elementen in een lijst "Lege elementen" (pag. 232).

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**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. 160).

$x1$ 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 $H_a: p < p_0$ stelt u *Hypoth*<0 in

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

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 $\sigma_1, \sigma_2, \text{Lijst1}, \text{Lijst2}, [\text{Freq1}], [\text{Freq2}], [\text{Hypoth}]$

(Invoer van een gegevenslijst)

zTest_2Samp $\sigma_1, \sigma_2, \bar{x}_1, n_1, \bar{x}_2, n_2, [\text{Hypoth}]$

(Invoer van samenvattingsstatistieken)

Berekent een *z*-toets voor twee steekproeven. Een samenvatting van de resultaten wordt opgeslagen in de variabele *stat.results* (pag. 160).

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 $H_1: \mu_1 > \mu_2$, *Hypoth*>0

Zie voor informatie over het effect van lege elementen in een lijst “Lege elementen” (pag. 232).

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. $\bar{x}1$, stat. $\bar{x}2$	Steekproefgemiddelde van de gegevensverzamelingen in <i>Lijst1</i> en <i>Lijst2</i>
stat.sx1, stat.sx2	Steekproefstandaarddeviaties van de gegevensverzamelingen in <i>Lijst1</i> en <i>Lijst2</i>
stat.n1, stat.n2	Grootte van de steekproeven

Symbolen

+ (optellen)		+ -toets
<i>Waarde1 + Waarde2⇒waarde</i>	56	56
Geeft de som van de twee argumenten.	56+4	60
	60+4	64
	64+4	68
	68+4	72
<i>Lijst1 + Lijst2⇒lijst</i>	$\left\{22,\pi,\frac{\pi}{2}\right\} \rightarrow l1$	$\{22,3.14159,1.5708\}$
<i>Matrix1 + Matrix2⇒matrix</i>	$\left\{10,5,\frac{\pi}{2}\right\} \rightarrow l2$	$\{10,5,1.5708\}$
Geeft een lijst (of matrix) met de som van de overeenkomstige elementen in <i>Lijst1</i> en <i>Lijst2</i> (of <i>Matrix1</i> en <i>Matrix2</i>).	$l1+l2$	$\{32,8.14159,3.14159\}$
De afmetingen van de argumenten moeten gelijk zijn.	$15+\{10,15,20\}$	$\{25,30,35\}$
	$\{10,15,20\}+15$	$\{25,30,35\}$
<i>Lijst1 + Waarde⇒lijst</i>		
Geeft een lijst met de som van <i>Waarde</i> en elk element in <i>Lijst1</i> .	$20+\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
<i>Waarde + Matrix1⇒matrix</i>		
<i>Matrix1 + Waarde⇒matrix</i>		
Geeft een matrix met <i>Waarde</i> opgeteld bij elk element op de diagonaal van <i>Matrix1</i> . <i>Matrix1</i> moet vierkant zijn.		
Opmerking: gebruik .+ (punt plus) om een uitdrukking bij elk element op te tellen.		

- (aftrekken)		- -toets
<i>Waarde1 - Waarde2⇒waarde</i>	6-2	4
Geeft <i>Waarde1</i> min <i>Waarde2</i> .	$\pi-\frac{\pi}{6}$	2.61799

–(aftrekken)



Lijst1 – *Lijst2* ⇒ *lijst*

$$\left\{ 22, \pi, \frac{\pi}{2} \right\} - \left\{ 10, 5, \frac{\pi}{2} \right\} \quad \left\{ 12, -1.85841, 0 \right\}$$

Matrix1 – *Matrix2* ⇒ *matrix*

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

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.

Waarde – *Lijst1* ⇒ *lijst*

$$15 - \{ 10, 15, 20 \} \quad \{ 5, 0, -5 \}$$

Lijst1 – *Waarde* ⇒ *lijst*

$$\{ 10, 15, 20 \} - 15 \quad \{ -5, 0, 5 \}$$

Trekt elk *Lijst1*-element af van *Waarde* of trekt *Waarde* af van elk *Lijst1*-element, en geeft een lijst met de uitkomsten.

Waarde – *Matrix1* ⇒ *matrix*

$$20 - \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

Matrix1 – *Waarde* ⇒ *matrix*

Waarde – *Matrix1* geeft een matrix van *Waarde* keer de identiteitsmatrix min *Matrix1*. *Matrix1* moet vierkant zijn.

Matrix1 – *Waarde* geeft een matrix van *Waarde* keer de eenheidsmatrix afgetrokken van *Matrix1*. *Matrix1* moet vierkant zijn.

Opmerking: gebruik .- (punt min) om een uitdrukking van elk element af te trekken.

• (vermenigvuldigen)



Waarde1 • *Waarde2* ⇒ *waarde*

$$2 \cdot 3.45 \quad 6.9$$

Geeft het product van de twee argumenten.

Lijst1 • *Lijst2* ⇒ *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*.

• (vermenigvuldigen)

✖-toets

De afmetingen van de lijsten moeten gelijk zijn.

$Matrix1 \cdot Matrix2 \Rightarrow matrix$

Geeft het matrixproduct van $Matrix1$ en $Matrix2$.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 7 & 8 \\ 7 & 8 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 42 & 48 \\ 105 & 120 \end{bmatrix}$$

Het aantal kolommen in $Matrix1$ moet gelijk zijn aan het aantal rijen in $Matrix2$.

$Waarde \cdot Lijst1 \Rightarrow lijst$

$$\pi \cdot \{4,5,6\} = \{12.5664, 15.708, 18.8496\}$$

$Lijst1 \cdot Waarde \Rightarrow lijst$

Geeft een lijst met de producten van $Waarde$ en elk element in $Lijst1$.

$Waarde \cdot Matrix1 \Rightarrow matrix$

$Matrix1 \cdot Waarde \Rightarrow matrix$

Geeft een matrix met de producten van $Waarde$ en elk element in $Matrix1$.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot 0.01 = \begin{bmatrix} 0.01 & 0.02 \\ 0.03 & 0.04 \end{bmatrix}$$

$$6 \cdot \text{identity}(3) = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{bmatrix}$$

Opmerking: gebruik $\cdot$ (punt vermenigvuldigen) om een uitdrukking met elk element te vermenigvuldigen.

/ (delen)

÷-toets

$Waarde1 / Waarde2 \Rightarrow waarde$

Geeft het quotiënt van $Waarde1$ gedeeld door $Waarde2$.

$$\frac{2}{3.45} = 0.57971$$

Opmerking: zie ook **Breuk-template**, pag. 1.

$Lijst1 / Lijst2 \Rightarrow lijst$

Geeft een lijst met de quotiënten van $Lijst1$ gedeeld door $Lijst2$.

$$\frac{\{1,2,3\}}{\{4,5,6\}} = \left\{0.25, \frac{2}{5}, \frac{1}{2}\right\}$$

De afmetingen van de lijsten moeten gelijk zijn.

$Waarde / Lijst1 \Rightarrow lijst$

$Lijst1 / Waarde \Rightarrow lijst$

$$\frac{6}{\{3,6,\sqrt{6}\}} = \{2, 1.2, 4.4949\}$$

$$\frac{\{7,9,2\}}{7 \cdot 9 \cdot 2} = \left\{\frac{1}{18}, \frac{1}{14}, \frac{1}{63}\right\}$$

/ (delen)



Geeft een lijst met de quotiënten van *Waarde* gedeeld door *Lijst1* of *Lijst1* gedeeld door *Waarde*.

Waarde / *Matrix1* $\Rightarrow$ *matrix*

$$\frac{\begin{bmatrix} 7 & 9 & 2 \end{bmatrix}}{7 \cdot 9 \cdot 2} \quad \begin{bmatrix} \frac{1}{18} & \frac{1}{14} & \frac{1}{63} \end{bmatrix}$$

Matrix1 / *Waarde* $\Rightarrow$ *matrix*

Geeft een matrix met de quotiënten van *Matrix1* / *Waarde*.

Opmerking: gebruik . / (punt gedeeld door) om een uitdrukking door elk element te delen.

^ (macht)



Waarde1 ^ *Waarde2* $\Rightarrow$ *waarde*

$$4^2 \quad 16$$

Lijst1 ^ *Lijst2* $\Rightarrow$ *lijst*

$$\{2,4,6\}^{\{1,2,3\}} \quad \{2,16,216\}$$

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.

Waarde ^ *Lijst1* $\Rightarrow$ *lijst*

$$\pi^{\{1,2,-3\}} \quad \{3.14159, 9.8696, 0.032252\}$$

Geeft *Waarde* verheven tot de macht van de elementen in *Lijst1*.

Lijst1 ^ *Waarde* $\Rightarrow$ *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 *Waarde*.

