

Hoofdstuk 16: Programmeren

Kennismaking: het volume van een cylinder berekenen

Deze kennismaking is een beknopte inleiding. Lees het vervolg van dit hoofdstuk voor meer gedetailleerde informatie.

Een programma is een verzameling opdrachten die achtereenvolgens door de TI-84 Plus worden uitgevoerd, precies alsof u ze op het toetsenbord zou hebben ingevoerd. Schrijf een programma waarin u de waarden voor de straal R en de hoogte H van een cilinder moet opgeven en dat vervolgens het volume van de cilinder berekent.

1. Druk **PRGM** **▶** **▶** om het menu **PRGM NEW** op te roepen.



2. Druk **[ENTER]** om de optie **1:Create New** te kiezen. De Name= aanwijzer verschijnt op het scherm en het toetsenbord wordt in alpha-lock gezet. Druk **[C]** **[Y]** **[L]** **[I]** **[N]** **[D]** **[E]** **[R]** en druk vervolgens **[ENTER]** om **CYLINDER** als naam van het programma op te geven.

```
PROGRAM:CYLINDER
:█
```

U bevindt zich nu in het programmascherm. De dubbele punt (:) in de eerste kolom van de tweede regel geeft aan dat dit het begin van een opdrachtregel is.

3. Druk **[PRGM]** **[▶]** **2** om de optie **2:Prompt** in het menu **PRGM I/O** te kiezen. In de opdrachtregel wordt **Prompt** ingevoegd. Druk **[ALPHA]** **[R]** **[,]** **[ALPHA]** **[H]** om de namen voor de variabelen van de straal en de hoogte op te geven. Druk vervolgens **[ENTER]**.

```
PROGRAM:CYLINDER
:Prompt R,H
:█
```

4. Druk **[2nd]** **[π]** **[ALPHA]** **[R]** **[x²]** **[ALPHA]** **[H]** **[1]** **[ALPHA]** **[A]** **[ENTER]** om de uitdrukking $\pi R^2 H$ in te voeren en het resultaat in de variabele **V** op te slaan.

```
PROGRAM:CYLINDER
:Prompt R,H
:πR2H→V
:█
```

5. Druk **[PRGM]** **[▶]** **3** om de optie **3:Disp** in het menu **PRGM I/O** te kiezen. In de opdrachtregel wordt **Disp** ingevoegd. Druk **[2nd]** **[ALPHA]** **["]** **[V]** **[O]** **[L]** **[U]** **[M]** **[E]** **[_]** **[I]** **[S]** **["]** **[ALPHA]** **[,]** **[ALPHA]** **[V]** **[ENTER]** zodat het programma de tekst **VOLUME IS** op één regel en de berekende waarde **v** op de volgende regel weergeeft.

```
PROGRAM:CYLINDER
:Prompt R,H
:πR2H→V
:Disp "VOLUME IS
":V
:█
```

6. Druk **[2nd]** **[QUIT]** om terug te keren naar het basisscherm.
7. Druk **[PRGM]** om het menu **PRGM EXEC** op te roepen. Dit menu geeft een overzicht van de namen van de opgeslagen programma's.

```
EXEC EDIT NEW
1 CYLINDER
```

8. Druk **[ENTER]** om de tekst `prgmCYLINDER` op de huidige positie van de cursor in te voegen. (Indien **CYLINDER** niet het eerste item (**1**) in uw menu **PRGM EXEC** is, moet u de cursor eerst verplaatsen tot aan **CYLINDER** voor u **[ENTER]** drukt.)

```
Pr9mCYLINDER█
```

9. Druk **[ENTER]** om het programma uit te voeren. Voor de straal voert u de waarde `1.5` in en drukt u vervolgens **[ENTER]**. Voor de hoogte voert u de waarde `3` in en drukt u vervolgens **[ENTER]**. Op het uitleesscherm verschijnt nu de tekst `VOLUME IS`, gevolgd door de waarde in `v` en de melding `Done`.

```
Pr9mCYLINDER
R=21.5
H=23
VOLUME IS
      21.20575041
      Done
```

Herhaal de stappen 7 tot en met 9 en gebruik nu andere waarden voor **R** en **H**.

Programma's maken en verwijderen

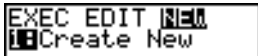
Wat is een programma?

Een programma is een reeks van één of meer opdrachtregels. Elke regel bevat één of meer instructies. Wanneer u een programma uitvoert, zal de TI-84 Plus elke instructie in elke opdrachtregel uitvoeren in precies de volgorde die u hebt gebruikt wanneer u deze hebt ingevoerd. De beperking van het aantal programma's en hun grootte is enkel afhankelijk van het beschikbare geheugen van de TI-84 Plus.

Een nieuw programma maken

Als u een nieuw programma wilt maken, moet u als volgt te werk gaan.

1. Druk **[PRGM]** **[↓]** om het menu **PRGM NEW** op te roepen.



2. Druk **[ENTER]** om de optie **1:Create New** te kiezen. De **Name=** aanwijzer verschijnt op het scherm en het toetsenbord wordt in alpha-lock gezet.
3. Druk op een lettertoets van A tot Z of θ om de eerste letter voor de naam van het nieuwe programma in te voeren.
Opmerking: de naam van het programma kan maximum acht tekens bevatten. Het eerste teken moet echter een alfabetisch teken van A tot Z of een θ zijn. Voor het tweede en de volgende tekens kunt u letters, cijfers of θ gebruiken.
4. U kunt verder maximum zeven letters, cijfers of θ -symbolen invoeren totdat u de volledige naam voor het nieuwe programma hebt opgegeven.
5. Druk vervolgens **[ENTER]**. Het programmascherm wordt nu weergegeven.

6. Voer één of meer programma-opdrachten in.
7. Druk **[2nd] [QUIT]** als u het programmascherm wilt verlaten en naar het basisscherm wilt terugkeren.

Geheugenbeheer en het wissen van een programma

Om te controleren of er voldoende geheugen is voor een programma dat u in wilt voeren:

1. Druk op **[2nd] [MEM]** om het menu **MEMORY** te openen.
2. Kies **2:Mem Mgmt/Del** om het menu **MEMORY MANAGEMENT/DELETE** te openen (hoofdstuk 18).
3. Kies **7:Prgm** om de **PRGM** editor te openen.

```
RAM FREE 19635
ARC FREE 847598
*PROGRAM1 3475
▶ PROGRAM2 2844
```

De TI-84 Plus drukt de geheugenhoeveelheid uit in bytes.

