



Numerieke benadering van vierkantwortels

Doel: De waarde van een vierkantwortel met een recursieve rij benaderen, het schrijven van een klein programma.

Sleutelwoorden: Reeks, recursie, programma, vierkantwortel.

Alles over de HP Prime:
<http://www.hp-prime.nl>

Werkwijze: Stel dat de wortel van het getal P berekend moet worden, kies een startwaarde (een getal waarvan u denkt dat het de oplossing benadert), deel nu het getal P door deze startwaarde als de startwaarde de wortel van P is dan is de uitkomst gelijk aan de startwaarde en bent u meteen klaar, goed gegokt. Maar ja, lang niet in alle gevallen zal de startwaarde precies de wortel zijn, stel $P = 100$ en de startwaarde wordt, enigszins achteloos, gelijk aan 50 gekozen, dan is P gedeeld door de startwaarde 2 en dus niet gelijk aan de startwaarde. Dan is de startwaarde niet gelijk aan de wortel. U ziet ook meteen dat 50 te groot is en (daardoor) 2 te klein oftewel de echte waarde van de wortel zit tussen 2 en 50. Door nu het gemiddelde van 2 en 50 als nieuwe startwaarde te kiezen is onze startwaarde verbeterd.

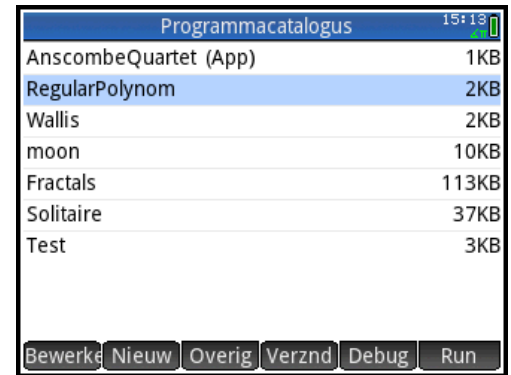
$$(2+50)/2=26$$

We voeren we de berekening nog een keer uit maar nu met startwaarde 26. Nu kunt u denken; „zo blijf ik aan de gang“ maar uit de praktijk blijkt dat in de meeste gevallen met een niet al te ongelukkig gekozen startwaarde er snel al een behoorlijke benadering van het juiste antwoord komt.

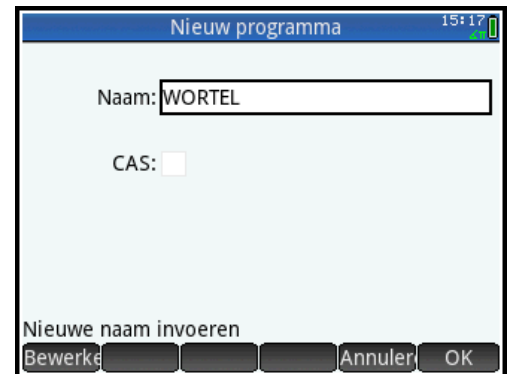
Instellingen: Kies voor de cijfergroepering met een decimale punt, zodat de schermen op de rekenmachine en in dit document overeenkomen.



Uitvoering: Kies **Shift** en **F1** waarmee de programma catalogus opent. (Afhankelijk van de programma's op uw rekenmachine wijkt dit scherm natuurlijk af van het scherm hiernaast).



Gebruik **Nieuw** voor het openen van een nieuw programma, voer de naam WORTEL in. Gebruik **ALPHA** waarna u aan een stuk de hoofdletters kan typen, sluit het typen van letters af met nog eens **ALPHA**.

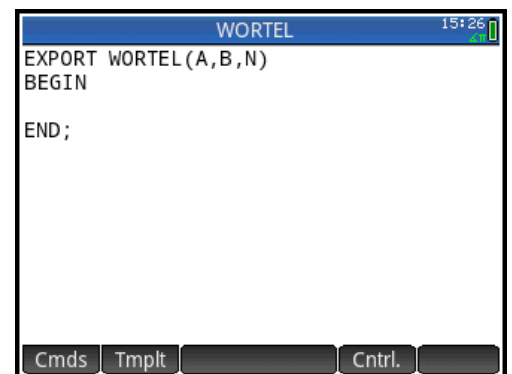


Bevestig met **Enter**. Er verschijnt een scherm voor de invoer van het programma, de eerste regels en de afsluiting staan al in beeld, het eigenlijke programma kan hiertussen worden ingevoerd.