^ (macht)**[^]-toets**

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.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \quad \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ \hline 2 & 2 \end{bmatrix}$$

vierkanteMatrix1 moet een vierkante matrix zijn.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2} \quad \begin{bmatrix} 11 & -5 \\ 2 & 2 \\ \hline -15 & 7 \\ \hline 4 & 4 \end{bmatrix}$$

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.

x2 (kwadraat)**[x^2]-toets**

*Waarde1*² ⇒ *waarde*

$$4^2 \quad 16$$

Geeft het kwadraat van het argument.

$$\{2,4,6\}^2 \quad \{4,16,36\}$$

*Lijst1*² ⇒ *lijst*

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^2 \quad \begin{bmatrix} 40 & 64 & 88 \\ 49 & 79 & 109 \\ 58 & 94 & 130 \end{bmatrix}$$

Geeft een lijst met de kwadraten van de elementen in *Lijst1*.

*vierkanteMatrix1*² ⇒ *matrix*

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}^{\wedge 2} \quad \begin{bmatrix} 4 & 16 & 36 \\ 9 & 25 & 49 \\ 16 & 36 & 64 \end{bmatrix}$$

Geeft het matrixkwadraat van *vierkanteMatrix1*. Dit is niet hetzelfde als het berekenen van het kwadraat van elk element. Gebruik *.^2* om het kwadraat van elk element te berekenen.

+. (punt optellen)**[.+]-toetsen**

Matrix1 .+ *Matrix2* ⇒ *matrix*

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix} \quad \begin{bmatrix} 11 & 32 \\ 23 & 44 \end{bmatrix}$$

Waarde .+ *Matrix1* ⇒ *matrix*

$$5 .+ \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix} \quad \begin{bmatrix} 15 & 35 \\ 25 & 45 \end{bmatrix}$$

Matrix1 .+ *Matrix2* geeft een matrix met de som van elk paar overeenkomstige elementen in *Matrix1* en *Matrix2*.

Waarde .+ *Matrix1* geeft een matrix met de sommen van *Waarde* en elk element in *Matrix1*.

.- (punt aftrekken)

☐ ☐ -toetsen

Matrix1 .- *Matrix2* $\Rightarrow$ matrix

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .- \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} -9 & -18 \\ -27 & -36 \end{bmatrix}$$

Waarde .- *Matrix1* $\Rightarrow$ matrix

$$5 .- \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} -5 & -15 \\ -25 & -35 \end{bmatrix}$$

Matrix1 .- *Matrix2* geeft een matrix met het verschil tussen elk paar overeenkomstige elementen in *Matrix1* en *Matrix2*.

Waarde .- *Matrix1* geeft een matrix met de verschillen van *Waarde* en elk element in *Matrix1*.

.· (punt vermenigvuldigen)

☐ ☒ -toetsen

Matrix1 .· *Matrix2* $\Rightarrow$ matrix

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} .· \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 10 & 40 \\ 90 & 160 \end{bmatrix}$$

Waarde .· *Matrix1* $\Rightarrow$ matrix

$$5 .· \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} 50 & 100 \\ 150 & 200 \end{bmatrix}$$

Matrix1 .· *Matrix2* geeft een matrix met het product van elk paar overeenkomstige elementen in *Matrix1* en *Matrix2*.

Waarde .· *Matrix1* geeft een matrix met de producten van *Waarde* en elk element in *Matrix1*.

. / (punt delen)

☐ ☒ -toetsen

Matrix1 . / *Matrix2* $\Rightarrow$ matrix

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} . / \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} \end{bmatrix}$$

Waarde . / *Matrix1* $\Rightarrow$ matrix

$$5 . / \begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{6} & \frac{1}{8} \end{bmatrix}$$

Matrix1 . / *Matrix2* geeft een matrix met het quotiënt van elk paar overeenkomstige elementen in *Matrix1* en *Matrix2*.

Waarde . / *Matrix1* geeft een matrix met de quotiënten van *Waarde* en elk element in *Matrix1*.

= (is gelijk)

 -toets

$Uitdr1 = Uitdr2 \Rightarrow$ Booleaanse uitdrukking

$Lijst1 = Lijst2 \Rightarrow$ Booleaanse lijst

$Matrix1 = Matrix2 \Rightarrow$ Booleaanse matrix

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.

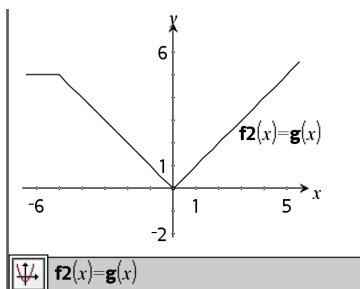
Opmerking bij het invoeren van het voorbeeld: Instructies over het invoeren van programma's met meerdere regels en functiedefinities vindt u in het hoofdstuk Rekenmachine van de handleiding van uw product.

Voorbeeldfunctie waarin wiskundige test-symbolen worden gebruikt: =, ≠, <, ≤, >, ≥

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

Done

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.

Zie het voorbeeld voor “=” (is gelijk).

$\neq$ (is niet gelijk)

  -toetsen

Bij lijsten en matrices vergelijkt dit commando element voor element.

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

$Uitdr1 > Uitdr2 \Rightarrow \text{Booleaanse uitdrukking}$

Zie het voorbeeld voor “=” (is gelijk).

$Lijst1 > Lijst2 \Rightarrow \text{Booleaanse lijst}$

$Matrix1 > Matrix2 \Rightarrow \text{Booleaanse matrix}$

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

$Uitdr1 \geq Uitdr2 \Rightarrow \text{Booleaanse uitdrukking}$

Zie het voorbeeld voor “=” (is gelijk).

$Lijst1 \geq Lijst2 \Rightarrow \text{Booleaanse lijst}$

$Matrix1 \geq Matrix2 \Rightarrow \text{Booleaanse matrix}$

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*

$5 > 3$ or $3 > 5$	true
$5 > 3 \Rightarrow 3 > 5$	false

*BooleaanseLijst1 ⇒ BooleaanseLijst2
levert Booleaanse lijst*

3 or 4	7
$3 \Rightarrow 4$	-4

*BooleaanseMatrix1 ⇒
BooleaanseMatrix2 levert Booleaanse
matrix*

$\{1, 2, 3\}$ 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*

$5 > 3$ xor $3 > 5$	true
$5 > 3 \Leftrightarrow 3 > 5$	false

*BooleaanseLijst1 ⇔ BooleaanseLijst2
levert Booleaanse lijst*

3 xor 4	7
$3 \Leftrightarrow 4$	-8

*BooleaanseMatrix1 ⇔
BooleaanseMatrix2 levert Booleaanse
matrix*

$\{1, 2, 3\}$ xor $\{3, 2, 1\}$	$\{2, 0, 2\}$
$\{1, 2, 3\} \Leftrightarrow \{3, 2, 1\}$	$\{-3, -1, -3\}$

*Geheel getal1 ⇔ Geheel getal2 levert
Geheel getal*

$\Leftrightarrow$ (logische dubbele implicatie, XNOR)

  toetsen

Geeft de ontkenning (negatie) van een XOR Booleaanse 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.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoegen door $\Leftrightarrow$ in te typen

! (faculteit)

 -toets

Waarde1! $\Rightarrow$ *waarde*

5! 120

Lijst1! $\Rightarrow$ *lijst*

$\{\{5,4,3\}\}!$ $\{120,24,6\}$

Matrix1! $\Rightarrow$ *matrix*

$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}!$ $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

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)

  -toetsen

String1 & *String2* $\Rightarrow$ *string*

"Hello " & "Nick" "Hello Nick"

Geeft een tekststring die bestaat uit *String2* toegevoegd aan *String1*.

d() (afgeleide)

Catalogus > 

d(*Uitdr1*, *Var*[,*Orde*]) |
Var=*Waarde* $\Rightarrow$ *waarde*

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

d(*Uitdr1*, *Var*[,*Orde*]) $\Rightarrow$ *waarde*

$x:=0; \frac{d}{dx}(|x|)$ undef

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

$x:=3; \frac{d}{dx}(\{x^2, x^3, x^4\})$ $\{6, 27, 108\}$

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

U moet een numerieke waarde in variabele *Var* opslaan voordat u **d()** uitwerkt, behalve als u de eerste syntax gebruikt. Zie de voorbeelden.

d() kan worden gebruikt om de eerste en tweede afgeleide in een punt numeriek te berekenen, met behulp van automatische differentiatiemethoden.

Orde moet, indien opgenomen, =1 of 2 zijn. De standaardwaarde is 1.

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

Opmerking: zie ook **Eerste afgeleide**, pag. 5 of **Tweede afgeleide**, pag. 6.

Opmerking: het algoritme **d()** kent een beperking: het werkt recursief door de niet-vereenvoudigde uitdrukking heen, door de numerieke waarde van de eerste afgeleide (en indien van toepassing de tweede) en de uitwerking van iedere subuitdrukking te berekenen, wat tot een onverwachte uitkomst kan leiden.

Zie het voorbeeld rechts. De eerste afgeleide van $x \cdot (x^2 + x)^{1/3}$ voor $x=0$ is gelijk aan 0. Omdat de eerste afgeleide van de subuitdrukking $(x^2 + x)^{1/3}$ echter onbepaald is voor $x=0$, en deze waarde gebruikt wordt om de afgeleide van de gehele uitdrukking te berekenen, geeft **d()** de uitkomst als onbepaald weer en toont een waarschuwingsbericht.