U kunt de beschikbare geheugenruimte op twee manieren vergroten. U kunt één of meer programma's wissen, of u kunt een aantal programma's archiveren.

De geheugenruimte vergroten door een bepaald programma te wissen:

1. Druk op **[2nd] [MEM]** en kies **2:Mem Mgmt/Del** uit het menu **MEMORY**.

```
MEMORY
1>About
2:Mem Mgmt/Del...
3:Clear Entries
4:ClrAllLists
5:Archive
6:UnArchive
7↓Reset...
```

2. Kies **7:Prgm** om de **PRGM** editor weer te geven (hoofdstuk 18).

```
RAM FREE 19635
ARC FREE 847598
*PROGRAM1 3475
▶ PROGRAM2 2844
```

3. Druk op en op om de selectiecursor (▶) naast het programma te zetten dat u wilt wissen, en druk vervolgens op . Het programma wordt uit het geheugen verwijderd.

Opmerking: u krijgt een melding waarin gevraagd wordt het verwijderen te bevestigen. Kies **2:yes** om door te gaan.

Om het scherm PRGM editor te verlaten zonder iets te verwijderen, drukt u op [QUIT]. Het basisscherm zal nu weer verschijnen.

De geheugenruimte vergroten door een programma te archiveren:

4. Druk op [MEM] en kies **2:Mem Mgmt/Del** uit het menu **MEMORY**.
5. Kies **2:Mem Mgmt/Del** om het menu **MEM MGMT/DEL** te openen.
6. Kies **7:Prgm...** om het menu **PRGM** te openen.

```
RAM FREE 22464
ARC FREE 844751
*PROGRAM1 3475
▶*PROGRAM2 2844
```

7. Druk op `ENTER` om het programma te archiveren. Links van het programma zal een sterretje verschijnen om aan te geven dat het een gearchiveerd programma betreft.

Om in dit scherm een programma uit het archief te halen, zet u de cursor naast het gearchiveerde programma en drukt u op `ENTER`. Het sterretje zal verdwijnen.

Opmerking: een gearchiveerd programma kan niet worden bewerkt of uitgevoerd. Om een gearchiveerd programma te kunnen bewerken of uitvoeren, moet u het eerst uit het archief halen.

Opdrachten invoeren en programma's uitvoeren

Een programma-opdracht invoeren

In de opdrachtregel kunt u elke instructie of uitdrukking invoeren die u ook in het basisscherm kunt ingeven. In het programmascherm begint elke nieuwe opdrachtregel met een dubbele punt (:). Als u verscheidene instructies of uitdrukkingen in één opdrachtregel wilt invoeren, moet u een dubbele punt als scheidingsteken gebruiken.

Opmerking: een opdrachtregel kan meer tekens bevatten dan horizontaal op het scherm kunnen worden weergegeven; in dit geval zal de opdrachtregel op de volgende schermregel doorlopen.

Terwijl u zich in het programmascherm bevindt, kunt u menu's oproepen en hierin opties kiezen. U kunt steeds vanuit een menu op de volgende twee manieren naar het programmascherm terugkeren:

- kies een item in het menu, waardoor dit in de actuele opdrachtregel wordt ingevoerd;
- druk **CLEAR**.

Als u klaar bent met het invoeren van een opdrachtregel drukt u **ENTER**. De cursor wordt automatisch naar de volgende opdrachtregel verplaatst.

In programma's kunt u gebruik maken van variabelen, lijsten, matrices en tekenreeksen die in het geheugen worden opgeslagen. Wanneer een programma een nieuwe waarde toekent aan een variabele, lijst, matrix of tekenreeks, wordt tijdens de uitvoering van het programma de waarde in het geheugen gewijzigd.

U kunt in een programma ook andere programma's oproepen in de vorm van een subroutine.

Een programma uitvoeren

Als u een programma wilt uitvoeren, moet u ervoor zorgen dat u zich in het basisscherm bevindt en de regel leeg is. Ga nu als volgt te werk.

1. Druk **PRGM** om het menu **PRGM EXEC** op te roepen.
2. Kies in het menu **PRGM EXEC** een programmaam. In het basisscherm wordt automatisch **prgmnaam** ingevoerd (bijvoorbeeld **prgmCYLINDER**).
3. Druk nu **ENTER** om dit programma uit te voeren. Terwijl het programma wordt uitgevoerd, verschijnt de aanduiding op het scherm dat de rekenmachine aan het berekenen is (bezig-aanduiding).

De variabele Last Answer (**Ans**) (laatste resultaat) verandert in de loop van de uitvoering van het programma, zodat u steeds de opdracht **Ans** in een opdrachtregel kunt invoeren.

De variabele Last Entry (laatste invoer) verandert echter niet wanneer de opdrachten worden uitgevoerd (hoofdstuk 1).

De TI-84 Plus zal tijdens de uitvoering van het programma de opdrachten op fouten controleren. De foutcontrole wordt dus niet automatisch uitgevoerd op het ogenblik dat u de instructies en uitdrukkingen van het programma invoert.

Een programma onderbreken

Als u de uitvoering van een programma wilt onderbreken, moet u **[ON]** drukken. Vervolgens verschijnt het menu **ERR:BREAK** op het scherm.

- Als u wilt terugkeren naar het basisscherm, kiest u de optie **1:Quit**.
- Als u wilt overschakelen naar de opdrachtregel waar u het programma hebt onderbroken, kiest u de optie **2:Goto**.

Programma's bewerken

Een programma bewerken

Als u een opgeslagen programma wilt wijzigen of bewerken, moet u als volgt te werk gaan.

1. Druk **[PRGM]** **[▶]** om het menu **PRGM EDIT** op te roepen.

2. Kies in het menu **PRGM EDIT** een programma-naam. De eerste zeven regels van het programma worden weergegeven.

Opmerking: in het programmascherf verschijnt geen ↓-symbool om aan te geven dat een programma meer lijnen bevat dan op dat ogenblik op het scherm kunnen worden weergegeven.