Voer tussen de haakjes en achter EXPORT de te gebruiken variabelen in; A, B en N.

A is het getal waarvan de wortel berekend moet worden, B is de startwaarde en N is het aantal stappen waarbij steeds een betere benadering voor de wortel wordt berekend.



Zet de cursor in de lege regel onder BEGIN; en kies **Tmplt** de optie 4 Variabele en 1 LOCAL.

Type daarna I en sluit de regel af met ; (puntkomma) ; de puntkomma staat onder **ALPHA** en **Ans**.

NB: Alle programmaregels moeten hiermee worden afgesloten.

Met de declaratie via LOCAL wordt de variabele I alleen (lokaal) binnen dit programma gebruikt, buiten het programma heeft de I geen waarde. De andere variabelen A, B en N zijn al automatisch leeg na gebruik van het programma.

Zet de cursor in de volgende (lege) regel en kies **Cmnds** dan optie 6 I/O (van input/output) en optie 9 PRINT.

Zet dan de cursor achter de haakjes en sluit wederom de regel af met een ,puntkomma'. Gebruik **Enter** voor de bevestiging van deze invoer, daarmee verplaatst u meteen de cursor naar de volgende regel (die als lege regel nieuw wordt ingevoegd).

De ingevoerde regel met alleen het PRINT commando zonder tekst tussen de haakjes maakt de Terminal leeg, de Terminal is het scherm waar de uitvoer van een programma wordt geplaatst. De Terminal kan worden geopend door de O toets vast te houden en dan op de t (van de letter T) te drukken.

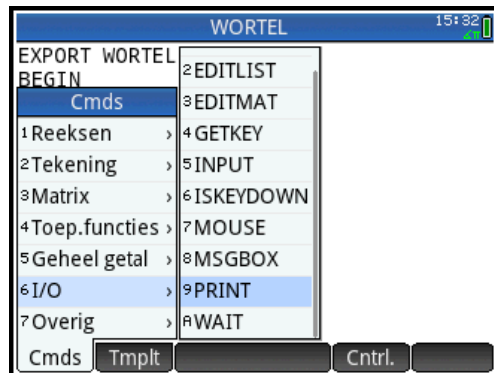
Gebruik nu **Tmplt** de optie 3.Loop en de optie 1.For

Hiermee opent een sjabloon voor een stukje programma dat zich een aantal keren herhaald, in ons programma ,loopt' de variabele I van 1 tot N en worden voor alle waarden die I doorloopt de volgende programmaregels steeds opnieuw uitgevoerd. Met de invoer van de waarde voor N bepaalt de gebruiker het aantal herhalingen.

U ziet nu stukjes tekst met lege stukken ertussen (het sjabloon) voor het FOR commando. Vul op de lege plekken de tekst in zoals in het scherm hiernaast.



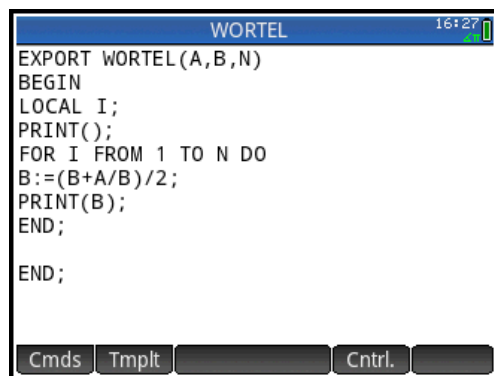
```
WORTEL 16:18
EXPORT WORTEL(A,B,N)
BEGIN
END;
Prgm. Opdrachten
1 Blokkeren >
2 Vertakking >
3 Loop (Lus) >
4 Variabele > 1 LOCAL
5 Functie > 2 EXPORT
Cmnds Tmplt Cntrl
```



```
WORTEL 15:32
EXPORT WORTEL
BEGIN
Cmnds 2 EDITLIST
3 EDITMAT
4 GETKEY
5 INPUT
6 I/O > 6 ISKEYDOWN
7 MOUSE
8 MSGBOX
9 PRINT
7 Overig > 9 WAIT
Cmnds Tmplt Cntrl
```