Als u deze beperking tegenkomt, verifieer de oplossing dan grafisch. U kunt ook proberen **centralDiff()** te gebruiken.

$\frac{d}{dx} \left(x \cdot (x^2 + x)^{\frac{1}{3}} \right) \Big _{x=0}$	undef
$\text{centralDiff} \left(x \cdot (x^2 + x)^{\frac{1}{3}}, x \right) \Big _{x=0}$	0.000033

$\int()$ (integraal)

Catalogus > 

$\int(\text{Uitdr1}, \text{Var}, \text{Onder}, \text{Boven}) \Rightarrow \text{waarde}$

Geeft de integraal van *Uitdr1* ten opzichte van de variabele *Var* van *Onder* tot *Boven*. Kan worden gebruikt om de bepaalde integraal numeriek te berekenen, met dezelfde methode als `nInt()`.

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

Opmerking: zie ook `nInt()`, pag. 109 en **Bepaalde integraal-template**, pag. 6.

$$\int_0^1 x^2 dx \quad 0.333333$$

$\sqrt{}$ (wortel)

  -toetsen

$\sqrt{(\text{Waarde1})} \Rightarrow \text{waarde}$

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

$$\sqrt{4} \quad 2$$
$$\sqrt{\{9,2,4\}} \quad \{3,1.41421,2\}$$

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.

$\prod()$ (prodSeq)

Catalogus > 

$\prod(\text{Uitdr1}, \text{Var}, \text{Laag}, \text{Hoog}) \Rightarrow \text{uitdrukking}$

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.

$$\prod_{n=1}^5 \left(\frac{1}{n} \right) \quad \frac{1}{120}$$
$$\prod_{n=1}^5 \left(\left\{ \frac{1}{n}, n, 2 \right\} \right) \quad \left\{ \frac{1}{120}, 120, 32 \right\}$$

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.

$\Pi(\text{Uitdr1}, \text{Var}, \text{Laag}, \text{Laag}-1) \Rightarrow 1$

$\Pi(\text{Uitdr1}, \text{Var}, \text{Laag}, \text{Hoog}) \Rightarrow 1/\Pi(\text{Uitdr1}, \text{Var}, \text{Hoog}+1, \text{Laag}-1)$ als $\text{Hoog} < \text{Laag}-1$

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.

$$\prod_{k=4}^3 \left(\frac{1}{k}\right) \quad 1$$

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \quad 6$$

$$\prod_{k=4}^1 \left(\frac{1}{k}\right) \cdot \prod_{k=2}^4 \left(\frac{1}{k}\right) \quad \frac{1}{4}$$

 $\Sigma()$ (sumSeq)

$\Sigma(\text{Uitdr1}, \text{Var}, \text{Laag}, \text{Hoog}) \Rightarrow \text{uitdrukking}$

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.

$\Sigma(\text{Uitdr1}, \text{Var}, \text{Laag}, \text{Laag}-1) \Rightarrow 0$

$\Sigma(\text{Uitdr1}, \text{Var}, \text{Laag}, \text{Hoog}) \Rightarrow -\Sigma(\text{Uitdr1}, \text{Var}, \text{Hoog}+1, \text{Laag}-1)$ als $\text{Hoog} < \text{Laag}-1$

$$\sum_{n=1}^5 \left(\frac{1}{n}\right) \quad \frac{137}{60}$$

$$\sum_{k=4}^3 (k) \quad 0$$

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.

$$\sum_{k=4}^1 \binom{k}{k} = -5$$

$$\sum_{k=4}^1 \binom{k}{k} + \sum_{k=2}^4 \binom{k}{k} = 4$$

ΣInt()

ΣInt(*NPmt1*, *NPmt2*, *N*, *I*, *PV*, [*Pmt*], [*FV*], [*PpY*], [*CpY*], [*PmtAt*], [*afgerondeWaarde*])⇒*waarde*

ΣInt(1,3,12,4,75,20000,,,12,12) -218.11

ΣInt

(*NPmt1*, *NPmt2*, *amortTable*)⇒*waarde*

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. 176.

- 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 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.

ΣInt(*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. 7.

tbl:=amortTbl(12,12,4,75,20000,,,12,12)				
0	0.	0.	20000.	
1	-79.17	-1630.69	18369.3	
2	-72.71	-1637.15	16732.2	
3	-66.23	-1643.63	15088.5	
4	-59.73	-1650.13	13438.4	
5	-53.19	-1656.67	11781.7	
6	-46.64	-1663.22	10118.5	
7	-40.05	-1669.81	8448.7	
8	-33.44	-1676.42	6772.28	
9	-26.81	-1683.05	5089.23	
10	-20.14	-1689.72	3399.51	
11	-13.46	-1696.4	1703.11	
12	-6.74	-1703.12	-0.01	
ΣInt(1,3,tbl)				-218.11

Opmerking: zie ook $\Sigma\text{Prn}()$, hieronder, en $\text{Bal}()$, pag. 16.

 $\Sigma\text{Prn}()$

$\Sigma\text{Prn}(\text{NPmt1}, \text{NPmt2}, N, I, PV, [\text{Pmt}], [\text{FV}], [\text{PpY}], [\text{CpY}], [\text{PmtAt}], [\text{afgerondeWaarde}]) \Rightarrow \text{waarde}$

$\Sigma\text{Prn}(1, 3, 12, 4.75, 20000, \dots, 12, 12) \quad -4911.47$

 ΣPrn

$(\text{NPmt1}, \text{NPmt2}, \text{amortTable}) \Rightarrow \text{waarde}$

Aflossingsfunctie die de som van de hoofdsom gedurende een gespecificeerd aantal betalingen berekent.

NPmt1 en NPmt2 definiëren de begin- en eindgrenzen van het betalingsbereik.

$N, I, PV, \text{Pmt}, FV, \text{PpY}, \text{CpY}$ en PmtAt worden beschreven in de tabel met TVM-argumenten, pag. 176.

- Als u Pmt weglaat, dan wordt de standaardwaarde $\text{Pmt}=\text{tvmPmt}(N, I, PV, FV, \text{PpY}, \text{CpY}, \text{PmtAt})$ 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.

$\Sigma\text{Prn}(\text{NPmt1}, \text{NPmt2}, \text{amortTable})$ 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 $\text{amortTbl}()$, pag. 7.

Opmerking: zie ook $\Sigma\text{Int}()$, hierboven, en $\text{Bal}()$, pag. 16.

$\text{tbl}:=\text{amortTbl}(12, 12, 4.75, 20000, \dots, 12, 12)$

0	0.	0.	20000.
1	-79.17	-1630.69	18369.3
2	-72.71	-1637.15	16732.2
3	-66.23	-1643.63	15088.5
4	-59.73	-1650.13	13438.4
5	-53.19	-1656.67	11781.7
6	-46.64	-1663.22	10118.5
7	-40.05	-1669.81	8448.7
8	-33.44	-1676.42	6772.28
9	-26.81	-1683.05	5089.23
10	-20.14	-1689.72	3399.51
11	-13.46	-1696.4	1703.11
12	-6.74	-1703.12	-0.01

$\Sigma\text{Prn}(1, 3, \text{tbl}) \quad -4911.47$

(indirectie)

  -toetsen

varNaamString

Verwijst naar de variabele met de naam *varNaamString*. Hiermee kunt u strings gebruiken om variabelenamen binnen een functie te creëren.

$xyz:=12$	12
$\# \{ "x" \& "y" \& "z" \}$	12

Creëert of verwijst naar de variabele xyz.

$10 \rightarrow r$	10
$"r" \rightarrow sI$	"r"
$\#sI$	10

Geeft de waarde van de variabele (r) waarvan de naam is opgeslagen in variabele s1.

E (wetenschappelijke notatie)

 -toets

mantisseEexponent

Voert een getal in wetenschappelijke notatie in. Het getal wordt geïnterpreteerd als *mantisse* × *10*^{*exponent*}.

23000.	23000.
23000000000.+4.1E15	4.1E15
$3 \cdot 10^4$	30000

Tip: als u een macht van 10 wilt invoeren zonder een resultaat met decimalen te veroorzaken, gebruik dan *10[^]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)

 -toets

Uitdr1g⇒uitdrukking

Lijst1g⇒lijst

Matrix1g⇒matrix

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 modus Graden, Decimale graden of Radialen:

$\cos(50^g)$	0.707107
$\cos(\{0,100^g,200^g\})$	$\{1,0,-1\}$

g (decimale graden)

1-toets

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)

1-toets

*Waarde1*r $\Rightarrow$ *waarde*

*Lijst1*r $\Rightarrow$ *lijst*

*Matrix1*r $\Rightarrow$ *matrix*

Deze functie geeft u een manier om een
hoek in radialen te specificeren terwijl u
in de modus Graden of Decimale graden
bent.

In de hoekmodus Graden:
vermenigvuldigt het argument met
 $180/\pi$.

In de hoekmodus Radialen: geeft het
argument ongewijzigd.