3. Breng de wijzigingen in de opdrachtregels van het programma aan.
 - Verplaats de cursor naar de gewenste positie en verwijder, vervang of voeg de vereiste instructie of uitdrukking in.
 - Druk **CLEAR** als u alle programma-opdrachten op de actuele opdrachtregel wilt wissen (de dubbele punt aan het begin van de opdrachtregel wordt niet gewist) en voer een nieuwe programma-opdracht in.

Opmerking: als u de cursor naar het begin van een opdrachtregel wilt verplaatsen, moet u **2nd** **←** drukken; om deze naar het einde van de opdrachtregel te verplaatsen, drukt u **2nd** **→**. Als u snel zeven opdrachtregels omlaag wilt schuiven, drukt u **ALPHA** **↓**; om snel zeven opdrachtregels omhoog te schuiven, drukt u **ALPHA** **↑**.

Opdrachtregels invoeren en verwijderen

Wanneer u een nieuwe opdrachtregel om het even waar in het programma wilt invoegen, moet u eerst de cursor verplaatsen naar de positie waar u de nieuwe regel wilt invoegen, **2nd** **INS** drukken en vervolgens **ENTER**. De dubbele punt vooraan geeft aan dat dit een nieuwe regel is.

Wanneer u een bestaande opdrachtregel wilt verwijderen, moet u eerst de cursor naar deze regel verplaatsen, **CLEAR** drukken om alle instructies en uitdrukkingen in deze regel te wissen en vervolgens **DEL** indrukken om de (lege) opdrachtregel, inclusief de dubbele punt, te verwijderen.

Programma's kopiëren en een nieuwe naam geven

Een programma kopiëren en een nieuwe naam geven

Wanneer u alle opdrachten uit één programma wilt kopiëren naar een nieuw programma, moet u de stappen 1 tot en met 5 van de sectie Een nieuw programma maken uitvoeren, en vervolgens als volgt te werk gaan.

1. Druk **[2nd] [RCL]**. Op de laatste regel van het nieuwe programma verschijnt in het programmascherm de tekst **Rcl** (hoofdstuk 1).
2. Druk nu **[PRGM] [↓]** om het menu **PRGM EXEC** op te roepen.
3. Kies in dit menu de naam van het programma. In het programmascherm wordt in de laatste regel automatisch **prgmnaam** ingevoegd.
4. Druk **[ENTER]**. Alle opdrachtregels van het door u gekozen programma worden nu in het nieuwe programma gekopieerd.

Het kopiëren van programma's biedt tenminste twee belangrijke en nuttige voordelen:

- u kunt een programmasjabloon maken voor de reeksen instructies die u vaak wilt gebruiken;
- u kunt een nieuwe naam toekennen aan een programma door de inhoud ervan te kopiëren naar een nieuw programma.

Opmerking: u kunt met behulp van de opdracht RCL (hoofdstuk 1) alle opdrachten van een programma ook kopiëren naar een reeds bestaand programma.

De menu's PRGM EXEC en PRGM EDIT doorlopen

De TI-84 Plus zorgt ervoor dat alle items in de menu's **PRGM EXEC** en **PRGM EDIT** automatisch in oplopende, alfanumerieke volgorde worden gerangschikt. Deze menu's kennen enkel aan de eerste 10 items een label toe van 1 tot 9, gevolgd door 0.

Als u automatisch wilt overschakelen naar het eerste item waarvan de programmaam begint met een specifiek alfabetisch letterteken of een θ , moet u **ALPHA** [*een lettertoets van A tot Z of θ*] drukken.

Opmerking: als u in deze menu's van het eerste item wilt overschakelen naar het laatste, drukt u **▲**. Om van het laatste item over te schakelen naar het eerste, moet u **▼** indrukken. Als u snel zeven items in het menu omlaag wilt schuiven, drukt u **ALPHA ▼**. Om snel zeven items in het menu omhoog te schuiven, drukt u **ALPHA ▲**.

PRGM CTL-instructies (programmabesturing)

Het menu PRGM CTL

U kunt het menu **PRGM CTL** (programmabesturing) alleen oproepen door in het programmascherm **PRGM** te drukken.

CTRL I/O EXEC

1: If	Begin van een voorwaardelijke test.
3: Else	Voert de opdrachten uit indien If waar is.
2: Then	Voert de opdrachten uit indien If onwaar is.
4: For (Begin van een (onvoorwaardelijke) lus die telkens wordt verhoogd.
5: While	Begin van een voorwaardelijke lus.
6: Repeat	Begin van een voorwaardelijke lus.
7: End	Geeft het einde van een programmagedeelte aan.
8: Pause	Onderbreekt de uitvoering van een programma.
9: Lbl	Definieert een label.
0: Goto	Om naar een specifiek label over te schakelen.
A: IS> (Verhoogt en slaat over als groter dan.
B: DS<(Vermindert en slaat over als kleiner dan.
C: Menu(Het bepalen van items in een menu en sprongen.

D:prgm	Voert een programma als subroutine uit.
E:Return	Einde van een subroutine en terugkeren naar hoofdprogramma.
F:Stop	Stopt de uitvoering van het programma.
G:DelVar	In het programma wordt een variabele gewist.
H:GraphStyle(Begin van een voorwaardelijke test.

Deze instructies bepalen het verloop van het programma dat wordt uitgevoerd. U kunt deze gebruiken om op een eenvoudige manier een reeks opdrachten te herhalen of over te slaan in de loop van de uitvoering van het programma. Wanneer u een item in dit menu kiest, wordt deze opdracht naam automatisch op de positie van de cursor in de opdrachtregel van het programma ingevoegd.

Als u wilt terugkeren naar het programmascherm zonder een item in het menu te kiezen, drukt u CLEAR.

Het programma-verloop besturen

Aan de hand van de instructies voor de programmacontrole kunt u aangeven welke opdracht in het programma als volgende opdracht door de TI-84 Plus zal worden uitgevoerd. Met de opdrachten **If**, **While** en **Repeat** stuurt u het programma afhankelijk van een bepaalde voorwaarde. In deze voorwaarden wordt vaak gebruik gemaakt van relationele of Boolese test (hoofdstuk 2), bijvoorbeeld:

If A<7:A+1>A

of

If N=1 and M=1:Goto Z.