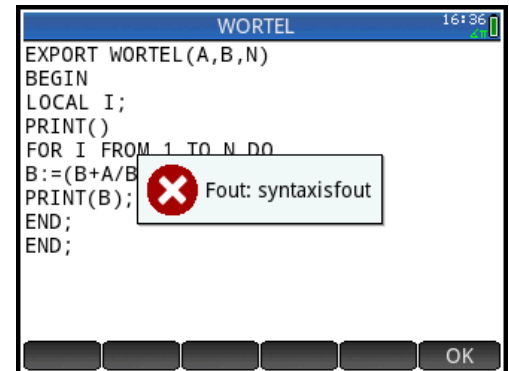
```
WORTEL 16:23
EXPORT WORTEL(A,B,N)
BEGIN
LOCAL I;
PRINT();
Prgm. Opdrachten 1 FOR
2 FOR STEP
3 FOR DOWN
4 FOR DOWN STEP
5 WHILE
6 REPEAT
7 BREAK
8 CONTINUE
1 Blokkeren >
2 Vertakking >
3 Loop (Lus) >
4 Variabele >
5 Functie >
Cmnds Tmplt Cntrl
```



```
WORTEL 16:27
EXPORT WORTEL(A,B,N)
BEGIN
LOCAL I;
PRINT();
FOR I FROM 1 TO N DO
B:=(B+A/B)/2;
PRINT(B);
END;
END;
Cmnds Tmplt Cntrl
```

Het begint met invullen van I, deze variabele begint bij 1 en gaat dan in stappen van 1 oplopen tot aan N. Bij iedere stap worden de twee regels eronder uitgevoerd. In de eerste regel wordt de (nieuwe) waarde van B berekend als het gemiddelde van de (oude) waarde van B en de waarde van A gedeeld door de (oude) waarde van B. Daarna wordt de dan berekende waarde van B geprint in het scherm. De END; in de volgende regel sluit het stuk dat moet worden herhaald af.

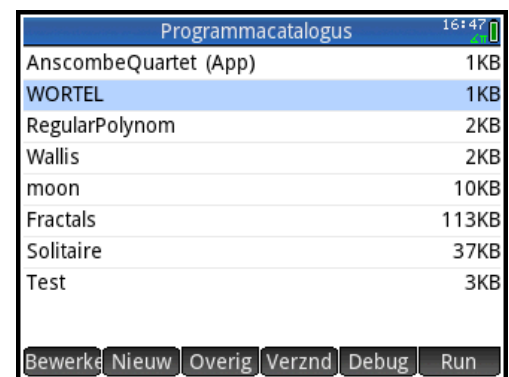
Eigenlijk is het programma nu klaar, alleen de laatste lege regel tussen de beide END commando's kan nog de lege regel worden verwijderd. Via de knop **Cntrl.** kan worden gecontroleerd of er geen (syntax) fouten in het programma staan. Druk erop en als u bijvoorbeeld ergens de puntkomma als afsluiting van een regel heeft vergeten krijgt u een foutmelding. Als voorbeeld haal ik even de puntkomma achter het eerste PRINT commando weg, dan verschijnt de melding zoals in het scherm hiernaast, druk dan **OK** en de cursor staat aan het begin van de volgende regel om aan te geven dat daar ergens de fout is gemaakt.