In de modus Decimale graden:
vermenigvuldigt het argument met
 $200/\pi$.

Tip: gebruik r als u radialen wilt forceren
in een functiedefinitie, ongeacht de
modus die de voorkeur heeft wanneer
de functie wordt gebruikt.

Opmerking: u kunt deze operator vanaf
het toetsenbord van de computer
invoeren door @r in te voeren.

In de hoekmodus Graden, Decimale graden of
Radialen:

$\cos\left(\frac{\pi}{4^r}\right)$	0.707107
$\cos\left(\left\{0^r, \left(\frac{\pi}{12}\right)^r, -(\pi)^r\right\}\right)$	{ 1, 0.965926, -1. }

° (graden)

1-toets

Waarde $I^\circ \Rightarrow$ *waarde*

Lijst $I^\circ \Rightarrow$ *lijst*

Matrix $I^\circ \Rightarrow$ *matrix*

Deze functie geeft u een manier om een hoek in graden te specificeren terwijl u in de modus 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:

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

In de hoekmodus Radialen:

$$\cos\left(\left\{0, \frac{\pi}{4}, 90^\circ, 30.12^\circ\right\}\right) \quad \{1., 0.707107, 0., 0.864976\}$$

°, ', " (graad/minuut/seconde)

ctrl  -toetsen

dd°mm'ss.ss" $\Rightarrow$ *uitdrukking*

*dd*Een positief of negatief getal

*mm*Een niet-negatief getal

*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 apostroffen ('), niet door een dubbel aanhalingsteken (").

In de hoekmodus Graden:

$$\begin{array}{r} 25^\circ 13' 17.5'' \\ 25^\circ 30' \\ \hline 25.2215 \\ \hline 51 \\ 2 \end{array}$$

∠ (hoek)

  -toetsen

[*Straal*,∠*θ*_Hoek]⇒vector

(polaire invoer)

[*Straal*,∠*θ*_Hoek,*Z*_Coördinaat]⇒vector

(cilindrische invoer)

[*Straal*,∠*θ*_Hoek,∠*θ*_Hoek]⇒vector

(bolvormige invoer)

Geeft coördinaten als een vector op basis van de modusinstelling voor vectoropmaak: rechthoekig, cilindrisch of bolvormig.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door @< in te voeren.

(*Grootte* ∠ *Hoek*)⇒*complexeWaarde*

(polaire invoer)

Voert een complexe waarde in ($r\angle\theta$) polaire vorm in. De *Hoek* wordt geïnterpreteerd volgens de huidige instelling van de hoekmodus.

In de modus Radialen en met vectoropmaak ingesteld op:

rechthoekig

$$\left[5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \\ \left[1.76777 \quad 3.06186 \quad 3.53553 \right]$$

cilindrisch

$$\left[5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \\ \left[3.53553 \quad \angle 1.0472 \quad 3.53553 \right]$$

bolvormig

$$\left[5 \quad \angle 60^\circ \quad \angle 45^\circ \right] \\ \left[5, \quad \angle 1.0472 \quad \angle 0.785398 \right]$$

In de hoekmodus Radialen en rechthoekige complexe opmaak:

$$5+3 \cdot i - \left(10 \angle \frac{\pi}{4} \right) \quad -2.07107-4.07107 \cdot i$$

_ (onderstrepingssteken als een leeg element)

Zie “Lege elementen”
pag. 232.

10^()

Catalogus > 

10^ (*WaardeI*)⇒waarde

$$10^{1.5} \quad 31.6228$$

10^ (*LijstI*)⇒lijst

Geeft 10 tot de macht van het argument.

Geeft bij een lijst 10 tot de macht van de elementen in *LijstI*.

10^()

Catalogus > 

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()**.

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

$$10^{\begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 6 & -2 & 1 \end{bmatrix}} = \begin{bmatrix} 1.14336\text{E}7 & 8.17155\text{E}6 & 6.67589\text{E}6 \\ 9.95651\text{E}6 & 7.11587\text{E}6 & 5.81342\text{E}6 \\ 7.65298\text{E}6 & 5.46952\text{E}6 & 4.46845\text{E}6 \end{bmatrix}$$

^-1(omgekeerde)

Catalogus > 

Waarde1 ^-1⇒waarde

$$(3.1)^{-1} = 0.322581$$

Lijst1 ^-1⇒lijst

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.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} = \begin{bmatrix} -2 & 1 \\ 3 & -1 \\ 2 & 2 \end{bmatrix}$$

| (beperkende operator)

  -toetsen

Uitdr | BooleaanseUitdr1
[andBooleaanseUitdr2]...

$$x+1|x=3 \quad 4$$

$$x+55|x=\sin(55) \quad 54.0002$$

Uitdr | BooleaanseUitdr1
[orBooleaanseUitdr2]...

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.

I (beperkende operator)

ctrl -toetsen

De beperkings-operator biedt drie basistypen functionaliteit:

- Substituties
- Intervalbeperkingen
- Uitsluitingen

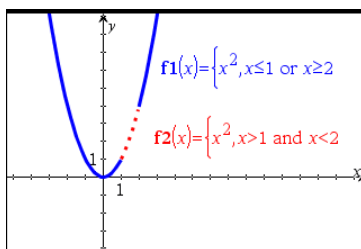
Substituties zijn in de vorm van een gelijkheid, zoals $x=3$ of $y=\sin(x)$. Om het meest effectief te zijn moet de linkerkant een enkelvoudige variabele zijn. *Uitdr* | *Variabele* = *waarde* substitueert *waarde* elke keer dat *Variabele* voorkomt in *Uitdr*.

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.

$x^3-2\cdot x+7 \rightarrow f(x)$	Done
$f(x) _{x=\sqrt{3}}$	8.73205

$\text{nSolve}(x^3+2\cdot x^2-15\cdot x=0,x)$	0.
$\text{nSolve}(x^3+2\cdot x^2-15\cdot x=0,x) _{x>0 \text{ and } x<5}$	3.



→ (opslaan)

ctrl var -toets

Waarde → *Var*

Lijst → *Var*

Matrix → *Var*

Uitdr → *Functie*(*Param1*,...)

Lijst → *Functie*(*Param1*,...)

Matrix → *Functie*(*Param1*,...)

$\frac{\pi}{4} \rightarrow myvar$	0.785398
$2\cdot \cos(x) \rightarrow yI(x)$	Done
$\{1,2,3,4\} \rightarrow lst5$	$\{1,2,3,4\}$
$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow matg$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
"Hello" → <i>str1</i>	"Hello"

Als de variabele *Var* niet bestaat, dan wordt deze gecreëerd en geïnitieerd naar *Waarde*, *Lijst* of *Matrix*.

Als de variabele *Var* reeds bestaat en niet vergrendeld of beveiligd is, dan wordt de inhoud ervan vervangen door *Waarde*, *Lijst* of *Matrix*.

Opmerking: u kunt deze operator vanaf het toetsenbord van de computer invoeren door `=:` als sneltoets in te voeren. Typ bijvoorbeeld `pi/4 =:` `mijnvar`.

Var := *Waarde*

Var := *Lijst*

Var := *Matrix*

Functie(*Param1*,...) := *Uitdr*

Functie(*Param1*,...) := *Lijst*

Functie(*Param1*,...) := *Matrix*

Als variabele *Var* niet bestaat, dan wordt *Var* gecreëerd en geïnitieerd naar *Waarde*, *Lijst* of *Matrix*.

Als *Var* reeds bestaat en niet vergrendeld of beveiligd is, dan wordt de inhoud ervan vervangen door *Waarde*, *Lijst* of *Matrix*.

<i>myvar</i> := $\frac{\pi}{4}$	.785398
<i>y1</i> (<i>x</i>) := $2 \cdot \cos(x)$	<i>Done</i>
<i>lst5</i> := { 1,2,3,4 }	{ 1,2,3,4 }
<i>matg</i> := $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
<i>str1</i> := "Hello"	"Hello"

© [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 *binair*Getal

In de Dec-grondtalmodus:

0b10+0hF+10

27

0h *hexadecimaal*Getal

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 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.

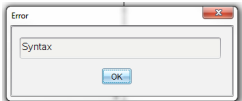
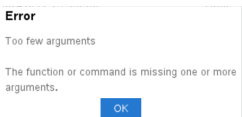
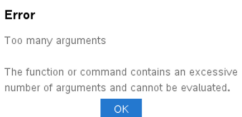
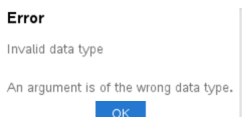
Grafisch scherm	Standaard
Hoogte	212
Breedte	318
Kleur	wit: 255,255,255

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.

Catalogus >
CXII

x , y , *breedte*, *hoogte* **wissen**

Wiss.

Wist het volledige scherm als er geen parameters zijn opgegeven.

Wist het volledige scherm

Als x , y , *breedte* en *hoogte* zijn opgegeven, wordt de rechthoek gewist die door de parameters is gedefinieerd.