If

If wordt gebruikt voor een test en een sprong. Indien *voorwaarde* onwaar is (waarde is nul), dan wordt de *opdracht* die onmiddellijk volgt na de If-instructie overgeslagen. Indien *voorwaarde* waar is (waarde is niet nul), dat wordt de volgende *opdracht* uitgevoerd. De If-instructies kunnen worden genest.

```
:if voorwaarde  
:opdracht (indien waar)  
:opdracht
```

Programma

```
PROGRAM:COUNT  
:0→A  
:Lb1 Z  
:A+1→A  
:Disp "A IS",A  
:If A≥2  
:Stop  
:Goto Z
```

Resultaat

```
PrgrMCOUNT  
A IS  
A IS 1  
Done 2
```

If-Then

Then is een instructie die u kunt gebruiken na een If-instructie om een reeks *opdrachten* uit te voeren indien *voorwaarde* waar is (waarde is niet nul). Met de **End**-instructie geeft u het einde van deze reeks *opdrachten* aan.

```
:if voorwaarde  
:Then
```

```
:opdracht (indien waar)
:opdracht (indien waar)
:End
:opdracht
```

Programma

```
PROGRAM:TEST
:1→X:10→Y
:If X<10
:Then
:2X+3→X
:2Y-3→Y
:End
:Disp X,Y
```

Resultaat

```
PrgrMTEST
                    5
                    17
                    Done
```

If-Then-Else

Else is een instructie die u kunt gebruiken na een **If-Then**-instructie om een reeks *opdrachten* uit te voeren indien *voorwaarde* onwaar is (waarde is nul). Met de **End**-instructie geeft u het einde van deze reeks *opdrachten* aan.

```
:if voorwaarde
:Then
:opdracht (indien waar)
:opdracht (indien waar)
:Else
:opdracht (indien onwaar)
:opdracht (indien onwaar)
```

:End

:opdracht

Programma

```
PROGRAM:TESTELSE  
:Input "X=",X  
:If X<0  
:Then  
:X2+Y  
:Else  
:X+Y  
:End
```

Resultaat

```
PrgrMTESTELSE  
X=5  
      (5 5)  
      Done  
X=-5  
      (-5 25)  
      Done
```

```
:Disp {X,Y}
```

For(

For(is een instructie die u gebruikt voor programmalussen en tellers. Hiermee kunt u een *variabele* van *begin* tot *einde* verhogen met *stapgrootte*. *stapgrootte* is optioneel (als u deze niet opgeeft, is de *stapgrootte* = 1) en kan zelfs een negatief getal zijn (*einde*<*begin*). *einde* is een maximale of minimale waarde die niet mag overschreden worden. Met de **End**-instructie geeft u het einde van de programmalus aan. De **For**(-lussen kunnen worden genest.

:For(*variabele,begin,einde*[,*stapgrootte*])

:opdracht (zolang *einde* niet wordt overschreden)

:opdracht (zolang *einde* niet wordt overschreden)

:End
:opdracht

Programma

```
PROGRAM: SQUARE  
: For(A, 0, 8, 2)  
: Disp A2  
: End
```

Resultaat

```
PrgrmSQUARE  
0  
4  
16  
36  
64  
Done
```

While

While is een instructie waarmee u een reeks *opdrachten* kunt uitvoeren zolang *voorwaarde* waar is. *voorwaarde* is vaak een relationele test (hoofdstuk 2). De *voorwaarde* wordt getest wanneer de **While**-instructie wordt ontmoet. Indien *voorwaarde* waar is (waarde is niet nul), dan zal het programma een reeks *opdrachten* uitvoeren. De **End**-instructie geeft het einde van deze reeks opdrachten aan. Wanneer *voorwaarde* onwaar is (waarde is nul), zal het programma elke *opdracht* die volgt na de **End**-instructie uitvoeren. De **While**-instructies kunnen worden genest.

:**While** *voorwaarde*
:opdracht (zolang *voorwaarde* waar is)
:opdracht (zolang *voorwaarde* waar is)

:End

:opdracht

Programma

```
PROGRAM: LOOP
: 0 → I
: 0 → J
: While I < 6
: J + 1 → J
: I + 1 → I
: End
: Disp "J=", J
```

Resultaat

```
Prgrm LOOP
J =
6
Done
```

Repeat

Repeat is een instructie waarmee u een reeks *opdrachten* kunt uitvoeren totdat *voorwaarde* waar is (waarde is niet nul). Deze instructie heeft dezelfde functie als de **While**-instructie, maar hier wordt *voorwaarde* getest op het ogenblik dat de **End**-instructie wordt ontmoet; op deze manier wordt de reeks *opdrachten* altijd tenminste één keer uitgevoerd. De **Repeat**-instructies kunnen worden genest.

:**Repeat** *voorwaarde*

:*opdracht* (totdat *voorwaarde* waar is)

:*opdracht* (totdat *voorwaarde* waar is)

:End

:opdracht

Programma

```
PROGRAM:RLOOP
:0→I
:0→J
:Repeat I≥6
:J+1→J
:I+1→I
:End
:Disp "J=",J
```

Resultaat

```
PrgrmRLOOP
J=
6
Done
```

End

End is een instructie die het einde van een reeks *opdrachten* aangeeft. U moet steeds een **End**-instructie invoegen aan het einde van elke **For**-, **While**- of **Repeat**-lus. U moet ook steeds een **End**-instructie invoegen aan het einde van elke **If-Then**- en elke **If-Then-Else**-instructiereeks.

Pause

Pause is een instructie waarmee u de normale uitvoering van het programma kunt onderbreken zodat u de resultaten of grafieken kunt bekijken. Tijdens het pauzeren ziet u rechts bovenaan het scherm een stippelijntje als pauzesymbool. Druk **[ENTER]** als u de uitvoering van het programma gewoon wilt verderzetten.

- **Pause** zonder argument onderbreekt tijdelijk de uitvoering van het programma. Indien de **DispGraph**- of **Disp**-instructie wordt uitgevoerd, dan wordt het desbetreffende scherm weergegeven.

- **Pause** met *waarde* onderbreekt de uitvoering van het programma en toont *waarde* in het huidige basisscherm. U kunt *waarde* over het scherm laten schuiven.