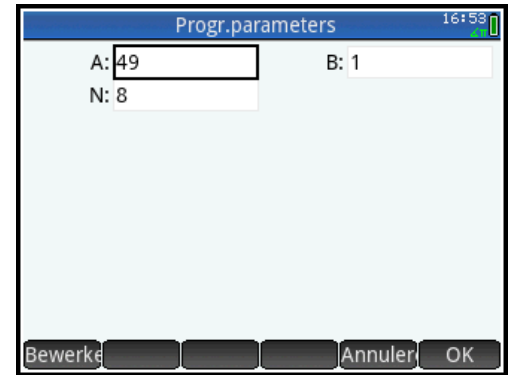
Verbeter de fout en gebruik opnieuw **Cntrl.** Als alles goed is ziet u dan dit scherm. Bevestig met **OK**



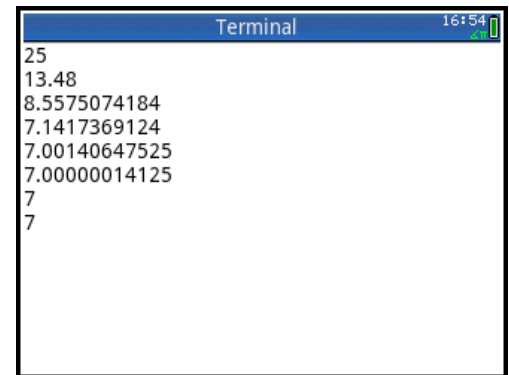
Kies **Esc** om uit de programma editor te komen en start het programma met **Run**



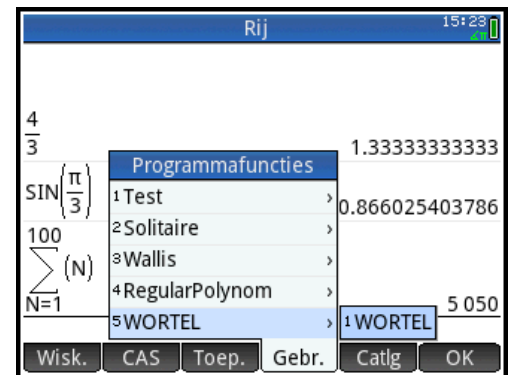
Dankzij de regel EXPORT(A,B,N); opent nu automatisch een dialoog scherm waar de waarden voor de drie te gebruiken variabelen kunnen worden ingegeven. Kies bijvoorbeeld A=49, (makkelijk om het programma te testen, we kennen de uitkomst al $\sqrt{49}=7$) dan de startwaarde B=1 en het aantal stappen N=8. Kies **OK** voor de executie van het programma.




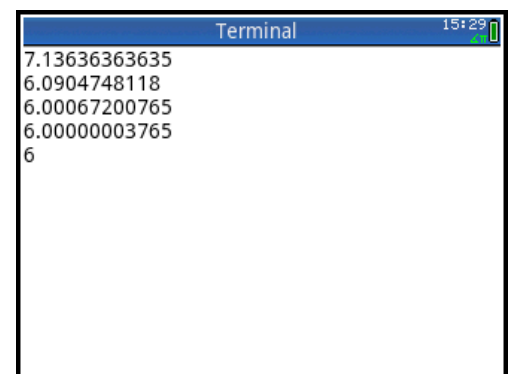
U ziet hier de Terminal met de bijbehorende uitvoer. Probeer nog maar eens een paar wortels met steeds verschillende startwaardes, u zult zien dat in veel gevallen heel snel de juiste waarde gevonden wordt.



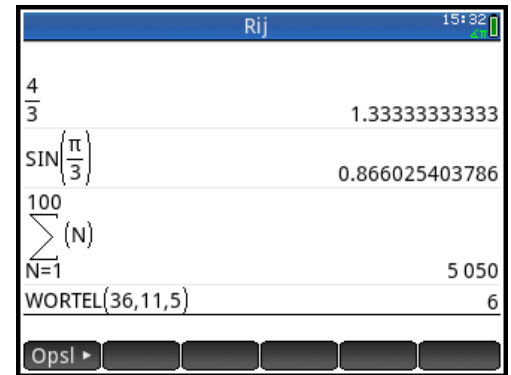
U kunt het programma ook in het Home scherm starten vanuit de gereedschapskist. Ga naar het Homescherm met  en druk op . Afhankelijk van het menu dat u hier de laatste keer heeft gebruikt opent zich een menu vanuit de onderrand van het scherm. Kies het menu met gebruikersfuncties (door de gebruiker gemaakte functies) via **Gebr.** en selecteer de optie WORTEL, dat moet in twee keer zoals u zult zien.