10,10,100,50 wissen

Wist een rechthoekig gebied met linkerbovenhoek op (10, 10) en met breedte 100, hoogte 50

DrawArcCatalogus > 
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

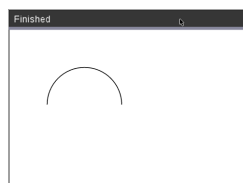
De "booghoek" definieert de zwaai van de boog.

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



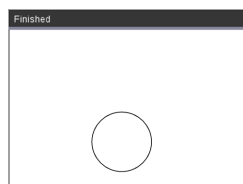
Zie ook: [FillArc](#)

DrawCircleCatalogus > 
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

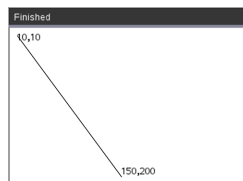
DrawLine $x1, y1, x2, y2$

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.

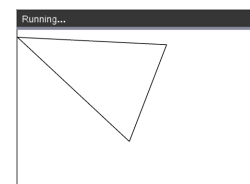
Opmerking: DrawPoly $x1, y1, x2, y2, x3, y3...xn, yn$ wordt **NIET** automatisch verbonden met $x1, y1$.

Uitdrukkingen die evalueren naar een lijst met echte drijvers
 $xlist, ylist$

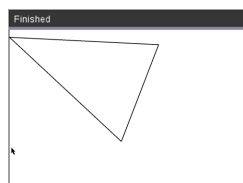
Uitdrukkingen die evalueren tot een enkele, echte drijver
 $x1, y1 ... xn, yn$ = coördinaten voor hoekpunten van veelhoek

Opmerking: DrawPoly: Afmetingen invoer (breedte/hogte) 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.

```
xlist:={0,200,150,0}  
ylist:={10,20,150,10}  
DrawPoly xlist,ylist
```



DrawPoly
0,10,200,20,150,150,0,10



Zie ook: [FillPoly](#)

DrawRect

DrawRect *x, y, breedte, hoogte*

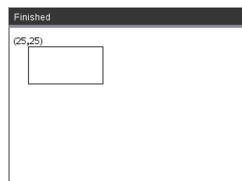
x, y: coördinaat linksboven van rechthoek

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

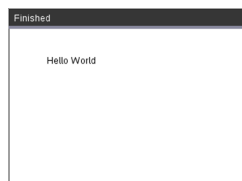
DrawText *x, y, exprOrString1 [,exprOrString2]...*

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"



FillArc

Catalogus > 
CXII**FillArc** *x, y, breedte, hoogte startAngle, arcAngle*

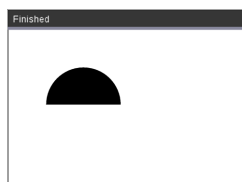
FillArc 50,50,100,100,0,180

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



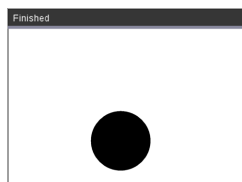
FillCircle

Catalogus > 
CXII**FillCircle** *x, y, radius*

FillCircle 150,150,40

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](#).

Hier!

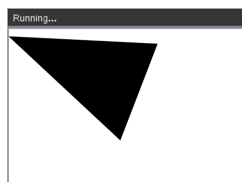
FillPoly

Catalogus > 
CXII**FillPoly** *xlist, ylist*

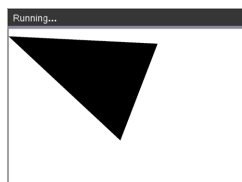
of

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*

x, y: coördinaat linksboven van rechthoek

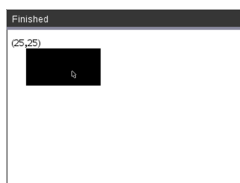
breedte, hoogte: breedte en hoogte van de rechthoek

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](#)

FillRect 25,25,100,50



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

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

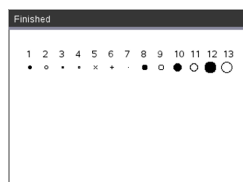
vorm: een getal tussen 1 en 13 dat de vorm specificeert

- 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

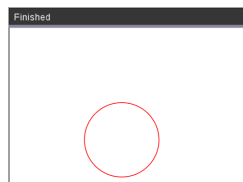


SetColorCatalogus > 
CXII**SetColor**

Roodwaarde, Groenwaarde, Blauwwaarde

Geldige waarden voor rood, groen en blauw liggen tussen 0 en 255

Stelt de kleur in voor de volgende tekenopdrachten

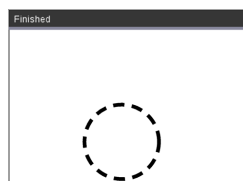
SetColor 255,0,0
DrawCircle 150,150,100**SetPen**Catalogus > 
CXII**SetPen**

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.

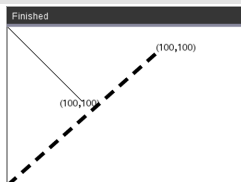
SetWindow 0,160,0,120
zal het uitvoervenster instellen op 0,0 in de linkeronderhoek met een breedte van 160 en een hoogte van 120
DrawLine 0,0,100,100
SetWindow 0,160,0,120
SetPen 3,3
DrawLine 0,0,100,100

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

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™ 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. 42 en **isVoid()**, pag. 80.

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	$\{1,_,1,1,1,1\}$

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	$\{_,_,0,0\}$
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 `sqr t ( 6 )` typen op de invoerregel. Wanneer u op `[enter]` drukt, verandert de uitdrukking `sqr t ( 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>sqr t (...)</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>

Vanaf het toetsenbord van de computer

Om dit in te voeren:	Typ deze snelkoppeling:
i (imaginaire constante)	<code>@i</code>
e (natuurlijke logaritme grondtal e)	<code>@e</code>
E (wetenschappelijke notatie)	<code>@E</code>

Om dit in te voeren:	Typ deze snelkoppeling:
$\top$ (transponeren)	@t
$^{\circ}$ (radialen)	@r
$^{\circ}$ (graden)	@d
g (decimale graden)	@g
$\angle$ (hoek)	@<
► (conversie)	@>
►Decimal, ►approxFraction (), en zo verder.	@>Decimal, @>approxFraction() , en zo verder.

EOS (Equation Operating System)-hiërarchie

In deze paragraaf wordt het Equation Operating System (EOS™) beschreven, dat gebruikt wordt door de TI-Nspire™-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™-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 $10 \rightarrow r$ en $"r" \rightarrow s1$, 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

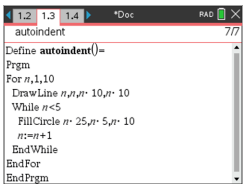
Auto-inspringen in de programmeereditor

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.



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

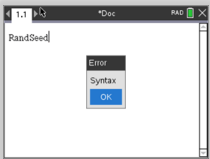
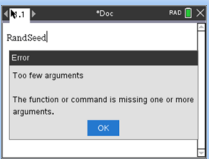
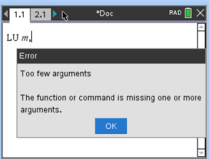
Fouten

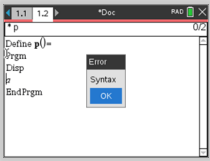
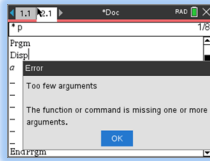
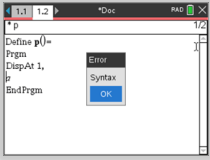
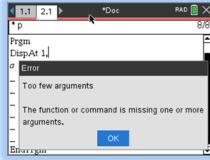
Fouttoestand	Nieuwe melding
Fout in conditieoverzicht (If/While)	Een conditionele instructie gaf geen oplossing voor TRUE of FALSE OPMERKING: Met de wijziging om de cursor op de regel met de fout te plaatsen, hoeven we niet langer te specificeren of de fout in de instructie "If" of "While" staat.
Ontbrekende 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

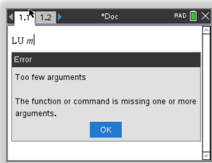
Syntaxfouten

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
	
	

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.



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	8987551792.261 _m/_F
_Fc	Constante van Faraday	96485.33212 _coul/_mol
_g	zwaartekrachtversnelling	9.80665 _m/_s ²
_Gc	zwaartekrachtconstante	6.6743E-11 _m ³ /_kg/_s ²
_h	Constante van Planck	6.62607015E-34 _J _s
_k	constante van Boltzmann	1.380649E-23 _J/_°K
_μ0	permeabiliteit van een vacuüm	1.25663706212E-6 _N/_A ²
_μb	Bohr magneton	9.274009994E-24 _J _m ² /_Wb
_Me	elektronenrustmassa	9.1093837015E-31 _kg
_Mμ	muon massa	1.883531627E-28 _kg
_Mn	neutronenrustmassa	1.67492749804E-27 _kg
_Mp	protonenrustmassa	1.67262192369E-27 _kg
_Na	getal van Avogadro	6.02214076E23 /_mol
_q	elektronenlading	1.602176634E-19 _coul
_Rb	Bohr-straal	5.29177210903E-11 _m
_Rc	Molaire gasconstante	8.314462618 _J/_mol/_°K
_Rdb	Rydbergconstante	10973731.568160/_m
_Re	elektronstraal	2.8179403262E-15 _m
_u	atomaire massa	1.6605390666E-27 _kg
_Vm	molair volume	2.241396954E-2 _m ³ /_mol
_ε0	diëlektrische constante van een vacuüm	8.8541878128E-12 _F/_m
_σ	constante van Stefan-Boltzmann	5.670367E-8 _W/_m ² /_°K ⁴
_φ0	magnetisch flux-quantum	2.067833831E-15 _Wb

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. 172.