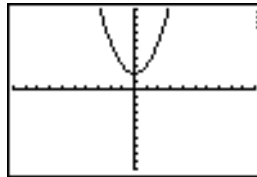
Pause [*waarde*]

Programma

```
PROGRAM: PAUSE
:10→X
:"X2+2"→Y1
:Disp "X=",X
:Pause
:DispGraph
:Pause
:Disp
```

Resultaat

```
PrgmPAUSE
X=
10
```



```
PrgmPAUSE
X=
10
Done
```

Lbl, Goto

Lbl (label) en **Goto** (ga naar) zijn instructies die u samen kunt gebruiken om sprongen binnen het programma uit te voeren.

Lbl bepaalt het *label* voor een opdracht. Een *label* kan bestaan uit één of twee tekens (**A** tot en met **Z**, **0** tot en met **99**, of θ).

Lbl *label*

Goto zorgt ervoor dat het programma verder wordt uitgevoerd vanaf *label* wanneer de **Goto**-instructie wordt ontmoet.

Goto *label*

Programma

```
PROGRAM: CUBE
:Lbl 99
:Input A
:If A≥100
:Stop
:Disp A³
:Pause
:Goto 99
```

Resultaat

```
PrgrnCUBE
?2                8
?3                27
?105              Done
```

IS>(

IS>((verhoog-en-sla-over) telt 1 op bij de *variabele*. Indien het resultaat $>$ is dan *waarde* (dit kan een uitdrukking zijn), wordt de volgende *opdracht* overgeslagen; indien het resultaat \leq is vergeleken met *waarde*, wordt de volgende *opdracht* uitgevoerd. De *variabele* kan geen systeemvariabele zijn.

:IS>(*variabele,waarde*)

:opdracht (indien resultaat \leq *waarde*)

:opdracht (indien resultaat $>$ *waarde*)

Programma

```
PROGRAM: ISKIP
:7→A
:IS>(A,6)
:Disp "NOT > 6"
:Disp "> 6"
```

Resultaat

```
PrgrmISKIP
> 6
Done
```

Opmerking: IS>(is geen lusiinstructie.

DS<(

DS<((verminder-en-sla-over) vermindert de *variabele* met 1. Indien het resultaat < is dan *waarde* (dit kan een uitdrukking zijn), wordt de volgende *opdracht* overgeslagen; indien het resultaat \geq is vergeleken met *waarde*, wordt de volgende *opdracht* uitgevoerd. De *variabele* kan geen systeemvariabele zijn.

:DS<(*variabele,waarde*)

:opdracht (indien resultaat \geq *waarde*)

:opdracht (indien resultaat < *waarde*)

Programma

```
PROGRAM:DSKIP
:1→A
:DS<(A,6)
:DISP "> 6"
:DISP "NOT > 6"
```

Resultaat

```
PrgrmDSKIP
NOT > 6           Done
```

Opmerking: DS<(is geen lusiinstructie.

Menu(

Menu(is een instructie waarmee u binnen het programma een sprong kunt uitvoeren. Wanneer in de loop van de uitvoering van het programma **Menu(** wordt ontmoet, wordt het menuscherm getoond met hierin de specifieke menu-opties, wordt de pauze-aanduiding weergegeven en de uitvoering van het programma onderbroken totdat u een optie in het desbetreffende menu hebt gekozen.

De menu*titel* moet u tussen aanhalingstekens (") invoeren, gevolgd door maximum zeven paren van menu-opties. Elk paar omvat een argument *tekst* (eveneens tussen aanhalingstekens) die op het scherm zal verschijnen als een menu-optie en het bijbehorende argument *label* voor de positie in het programma waarnaar wordt gesprongen wanneer u de overeenkomstige menu-optie hebt gekozen.

Menu("titel","tekst1",label1,"tekst2",label2, . . .)

Programma

```
PROGRAM: TOSSDICE
:Menu("TOSS DICE
", "FAIR DICE",A,
"WEIGHTED DICE",
B)
```

Resultaat

```
TOSS DICE
1:FAIR DICE
2:WEIGHTED DICE
```

De uitvoering van het programma wordt onderbroken totdat u de optie 1 of 2 kiest. Wanneer u bijvoorbeeld de optie 2 kiest, zal het menu verdwijnen en wordt de uitvoering van het programma verdergezet vanaf het label **Lbl B**.

prgm

Met de instructie **prgm** kunt u andere programma's uitvoeren als subroutines van het huidige programma. Als u de instructie **prgm** kiest, wordt deze automatisch op de positie van de cursor ingevoegd. Voer de lettertekens in voor de *naam* van het programma. Als u gebruik maakt van de instructie **prgm**, stemt dit overeen met het kiezen van bestaande programma's in het menu **PRGM EXEC**; hiermee kunt u echter een naam opgeven van een programma dat u nog niet hebt gemaakt.

prgm*naam*

Opmerking: als u de instructie RCL gebruikt, kunt u de naam van de subroutine niet invoeren. U moet in dit geval de naam automatisch invoegen door een keuze te maken in het menu PRGM EXEC.

Return

Return is een instructie waarmee de subroutine wordt verlaten en het hoofdprogramma verder wordt uitgevoerd, zelfs indien deze in geneste lussen voorkomt. Eventuele lussen worden beëindigd. Het programma veronderstelt aan het einde van elk programma, dat als een subroutine wordt aangeroepen, een impliciete **Return**-instructie. Wanneer in de loop van de uitvoering een **Return**-instructie wordt ontmoet binnen het hoofdprogramma, zal het programma stoppen en keert u terug naar het basisscherm.

Stop

Stop is een instructie waarmee u de uitvoering van een programma kunt beëindigen en vervolgens kunt terugkeren naar het basisscherm. Een **Stop**-instructie aan het einde van een programma mag u invoegen, dit is niet noodzakelijk.

DelVar

DelVar is een instructie waarmee u de inhoud van een *variabele* uit het geheugen kunt wissen.

DelVar *variabele*

```
PROGRAM:DELMATR
:DelVar [A]█
```

GraphStyle(

GraphStyle(is de instructie waarmee u kunt aangeven welke grafiekstijl bij het plotten moet worden toegepast. *functie#* is het nummer van de Y= functienaam in de huidige grafiekinstelling. *grafiekstijl* is een getal van 1 tot 7 dat overeenstemt met de grafiekstijl, zoals hieronder aangegeven.