In de invoer regel van het Homescherm staat nu WORTEL(), met een knipperende cursor tussen de beide haakjes, vul hier drie getallen in bijvoorbeeld 36, 11 en 5 gescheiden door de komma en druk  voor de berekening, de terminal opent en u ziet in 5 stappen de benadering van de wortel van 36 met als startwaarde 11. Toevallig juist genoeg stappen om bij de correcte waarde 6 te eindigen.



Druk een willekeurige toets om de terminal te sluiten en u ziet het commando met de bijbehorende (laatste) uitvoer uit de terminal in de historie van het Homescherm.

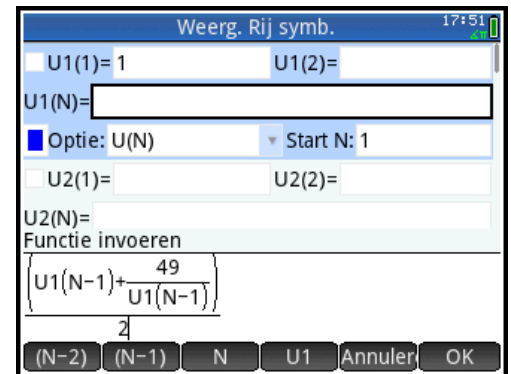


Met de Rij applicatie van de HP Prime kan dit recursieve proces ook worden bekeken. Kies **Apps Info** om de bibliotheek te openen en selecteer de app Rij.



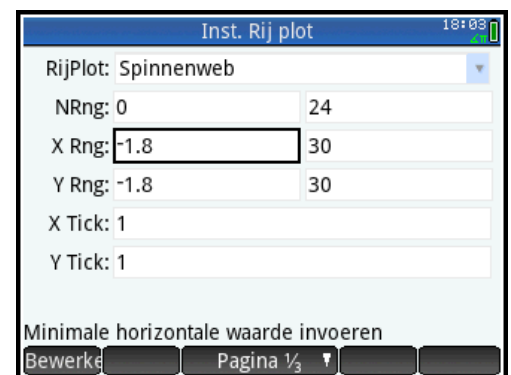
Maak de app eventueel leeg met **Shift** en **Esc Clear**. Wilt u de huidige inhoud bewaren, kies dan in de app bibliotheek voor **Opnl** en bewaar de inhoud van de app onder een passende naam, waarna u de Rij app zelf opnieuw opent en leegmaakt.

Vul de startwaarde 1 in achter U1(1) en zet de cursor op het veld voor de definitie van U1(N), voer de formule uit het scherm hiernaast in, u ziet onderin de beeldrand de termen U1 en (N-1) zodat u die niet letter voor letter moet intypen maar uit dit menu kan kopiëren.

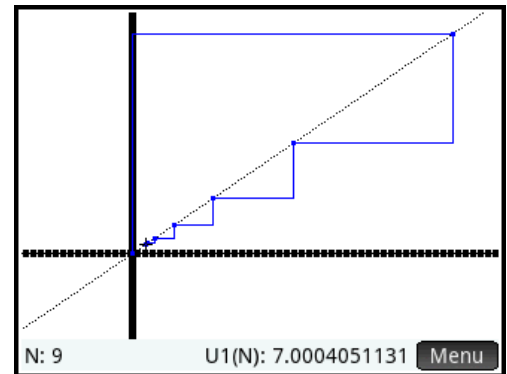
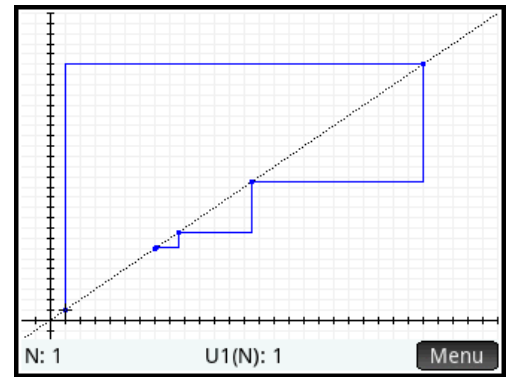


Bevestig met **Enter**.