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 <code>If a<b</code> veroorzaakt deze fout als <code>a</code> of <code>b</code> ongedefinieerd is wanneer de <code>If</code> -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 Booleaanse 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 onderstrepingstekens 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 $\text{rand}(0)$ 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

Foutcode	Beschrijving
300	Lijst of matrix verwacht met 2 of 3 elementen
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.

Foutcode	Beschrijving
565	Ongeldig buiten programma
570	Ongeldige padnaam Bijvoorbeeld: \var is ongeldig.
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

Foutcode	Beschrijving
740	Ontbrekend Then in het blok If..EndIf
750	Naam is geen functie of programma
765	Geen functies geselecteerd
780	Geen oplossing gevonden
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	Recursie 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: sto →

Foutcode	Beschrijving
	• := • Define om waarden aan variabelen toe te kennen.
965	Niet-gelicenseerd OS
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

Foutcode	Beschrijving
1120	Geen tekenverandering
1130	Argument kan geen lijst of matrix zijn
1140	Argumentfout 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.

Foutcode	Beschrijving
	Bibliotheken vernieuwen. Zie voor meer informatie de paragraaf Bibliotheek in de documentatie.
1210	Ongeldige naam voor sneltoets bibliotheek. Zorg ervoor dat de naam: - geen punt bevat - niet met een onderstrepingssteken 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.

Foutcode	Beschrijving
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 op te slaan van waarschuwingen die gegenereerd zijn door het uitwerken van een uitdrukking. Deze tabel geeft een overzicht van de numerieke waarschuwingcodes en de bijbehorende berichten. Zie voor een voorbeeld van het opslaan van waarschuwingcodes **warnCodes()**, pag. 181.

Waarschuwingcode	Melding
10000	Bewerking geeft mogelijk ongeldige oplossingen. Probeer, indien van toepassing, grafische methoden te gebruiken om de resultaten te verifiëren.
10001	Het differentiëren van een vergelijking kan een valse vergelijking opleveren.
10002	Twijfelachtige oplossing Probeer, indien van toepassing, grafische methoden te gebruiken om de resultaten te verifiëren.
10003	Twijfelachtige nauwkeurigheid Probeer, indien van toepassing, grafische methoden te gebruiken om de resultaten te verifiëren.
10004	Bewerking veroorzaakt mogelijk verlies van oplossingen. Probeer, indien van toepassing, grafische methoden te gebruiken om de resultaten te verifiëren.
10005	cSolve specificeert mogelijk meer nulpunten.
10006	Solve specificeert mogelijk meer nulpunten. Probeer, indien van toepassing, grafische methoden te gebruiken om de resultaten te verifiëren.
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) Probeer, indien van toepassing, grafische methoden te gebruiken om de resultaten te verifiëren.
10008	Het domein van het resultaat is mogelijk kleiner dan het domein van de invoer.

Waarschuwingcode	Melding
10009	Het domein van het resultaat is mogelijk groter dan het domein van de invoer.
10012	Niet-reële berekening
10013	∞^0 of undef^0 vervangen door 1
10014	undef^0 is vervangen door 1
10015	1^∞ of 1^undef vervangen door 1
10016	1^undef is vervangen door 1
10017	Overschrijding is vervangen door ∞ of $-\infty$
10018	De bewerking vraagt om en retourneert een 64-bits waarde.
10019	Bron uitgeput, vereenvoudiging is mogelijk onvolledig.
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	Scalaire grootheid is vermenigvuldigd met de eenheidsmatrix.
10024	Resultaat verkregen door middel van een rekenkundige benadering.
10025	Equivalentie kan niet worden geverifieerd in de EXACT-modus.
10026	Een beperking kan genegeerd worden. Specificeer een beperking in de vorm "\" 'Variabele MathTestSymbool Constante' of een samenvoeging van deze vormen, bijvoorbeeld 'x<3 and 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 meer informatie over de duur en voorwaarden van de garantie of over de productservice.

Beperkte garantie. Deze garantie heeft geen invloed op uw wettelijke rechten.

Texas Instruments Incorporated

12500 TI Blvd.

Dallas, TX 75243

Index

,			
' , minutennotatie	210	, beperkende operator	212
-		+	
-, aftrekken[*]	191	+, optellen	191
!		/	
!, faculteit	202	/, delen[*]	193
"		=	
" , secondennotatie	210	=, is gelijk	198
#		≠, is niet gelijk aan[*]	198
#, indirectie	208	>	
#, indirectie-operator	237	>, groter dan	200
%		Π	
%, percentage	197	Π, product[*]	204
&		Σ	
&, toevoegen	202	Σ(), som[*]	205
*		ΣInt()	206
*, vermenigvuldigen	192	ΣPrn()	207
.		√	
.-, punt aftrekken	196	√, wortel[*]	204
.*, puntvermenigvuldiging	196	∫	
./, punt delen	196	∫, integraal[*]	204
.^, punt machtsverheffen	197	≤	
+. , punt optellen	195	≤, kleiner dan of gelijk aan	199
:		≥	
:=, toewijzen	214	≥, groter dan of gelijk aan	200
^		►	
^-1, omgekeerde	212	►, converteren naar hoek in	
^, macht	194	decimale graden[Grad]	72
		►approxFraction()	13

►Cilind, weergeven als cilindrische vector[Cilind]	37	1	
►DD, weergeven als decimale hoek [DD]	38	$10^{\wedge}()$, macht van tien	211
►Decimal, resultaat weergeven als decimaal[Decimaal]	38	2	
►DMS, weergeven als graden/minuten/seconden [DMS]	46	2-stuks stuksgewijs gedefinieerde functie template voor	2
►Grondtal10, weergegeven als decimaal geheel getal [Grondtal10]	18	A	
►Grondtal16, weergeven als hexadecimaal[Grondtal16]	19	aantal dagen tussen datums, dbd()	37
►Grondtal2, weergeven als binair [Grondtal2]	17	abs(), absolute waarde	7
►Polar, weergeven als polaire vector [Polar]	121	absolute waarde template voor	4
►Rad, converteren naar radialen	129	afgeleiden	
►Rect, weergeven als rechthoekige vector	132	eerste afgeleide, d()	202
►Sphere, weergeven als bolvormige vector[Sphere]	159	numerieke afgeleide, nDeriv()	108-109
⇒		numerieke afgeleide, nDerivative()	107
⇒ , logische implicatie[*]	201, 234	aflossingstabel, amortTbl()	7, 16
→		afmeting, dim()	43
→, opslaan	213	afronden, round()	142
↔		afsluiten, Exit	52
↔, logische dubbele implicatie[*] ..	201	af trekken, -	191
©		als, If	73
©, commentaar	215	amortTbl(), aflossingstabel	7, 16
°		and, Booleanse operator	8
°, graden/minuten/seconden[*]	210	anders, Else	73
°, gradennotatie[*]	210	angle(), hoek	9
0		ANOVA, eenwegs variantieanalyse ..	10
Ob, binaire indicator	215	ANOVA2way, tweewegs variantieanalyse	11
Oh, hexadecimale indicator	215	Ans, laatste antwoord	13
		antwoord (laatste), Ans	13
		approx(), benaderend	13
		approxRational()	14
		arccos()	14
		arccosh()	14
		arccosinus, cos ⁻¹ ()	28
		arccot()	14
		arcoth()	14
		arccsc()	14
		arccsch()	14
		arcsec()	14
		arcsech()	14
		arcsin()	15
		arcsinh()	15
		arcsinus, sin ⁻¹ ()	155