1 = \ (lijn)

2 = ¶ (dik)

3 = ¶ (schaduw bovenaan)

4 = ¶ (schaduw onderaan)

5 = ¶ (pad)

6 = ¶ (actief)

7 = ` (punt)


GraphStyle(*functie#*,*grafiekstijl*)

Voorbeeld: **GraphStyle(1,5)** in de **Func** instelling stelt de grafiekstijl voor **Y1** in op (pad; 5).

Niet alle grafiekstijlen zullen echter beschikbaar zijn in alle grafiekinstellingen. Zie de tabel van grafiekstijlen in hoofdstuk 3 voor meer gedetailleerde informatie voor de verschillende grafiekstijlen.

PRGM I/O-instructies (Input/Output-instructies)

Het menu PRGM I/O

U kunt het menu **PRGM I/O** (programma-invoer/uitvoer) alleen oproepen door in het programmascherm **PRGM**  te drukken.

CTRL I/O EXEC

1:Input	van de cursor.
2:Prompt	Vraagt om de invoer van waarde voor een variabele.
3:Disp	Toont een tekst, waarde of het basisscherm.
4:DispGraph	Toont de huidige grafiek.
5:DispTable	Toont de huidige tabel.
6:Output (Toont tekst op een specifieke positie.
7:getKey	Wacht totdat een toets wordt ingedrukt.
8:ClrHome	Wist het uitleesscherm.
9:ClrTable	Wist de huidige tabel.
0:GetCalc (Haalt een variabele op uit een andere TI-84 Plus.
A:Get (Haalt een variabele op uit CBL 2™/CBL™ of CBR™.
B:Send (Zendt een variabele naar CBL 2/CBL of CBR.

Met deze instructies kunt u de in- en uitvoer van een programma tijdens de uitvoering ervan beheren. Op deze manier kunt waarden invoeren en resultaten op het scherm weergeven in de loop van de uitvoering van het programma.

Als u wilt terugkeren naar het programmascherm zonder een item in het menu te kiezen, drukt u **[CLEAR]**.

Een grafiek tonen met de instructie Input

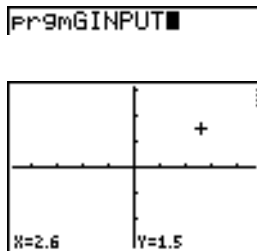
De instructie **Input** zonder argument geeft de huidige grafiek weer. U kunt de vrij beweegbare cursor verplaatsen, waardoor de coördinaten X en Y worden bijgewerkt. Op het scherm verschijnt dan het pauzesymbool. Druk **[ENTER]** als u de uitvoering van het programma wilt verderzetten.

Input

Programma

```
PROGRAM:GINPUT
:FnOff
:ZDecimal
:Input
:Disp X,Y
```

Resultaat



```
Pr9mGINPUT
      2.6
      1.5
Done
```

Een waarde in een variabele opslaan met de instructie Input

De instructie **Input** gevolgd door *variabele* toont een ? (vraagteken) als aanwijzer in de loop van de uitvoering van het programma. De *variabele* kan een reëel getal, een complex getal, een lijst, matrix, tekenreeks of Y= functie zijn. Wanneer het programma wordt gestart en dit teken verschijnt, voert u een waarde in (dit kan een uitdrukking zijn) en druk vervolgens **ENTER**. Deze waarde wordt geëvalueerd en in *variabele* opgeslagen. De uitvoering van het programma wordt verdergezet.

Input [*variabele*]

U kunt maximum 16 tekens *tekst* of de inhoud van **Strn** (een tekenreeksvariabele) als aanwijzer op het scherm weergegeven. Wanneer het programma wordt gestart en deze tekst verschijnt, voert u een waarde in na de aanwijzer en drukt u vervolgens **ENTER**. Deze waarde wordt dan in de *variabele* opgeslagen. De uitvoering van het programma wordt verdergezet.

Input ["*tekst*",*variabele*]

Input [**Strn**,*variabele*]

Programma

```
PROGRAM:HINPUT
:Input A
:Input L1
:Input "Y1=",Y1
:Input "DATA=",L
DATA
:Disp Y1(A)
:Disp Y1(L1)
```

```
:Disp Y1(LDATA)
```

Resultaat

```
PrgrMHINPUT
?2

?(1,2,3)
Y1="2X+2"
DATA={4,5,6}
      6
      {4 6 8}
      {10 12 14}
      Done
```

Opmerking: wanneer een programma gedurende de uitvoering ervan door middel van een aanwijzer vraagt om de invoer van lijsten en uitdrukkingen, moet u ook de accolades ({}) voor en na de items in de lijst en de aanhalingstekens voor en na de uitdrukkingen invoeren.

Prompt

Tijdens de uitvoering van het programma, geeft de instructie **Prompt** elke *variabele*, één na één, weer op het scherm, gevolgd door =?. Na elke aanwijzer moet u dan een waarde of een uitdrukking voor elke *variabele* invoeren en vervolgens **[ENTER]** indrukken. Deze waarden worden opgeslagen en de uitvoering van het programma wordt verdergezet.

Prompt *variabeleA[,variabeleB,...,variabele n]*

Programma

```
PROGRAM:WINDOW
:Prompt Xmin
:Prompt Xmax
:Prompt Ymin
:Prompt Ymax
```

Resultaat

```
PrgrmWINDOW
Xmin=?-10
Xmax=?10
Ymin=?-3
Ymax=?3
Done
```

Opmerking: met de instructie **Prompt** kunt u geen Y= functies invoeren.

Het basisscherm tonen

De instructie **Disp** (tonen) zonder een argument toont het basisscherm. Als u tijdens de uitvoering van het programma het basisscherm wilt zien, moet u de **Disp**-instructie gevolgd door een **Pause**-instructie gebruiken.

Disp

Waarden en meldingen tonen

De instructie **Disp** met één of meer *waarden* toont deze waarden.

Disp [*waardeA,waardeB,waardeC,...,waarde n*]

- Indien *waarde* een variabele is, wordt de huidige waarde van deze variabele weergegeven.
- Indien *waarde* een uitdrukking is, wordt deze eerst geëvalueerd en zal het resultaat aan de rechterkant op de volgende regel verschijnen.
- Indien *waarde* een tekenreeks tussen aanhalingstekens is, zal deze aan de linkerkant van de huidige regel op het scherm verschijnen. → kan niet als teken in de tekst worden gebruikt.