Voor de grafiek nog geschikte afmetingen van het X-Y vlak via **Shift** en **Plot Setup** kiezen. Bijvoorbeeld zoals hiernaast, kies ook de Spinnenweb optie.



Bekijk nu de grafiek via **Plot** \rightarrow **Setup**. Nu ziet u wellicht waarom dit een spinnenwebgrafiek heet. De grafiek begint bij de blauwe punt linksonder op de lijn $Y=X$ daar hebben beide de waarde 1, dan wordt $U1(2)$ berekend = 25 de blauwe lijn loopt omhoog naar $Y=25$. Dan gaat de lijn opzij weer tot aan de lijn $Y=X$, want de berekende waarde (uitkomst) 25 wordt nu de invoer waarmee berekend gaat worden, oftewel we stappen van de laatste Y -waarde naar de volgende X -waarde. Met $X=25$ als invoer is de uitkomst 13,48, de blauwe lijn daalt naar die Y -waarde en gaat opzij naar de lijn $Y=X$ en met 13,48 als invoer wordt de volgende waarde berekend etc. tot na 7 berekeningen de wortel al zeer nauwkeurig is benaderd. Door in het symbolische scherm de startwaarde te veranderen die staat achter $U1(1)=$ kunt u zelf voor heel verschillende startwaarden zien hoe snel een goede benadering wordt gevonden.



Hiermee hebben we de numerieke benadering van een vierkantswortel of tweede machtwortel gerealiseerd, voor een derdemacht wortel is de benodigde aanpassing zeer eenvoudig.

Eerst in de Rij applicatie; ga naar het symbolische scherm met en zet de cursor op het invoerveld achter $U1(N)=$

Kies **Opsl** \rightarrow en pas de definitie van $U1(N)$ aan zoals in het scherm hiernaast, het is even uitproberen hoe de cursor besturing binnen de verschillende niveaus van de vergelijking werkt. Verander de startwaarde van 49 naar 64 en kwadrateer het delen door $U1(N-1)$.

Weerg. Rij symb. 11:22

$\sqrt{U1(1)=\frac{1}{7}}$ U1(2)=

$U1(N)=\frac{U1(N-1)+\frac{49}{U1(N-1)}}{2}$

Optie: U1(N) Start Nr: 1

Functie invoeren

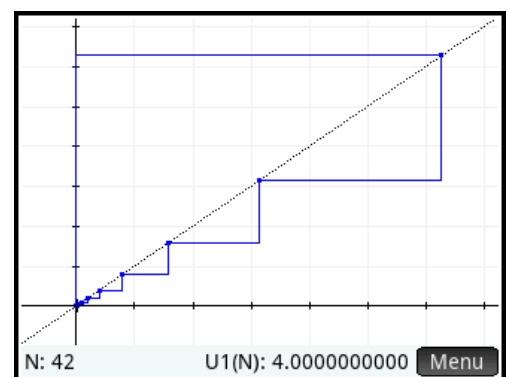
$U1(N-1)+\frac{64}{U1(N-1)^2}$

(N-2) (N-1) N U1 Annuler OK

Het verschil wordt dat het getal waarvan we de derdemachtwortel zoeken wordt gedeeld door het kwadraat van de startwaarde, als de startwaarde precies de derdemachtwortel is dan is de juiste waarde gevonden, zo niet wordt het gemiddelde van de startwaarde en de uitkomst van de deling de nieuwe startwaarde. Ook hier geldt dat iedere stap ons dichterbij de juiste waarde brengt.