arctan()	15	char(), tekenreeks	21
arctangens, $\tan^{-1}()$	167	cilindrische vectorweergave, ►Cylind	37
arctanh()	15	ClearAZ	23
argumenten in TVM-functies	176	ClrErr, fout wissen	24
augment(),		colAugment	24
uitbreiden/aaneenvoegen	15	colDim(), matrixkolomafmeting	24
avgRC(), gemiddelde		colNorm(), matrixkolomnorm	25
veranderingssnelheid	15	combinaties, nCr()	106
		commentaar, ©	215
B		complex	
beëindigen		geconjugeerde, conj()	25
terwijl, EndWhile	183	conj(), complex geconjugeerde	25
benaderend, approx()	13	constructMat(), matrix construeren	25
bepaalde integraal		converteren	
template voor	6	►Rad	129
beperkende operator " "	212	4Grad	72
beperkende operator, volgorde van		correlatiematrix, corrMat()	27
uitwerking	236	corrMat(), correlatiematrix	27
bibliotheek		$\cos^{-1}$, arccosinus	28
snellkoppelingen naar objecten		cos(), cosinus	27
creëren	82	$\cosh^{-1}()$, arccosinus hyperbolicus	29
binair		cosh(), cosinus hyperbolicus	29
indicator, Ob	215	cosinus, cos()	27
weergeven, ►Grondtal2	17	$\cot^{-1}()$, arccotangens	31
binnen string, inString()	76	cot(), cotangens	30
binomCdf()	20, 78	cotangens, cot()	30
binomPdf()	20	$\coth^{-1}()$, arccotangens hyperbolicus	31
bodem, floor()	57	coth(), cotangens hyperbolicus	31
bolvormige vectorweergave,		count(), items tellen in een lijst	32
►Sphere	159	countif(), items in een lijst	
Booleaanse operatoren		voorwaardelijk tellen	32
⇒	201, 234	cPolyRoots()	33
⇔	201	crossP(), uitwendig product	33
and	8	$\csc^{-1}()$, inverse cosecans	34
niet	112	csc(), cosecans	34
niet en	105	$\operatorname{csch}^{-1}()$, inverse cosecans	
noch	110	hyperbolicus	35
of	116	csch(), cosecans hyperbolicus	35
xof	183	CubicReg, derdegraads regressie	35
breuken		cumulatieve som, cumulativeSum()	36
propFrac	124	cumulativeSum(), cumulatieve som	36
template voor	1	cycle, Cycle	37
		Cycle, cycle	37
C		D	
Cdf()	55	d(), eerste afgeleide	202
ceiling(), plafond	20	dagen tussen datums, dbd()	37
centralDiff()	21	dbd(), dagen tussen datums	37

frequency()	60		
Frobenius-norm, norm()	111	Goto, ga naar	72
Func, functie	62	graden/minuten/seconden-notatie	210
Func, functie programmeren	62	graden/minuten/seconden- weergave, ►DMS	46
functie beëindigen, EndFunc	62	gradennotatie, °	210
functies		groepen, vergrendelen en ontgrendelen	91, 179
deel, fpart()	59	groepen, vergrendelingsstatus testen	69
functie programmeren, Func	62	grootste gemene deler, gcd()	62
functies en variabelen		groter dan of gelijk aan,	200
kopieëren	26	groter dan, >	200
functions			
user-defined	39		
G			
g, decimale graden	208	H	
ga naar, Goto	72	hexadecimaal	
gcd(), grootste gemene deler	62	indicator, Oh	215
gegevens weergeven, Disp	44, 146	weergeven, ►Grondtal16	19
geheel deel, iPart()	79	hoek, angle()	9
geheel getal, int()	76	hyperbolicus	
gemengde breuken, propFrac()		arccosinus, cosh ⁻¹ ()	29
gebruiken met	124	arcsinus hyperbolicus, sinh ⁻¹ ()	156
gemiddelde veranderingssnelheid, avgRC()	15	arctangens, tanh ⁻¹ ()	168
gemiddelde, mean()	97	cosinus, cosh()	29
geomCdf()	63	sinus hyperbolicus, sinh()	155
geomPdf()	63	tangens, tanh()	168
gereduceerde rij-echelonvorm, rref()	144	I	
Get	63, 226	If, als	73
getallenrij, seq()	147	ifFn()	74
getDenom(), noemer		imag(), imaginair deel	75
ophalen/terugsturen	64	imaginair deel, imag()	75
getKey()	65	indirectie-operator (#)	237
getLangInfo(), taalinformatie		indirectie, #	208
ophalen/terugsturen	68	instellen	
getLockInfo(), test de vergrendelingsstatus van de variabele of variabelegroep	69	modus, setMode()	149
getMode(), modusinstellingen		instellingen, huidige ophalen	69
ophalen	69	inString(), binnen string	76
getNum(), getal		int(), geheel getal	76
ophalen/terugsturen	70	intDiv(), delen door geheel getal	76
GetStr	70	integraal, ∫	204
getType(), get type of variable	71	interpolate(), interpoleren	77
getVarInfo(), variabele-informatie		inverse cumulatieve normale verdeling (invNorm()	79
ophalen/terugsturen	71	inverse, ^-1	212
gewijzigde interne rentabiliteit, mirr()	101	invF()	78
		invNorm(), inverse cumulatieve	79

normale verdeling)		nieuw, newList()	108
invf()	79	oplopend sorteren, SortA	158
Inv χ^2 ()	77	product, product()	124
iPart(), geheel deel	79	som, sum()	164-165
irr(), interne rentabiliteit		uitbreiden/aaneenvoegen,	
interne rentabiliteit, irr()	79	augment()	15
is gelijk, =	198	uitwendig product, crossP() ..	33
is niet gelijk aan, $\neq$	198	verschil, Δ list()	88
isPrime(), priemtoets	80	verschillen in een lijst, @list() ..	88
isVoid(), testen op lege elementen ..	80	lineaire regressie, LinRegAx	84
items in een lijst voorwaardelijk		lineaire regressie, LinRegBx	83, 85
tellen, countif()	32	links, left()	82
items tellen in een lijst, count()	32	LinRegBx, lineaire regressie	83
		LinRegMx, lineaire regressie	84
K		LinRegtIntervals, lineaire regressie ..	85
kansdichtheid, normPdf()	112	LinRegtTest	86
kleiner dan of gelijk aan, {	199	linSolve()	87
kleinste gemene veelvoud, lcm	81	list $\rightarrow$ mat(), lijst naar matrix	88
kwadratische regressie, QuadReg	126	ln(), natuurlijk logaritme	89
		LnReg, logaritmische regressie	89
L		Local, lokale variabele	91
label, Lbl	81	Lock, variabele of variabelegroep	
Lbl, label	81	vergrendelen	91
lcm, kleinste gemene veelvoud	81	Log	
left(), links	82	template voor	2
lege elementen	232	logaritmen	89
lege elementen, testen op	80	logaritmische regressie, LnReg	89
lege elementen, verwijderen	42	logische dubbele implicatie, $\Leftrightarrow$	201
lengte van string	43	logische implicatie, $\Rightarrow$	201, 234
LibPriv	40	Logistic, logistische regressie	92
LibPub	40	LogisticD, logistische regressie	93
libShortcut(), snelkoppelingen naar		logistische regressie, Logistic	92
bibliotheekobjecten		logistische regressie, LogisticD	93
creëren	82	lokaal, Local	91
lijst naar matrix, list $\rightarrow$ mat()	88	lokale variabele, Local	91
lijst, items tellen in	32	loop, Loop	95
lijst, items voorwaardelijk tellen in ..	32	Loop, loop	95
lijsten		LU, matrix beneden-boven-	
aflopend sorteren, SortD	158	decompositie	95
cumulatieve som,		M	
cumulativeSum()	36	macht van tien, 10^()	211
inwendig product, dotP()	46	macht, ^	194
lege elementen in	232	machtsregressie, PowerReg122, 135, 137, 169	
lijst naar matrix, list $\rightarrow$ mat()	88	mat $\rightarrow$ list(), matrix naar lijst	96
matrix naar lijst, mat $\rightarrow$ list()	96	matrices	
maximum, max()	96	afmeting, dim()	43
mid-string, mid()	99	beneden-boven-decompositie,	95
minimum, min()	100		

niet, Booleaanse operator	112	P►Ry(), rechthoekig y-coördinaat	119
nieuw		PassErr, fout overbrengen	119
lijst, newList()	108	Pdf()	59
matrix, newMat()	108	percentage, %	197
nInt(), numerieke integraal	109	permutaties, nPr()	113
noch, Booleaanse operator	110	piecewise()	120
nom), effectief naar nominaal		plafond, ceiling()	20-21, 33
percentage converteren	110	poissCdf()	120
nominaal percentage, nom()	110	poissPdf()	120
norm(), Frobenius-norm	111	polair	
normale verdelingskans, normCdf()	111	vectorweergave, ►Polar	121
normCdf()	111	polyEval(), veelterm uitwerken	121
normPdf()	112	PolyRoots()	122
notatie in decimale graden, g	208	pool	
nPr(), permutaties	113	coördinaat, R►Pr()	129
npv(), netto contante waarde	114	coördinaat, R►Pθ()	128
nSolve(), numerieke oplossing	114	PowerReg, machtsregressie	122
numeriek		Prgm, programma definiëren	123
afgeleide, nDeriv()	108-109	priemgetaltoets, isPrime()	80
afgeleide, nDerivative()	107	proberen, Try	172
integraal, nInt()	109	prodSeq()	124
oplossing, nSolve()	114	product (P)	
		template voor	5
O		product(), product	124
objecten		product, product()	124
snelkoppelingen naar		product, Π()	204
bibliotheek creëren	82	programma's en programmeren	
of (Booleaans), of	116	I/O-scherm weergeven, Disp	146
of Booleaanse operator	116	programma's en programmeren	
omgekeerde, $\wedge^{-1}$	212	fout wissen, ClrErr	24
OneVar, statistieken voor één		I/O-scherm weergeven, Disp	44
variabele	115	proberen, Try	172
ontbinden, factor()	55	programmeren	
operatoren		fout overbrengen, PassErr	119
volgorde van uitwerking	236	gegevens weergeven, Disp	44, 146
ophalen/terugsturen		programma definiëren, Prgm	123
getal, getNum()	70	programs	
noemer, getDenom()	64	defining private library	40
variabele-informatie, getVarInfo		defining public library	40
()	68, 71	propFrac, echte breuk	124
opmaakstring, format()	58	punt	
opslaan		aftrekken, .N	196
symbool, &	213-214	delen, .P	196
optellen, +	191	macht, .^	197
ord(), numerieke tekencode	118	optellen, .+	195
		vermenigvuldigen, .*	196
P		Q	
P►Rx(), rechthoekige x-coördinaat	118	QR-ontbinding, QR	125