Programma

```
PROGRAM:A  
:Disp "THE ANSWE  
R IS ",π/2
```

Resultaat

```
PrgrMA  
THE ANSWER IS  
1.570796327  
Done
```

Indien de **Disp**-instructie wordt gevolgd door de **Pause**-instructie, zal de uitvoering van het programma tijdelijk worden gepauzeerd zodat u het scherm rustig kunt bekijken.

Wanneer u vervolgens de uitvoering van het programma wilt verderzetten, drukt u

ENTER.

Opmerking: indien een matrix of een lijst te groot is om volledig op het scherm te worden weergegeven, verschijnen er in de laatste kolom weglatingstekens (...), hoewel u de

matrix of lijst niet kunt verschuiven. Als u toch de matrix of lijst wilt verschuiven, moet u de instructie **Pause** *waarde* gebruiken.

DispGraph

Met de instructie **DispGraph** (grafiek tonen) kunt u de huidige grafiek op het scherm weergeven. Indien de **DispGraph**-instructie wordt gevolgd door de **Pause**-instructie, zal de uitvoering van het programma tijdelijk worden gepauzeerd zodat u het scherm rustig kunt bekijken. Wanneer u vervolgens de uitvoering van het programma wilt verderzetten, drukt u **[ENTER]**.

DispTable

Met de instructie **DispTable** (tabel tonen) kunt u de huidige tabel op het scherm weergeven. De uitvoering van het programma zal tijdelijk worden gepauzeerd zodat u het scherm rustig kunt bekijken. Wanneer u vervolgens de uitvoering van het programma wilt verderzetten, drukt u **[ENTER]**.

Output(

De instructie **Output(** toont *tekst* of *waarde* op het huidige basisscherm vanaf *rij* (1 tot en met 8) en *kolom* (1 tot en met 16), waardoor eventuele tekens op deze regel worden overschreven.

Opmerking: u kunt het best de instructie **ClrHome** gebruiken voor u de instructie **Output(** in het programma invoert. Opmerking:

Uitdrukkingen worden eerst geëvalueerd en de waarden worden vervolgens op het scherm weergegeven, rekening houdend met de huidige **MODE**-instellingen. Matrices

worden weergegeven zoals ze werden ingevoerd en als een lijn te lang is om op het scherm te worden weergegeven, gaat deze op de volgende regel verder. → kan niet als teken in de tekst worden gebruikt.

Output(*rij*,*kolom*,"*tekst*")

Output(*rij*,*kolom*,*waarde*)

Programma

```
PROGRAM: OUTPUT
: 3+5→B
: ClrHome
: Output(5, 4, "ANS
WER: ")
: Output(5, 12, B)
```

Resultaat

```
ANSWER: 8
```

Als u de instructie **Output**(gebruikt wanneer het scherm door middel van **Horiz** wordt gesplitst, dan is de maximale waarde voor *lijn* 4. Als u de instructie **Output**(gebruikt wanneer het scherm door **G-T** wordt gesplitst, is de maximale waarde voor *rij* 8 en de maximale waarde voor *kolom* 16. Dit zijn dezelfde waarden als voor een **Full** (volledig) scherm.

getKey

De instructie **getKey** heeft als resultaat een getal dat overeenstemt met de laatst ingedrukte toets, overeenkomstig het toetsendiagram. Indien er geen toets werd ingedrukt, dan is het resultaat van **getKey** 0. U kunt de instructie **getKey** in lussen gebruiken om de besturing

van het programma in functie van de toetsaanslagen te stellen, bijvoorbeeld wanneer u videospelletjes maakt.

Programma

```
PROGRAM:GETKEY  
:While 1  
:getKey→K  
:While K=0  
:getKey→K  
:End  
:Disp K  
:If K=105
```

```
:Stop  
:End
```

Resultaat

```
PRGMGETKEY  
41  
42  
43  
105  
Done
```

Opmerking: De toetsen **MATH**, **APPS**, **PRGM** en **ENTER** werden tijdens de uitvoering van het programma ingedrukt.

Opmerking: u kunt op elk ogenblik **ON** drukken om de uitvoering van het programma te stoppen.

Toetsendiagram van de TI-84 Plus

11	12	13	14	15
			25	
21	22	23	24	26
31	32	33	34	
41	42	43	44	45
51	52	53	54	55
61	62	63	64	65
71	72	73	74	75
81	82	83	84	85
91	92	93	94	95
	102	103	104	105

ClrHome, ClrTable

Met de instructie **ClrHome** (basisscherm wissen) kunt u ervoor zorgen dat tijdens de uitvoering van het programma het basisscherm wordt leeggemaakt.

De instructie **ClrTable** (tabel wissen) zorgt ervoor dat de waarden in het tabelscherm tijdens de uitvoering van het programma worden gewist.

GetCalc(

Met de instructie **GetCalc(** kunt u de inhoud van *variabele* op een andere TI-84 Plus inlezen en opslaan in *variabele* op de ontvangende TI-84 Plus. *variabele* kan een getal zijn,

een item in een lijst, een lijstnaam, een element uit een matrix, een naam van een matrix, een tekenreeks, een Y=-variabele, een grafiekdatabase of een afbeelding.

GetCalc(*variabele*)[*poortmarkering*])

De TI-84 Plus gebruikt standaard de USB-poort als deze aangesloten is. Als de USB-kabel niet aangesloten is, gebruikt de rekenmachine de I/O-poort. Als u de USB- of I/O-poort wilt specificeren, gebruikt u de volgende poortmarkeringsnummers:

poortmarkering=0 gebruik USB-poort indien aangesloten;

poortmarkering=1 gebruik USB-poort;

poortmarkering=2 gebruik I/O-poort

Opmerking: **GetCalc**(werkt niet tussen TI-82 en TI-83 Plus of TI-82 en TI-84 Plus rekenmachines.

Get(, Send(

De instructie **Get**(haalt gegevens op uit de CBL 2/CBL of CBR en slaat deze op in *variabele* op de ontvangende TI-84 Plus. *variabele* kan een reëel getal zijn, een item in een lijst, een lijstnaam, een element uit een matrix, een naam van een matrix, een tekenreeks, een Y=-variabele, een grafiekdatabase of een afbeelding.