Deze procedure is minder snel dan bij de tweedemacht wortel, zo zijn er voor de berekening van de derdemachtwortel van 64 (=4, want $4*4=16$ en $4*16=64$) maar liefst 43 stappen nodig voordat de machine op 4 afrondt zonder decimalen achter de komma.

In deze grafiek is ook fors uitgezoomd, de kleine markeringen langs de X- en Y-as staan bij ieder veelvoud van 250.



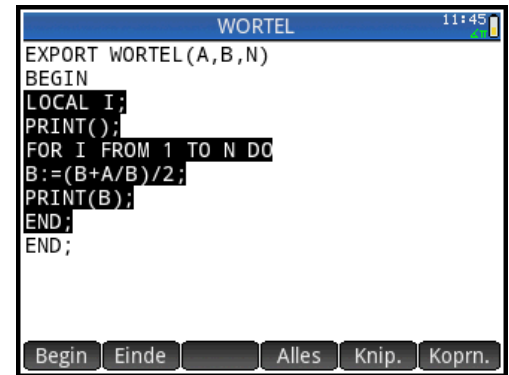
Ook het programma kan eenvoudig worden aangepast, eerst open ik het programma WORTEL, kies dan **Shift** en **View Copy**. Nu de cursor vooraan LOCAL plaatsen en op **Begin** drukken, verplaats de cursor tot het einde van de voorlaatste regel, dus na de puntkomma achter END; en kies **Einde**. Een deel van de programmatekst is nu geselecteerd, toets **Koprn.** (voor de zekerheid).

Nu het programma verlaten met **Esc** en op **Nieuw** drukken, hiermee opent een scherm waar de naam van het nieuwe programma kan worden ingevoerd, gebruik **ALPHA** **ALPHA** en typ QUBIC (okay dat is Engels maar wel een stuk korter dan „derdemachtwortel“), sluit het typen af met **ALPHA**. Kies **OK** en u kunt het programma invoeren, zet de cursor in de regel onder BEGIN en gebruik **Shift** **Menu Paste** om de eerder gekopieerde programmatekst te plakken. De eerste regel uit het popup scherm bevat de gevraagde programmatekst (als u tussendoor geen uitstapjes heeft gemaakt), bevestig met **Enter**.

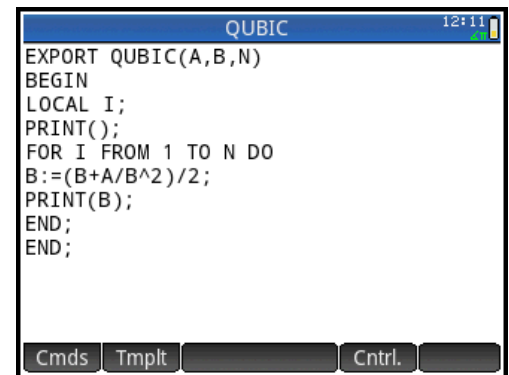
Nu hoeft u nog maar twee aanpassingen te doen:

1. Typ in de bovenste regel tussen de haakjes A,B,N
2. Verander in de regel $B:=(B+A/B)/2;$

in $B:=(B+A/B^2)/2$ dit kan door de cursor op de juiste plaats te zetten en l te gebruiken. Nu is het programma QUBIC al klaar, test het uit!



```
EXPORT WORTEL(A,B,N)
BEGIN
LOCAL I;
PRINT();
FOR I FROM 1 TO N DO
B:=(B+A/B)/2;
PRINT(B);
END;
END;
```



```
EXPORT QUBIC(A,B,N)
BEGIN
LOCAL I;
PRINT();
FOR I FROM 1 TO N DO
B:=(B+A/B^2)/2;
PRINT(B);
END;
END;
```

NB: Het schrijven van programma's kan ook met de PC of MAC. Installeer hiervoor de gratis HP Prime emulator software en de HP Aansluitkit, dan kunt u met het toetsenbord van uw computer de tekst intypen en vervolgens makkelijk overzetten naar uw rekenmachine.

- Pieter Schadron, april 2017