QR, QR-ontbinding	125	TwoVar	160
QuadReg, kwadratische regressie	126	resultaten, statistieken	160
QuartReg, vierdegraads regressie	127	Return, terugkeren	138
R		right(), rechts	139
R, radialen	209	right, right()	50, 181
R►Pr(), poolcoördinaat	129	rij-echelonvorm, ref()	133
R►Pθ(), poolcoördinaat	128	rk23(), Runge-Kutta-functie	139
radialen, R	209	rotate(), roteren	141
rand(), willekeurig getal	129	roteren, rotate()	141
randBin, willekeurig getal	129	round(), afronden	142
randInt(), willekeurig geheel getal	130	rowAdd(), matrixrij-optelling	143
randMat(), willekeurige matrix	130	rowDim(), matrixrij-afmeting	143
randNorm(), willekeurige norm	131	rowNorm(), matrixrijnorm	143
randPoly(), willekeurige veelterm	131	rowSwap(), matrixrij-omwisseling	144
randSamp()	131	rref(), gereduceerde rij- echelonvorm	144
RandSeed, willekeurig getal-seed	131	S	
real(), reëel	132	sec ⁻¹ (), inverse secans	145
rechthoekig y-coördinaat, P►Ry()	119	sec(), secans	145
rechthoekige vectorweergave, ►Rect	132	sech ⁻¹ (), inverse secans hyperbolicus	146
rechthoekige x-coördinaat, P►Rx()	118	sech(), secans hyperbolicus	145
rechts, right()	77, 139	secondennotatie, "	210
reëel, real()	132	seq(), getallenrij	147
reeksen		seqGen()	147
gebruiken om variabelenamen te creëren	237	seqn()	148
tekenreeks, char()	21	sequence, seq()	147-148
ref(), rij-echelonvorm	133	setMode(), instellingsmodus	149
RefreshProbeVars	134	shift(), verschuiven	150
regressies		sign(), teken	152
derdegraads, CubicReg	35	simult(), simultane vergelijkingen	153
exponentiële, ExpReg	53	simultane vergelijkingen, simult()	153
kwadratisch, QuadReg	126	sin ⁻¹ (), arcsinus	155
lineaire regressie, LinRegAx	84	sin(), sinus	154
lineaire regressie, LinRegBx	83, 85	sinh ⁻¹ (), arcsinus hyperbolicus	156
logaritmisch, LnReg	89	sinh(), sinus hyperbolicus	155
Logistic	92	SinReg, sinusoidale regressie	156
logistisch, Logistic	93	sinus, sin()	154
machtsregressie, PowerReg	122, 135, 137, 169	sinusoidale regressie, SinReg	156
mediaan-mediaan-lijn, MedMed	98	snelkoppelingen, toetsenbord	234
MultReg	102	sneltoetsen	234
sinusoïde, SinReg	156	som (G) template voor	5
vierdegraads, QuartReg	127	som van hoofdsombetalingen	207
remain(), rest	135	som van rentebetalingen	206
rest, remain()	135	som, sum()	164
resultaatwaarden, statistieken	161	som, Σ()	205
resultaten voor twee variabelen,	177		

n-de wortel	2		
N-stuks stuksgewijs		U	
gedefinieerde functie	3	uitbreiden/aaneenvoegen, augment(	
product (P)	5	)	15
som (G)	5	uitdrukkingen	
stelsel van 2 vergelijkingen	3	string naar uitdrukking, expr() ..	53
stelsel van vergelijkingen (N		uitsluiting met operator " "	212
vergelijkingen)	3	uitwendig product, crossP()	33
tweede afgeleide	6	uitwerking, volgorde van	236
wortel	1	unitV(), eenheidsvector	179
terugkeren, Return	138	unLock, variabele of variabelegroep	
terwijl beëindigen, EndWhile	183	ontgrendelen	179
terwijl, While	183	user-defined functions	39
Test_2S, F-toets met 2 steekproeven	61	user-defined functions and	
testen op lege elementen, isVoid() ..	80	programs	40
tijdwaarde van geld, aantal			
betalingen	175	V	
tijdwaarde van geld, betalingsbedrag	176	variabele	
tijdwaarde van geld, contante		naam van een tekenreeks	
waarde	176	creëren	237
tijdwaarde van geld, Rente	175	variabele of functie kopiëren,	
tijdwaarde van geld, toekomstige		CopyVar	26
waarde	175	variabelen	
TInterval, t-		alle variabelen bestaande uit	
betrouwbaarheidsinterval ..	170	één letter wissen	23
tInterval_2Samp, t-		lokaal, Local	91
betrouwbaarheidsinterval		wissen, DelVar	41
met 2 steekproeven	171	variabelen en functies	
toevoegen, &	202	kopiëren	26
tPdf(), studentt-kansdichtheid	171	variabelen en variabelegroepen	
trace()	172	ontgrendelen	179
transponeren, T	166	variabelen en variabelegroepen	
Try, foutbehandelingscommando	172	vergrendelen	91
Try, proberen	172	variabelen, vergrendelen en	
tTest, t-toets	173	ontgrendelen	69, 91, 179
tTest_2Samp, t-toets met twee		variantie, variance()	180
steekproeven	174	varPop()	179
TVM-argumenten	176	varSamp(), steekproefvariantie	180
tvmFV()	175	vectoren	
tvmI()	175	cilindrische vectorweergave,	
tvmN()	175	4Cilind	37
tvmPmt()	176	eenheid, unitV()	179
tvmPV()	176	inwendig product, dotP()	46
tweede afgeleide		uitwendig product, crossP() ..	33
template voor	6	veelterm uitwerken, polyEval()	121
TwoVar, resultaten voor twee		veeltermen	
variabelen	177	uitwerken, polyEval()	121
		willekeurig, randPoly()	131

verdelingsfuncties			
binomCdf()	20, 78	veelterm, randPoly()	131
binomPdf()	20	willekeurige steekproef	131
invNorm()	79	Wiss.	220
invt()	79	wissen	
Inv χ^2 ()	77	lege elementen uit een lijst	42
normCdf()	111	variabele, DelVar	41
normPdf()	112	Wissen	
poissCdf()	120	fout, ClrErr	24
poissPdf()	120	wortel	
tCdf()	169	template voor	1
tPdf()	171	wortel, $\pm$ ()	159, 204
χ^2 2way()	21		
χ^2 Cdf()	22	X	
χ^2 GOF()	22	x^2 , kwadraat	195
χ^2 Pdf()	23	XNOR	201
vermenigvuldigen, *	192	xof, Booleaans exclusief of	183
verschuiven, shift()	150		
verwijderen		Z	
lege elementen uit een lijst	42	zInterval, z-	
Verzoek	135	betrouwbaarheidsinterval	184
VerzoekStr	137	zInterval_1Prop, z-	
vierdegraads regressie (QuartReg)	127	betrouwbaarheidsinterval	
vullen	224-225	met één proportie	185
		zInterval_2Prop, z-	
W		betrouwbaarheidsinterval	
waarschuwingscodes en berichten	251	met twee proporties	185
Wait-opdracht	180	zInterval_2Samp, z-	
wanneer, when()	182	betrouwbaarheidsinterval	
warnCodes(), Warning codes	181	met twee steekproeven	186
weergegeven als		zTest	187
binair, ►Grondtal2	17	zTest_1Prop, z-toets met één	
bolvormige vector, ►Sphere	159	proportie	188
cilindrische vector, ►Cylind	37	zTest_2Prop, z-toets voor twee	
decimaal geheel getal,		proporties	188
►Grondtal10	18	zTest_2Samp, z-toets met twee	
decimale hoek, ►DD	38	steekproeven	189
graden/minuten/seconden,			
►DMS	46	Δ	
hexadecimaal, ►Grondtal16	19	Δ list(), lijstverschil	88
polaire vector, ►Polar	121		
rechthoekige vector, ►Rect	132	X	
when(), wanneer	182	χ^2 2way	21
While, terwijl	183	χ^2 Cdf()	22
willekeurig		χ^2 GOF	22
getal-seed, RandSeed	131	χ^2 Pdf()	23
matrix, randMat()	130		
norm, randNorm()	131		