Get(*variabele*)

Opmerking: wanneer u een programma van een TI-82 naar een TI-84 Plus stuurt, dat de instructie **Get**(bevat, dan zal de TI-84 Plus deze instructie interpreteren als de hierboven beschreven **Get**(-instructie. Met de **Get**(-instructie kunt u echter geen gegevens uit een andere TI-84 Plus ophalen. Hiervoor moet u gebruik maken van de instructie **GetCalc**(.

De instructie **Send**(zendt de inhoud van *variabele* naar de CBL 2/CBL of CBR. U kunt deze instructie niet gebruiken om gegevens naar een andere TI-84 Plus te zenden. *variabele* kan een reëel getal zijn, een item in een lijst, een lijstnaam, een element uit een matrix, een naam van een matrix, een tekenreeks, een Y=-variabele, een grafiekdatabase of een afbeelding, bijvoorbeeld als invoer van statistische gegevens. *variabele* kan een lijst van elementen zijn.

Send(*variabele*)

```
PROGRAM:GETSOUND
:Send( (3, .00025,
99, 1, 0, 0, 0, 0, 1) )
:Get(L1)
:Get(L2)
```

Opmerking: Dit programma haalt geluidsgegevens en tijdgegevens in seconden op uit de CBL 2™.

Opmerking: u kunt de instructies **Get**(, **Send**(en **GetCalc**(oproepen vanuit CATALOG als u deze in het basisscherm wilt uitvoeren (hoofdstuk 15).

Andere programma's oproepen als subroutines

Een programma vanuit een ander programma oproepen

Met de TI-84 Plus beschikt u over een functie waarmee u een opgeslagen programma als subroutine kunt oproepen vanuit een ander programma. Voer de naam van het programma dat u als subroutine wilt laten uitvoeren in een regel in.

U kunt de naam van een programma in de opdrachtregel op twee manieren invoeren.

- Druk **PRGM** **4** om het menu **PRGM EXEC** op te roepen en kies de naam van het programma. Op de huidige positie van de cursor wordt in de opdrachtregel **prgmnaam** automatisch ingevoegd.
- Kies in het menu **PRGM CTL** de optie **prgm** en voer vervolgens de naam van het programma in.

prgmnaam

Wanneer tijdens de uitvoering van het programma **prgmnaam** wordt ontmoet, dan wordt de eerste opdracht van het secundaire (opgeroepen) programma de volgende opdracht die zal worden uitgevoerd. Het programma keert terug naar de eerstvolgende opdracht in het oproepende (hoofd-)programma wanneer ofwel een **Return**-instructie wordt ontmoet, ofwel wanneer een impliciete **Return** wordt verondersteld aan het einde van het opgeroepen programma.

Hoofdprogramma

```
PROGRAM:VOLCYL
:Input "D=";D
:Input "H=";H
:prgmAREACIR
:A*H→V
:Disp V
```



Resultaat

```
prgmVOLCYL
D=4
H=5
62.83185307
Done
```

Subroutine ↓ ↑

```
PROGRAM:AREACIR
:D/2→R
:π*R²→A
:Return
```

Opmerkingen over het oproepen van andere programma's

Alle variabelen zijn globale variabelen.

Het *label* dat wordt gebruikt in de **Goto**- en **Lbl**-instructies is een lokale variabele binnen het programma waarin dit wordt toegekend. Het *label* dat in het ene programma wordt gedefinieerd, kunt u echter niet in het andere programma gebruiken. U kunt dus geen gebruik maken van de **Goto**-instructie om een sprong naar een *label* in een ander programma te maken.

Met de instructie **Return** kunt u een subroutine verlaten en terugkeren naar het oproepende programma, zelfs wanneer deze instructie zich in geneste lussen bevindt.

Een assembleertaalprogramma uitvoeren

Het is mogelijk programma's uit te voeren die voor de TI-84 Plus geschreven zijn in assembleertaal. Assembleertaalprogramma's zijn in het algemeen veel sneller en bieden meer controlemogelijkheden dan de via het toetsenbord ingevoerde programma's die u in de ingebouwde programma-editor kunt schrijven.

Opmerking: een assembleertaalprogramma heeft meer controle over de rekenmachine, wat echter ook inhoudt dat, indien uw assembleertaalprogramma (een) fout(en) bevat, dit ertoe kan leiden dat de rekenmachine alle gegevens, programma's en toepassingen die in het geheugen zijn opgeslagen reset en dus verliest.

Wanneer u een assembleertaalprogramma download, wordt het bij de andere programma's opgeslagen als een PRGM menu-optie. U kunt:

- Het programma verzenden met behulp van de communicatiepoort van de TI-84 Plus (hoofdstuk 19).

- Het programma wissen met behulp van het scherm **MEM MGMT DEL** (hoofdstuk 18).

Om een assembleertaalprogramma uit te voeren, is de syntaxis:

Asm(AssemblyProgramName)

Gebruik bij het schrijven van een assembleertaalprogramma de twee onderstaande instructies uit de **CATALOG** om het programma te identificeren en te compileren.

Instructies	Commentaar
AsmComp (<i>prgmASM1</i> , <i>prgmASM2</i>)	Compileert een in ASCII geschreven assembleertaalprogramma en slaat de hex versie op
AsmPrgm	Identificeert een assembleertaalprogramma; moet worden ingevoerd als eerste regel van een assembleertaalprogramma

Om een assembleerprogramma dat u heeft geschreven te compileren:

1. Voer de stappen voor het schrijven van een programma uit. Vergeet niet **AsmPrgm** als eerste regel van uw programma in te voeren.
2. Druk in het basisscherm op **[2nd]** [**CATALOG**] en kies vervolgens **AsmComp** om het in het scherm te plakken
3. Druk op **[PRGM]** om het menu **PRGM EXEC** weer te geven.
4. Kies het programma dat u wilt compileren. Het zal in het basisscherm geplakt worden.
5. Druk op **[,]** en kies vervolgens **prgm** uit de **CATALOG**
6. Toets de naam in die u heeft gekozen voor het the output-programma.

Opmerking: deze naam moet uniek zijn – het mag geen kopie van een bestaande programmaam zijn.

7. Druk op `]` om de invoer te voltooien.

De volgorde van de argumenten dient als volgt te zijn:

AsmComp(*prgmASM1*, *prgmASM2*)

8. Druk op `ENTER` om uw programma te compileren en het output-programma te genereren.