

In deze derde les van module 2 leer je over het gebruik van uitdrukkingen en het opslaan van waarden in variabelen binnen programma's.

**Doelen:**

- Leren over het programmeren van wiskundige uitdrukkingen Understand order of operations.
- Begrijpen wat het verschil is tussen wiskundige variabelen en variabelen in computerprogramma's.
- **Uitdrukkingen** uitwerken.
- De resultaten van uitdrukkingen **opslaan** in variabelen.

**Uitdrukkingen**

$A^2$  en  $A+B$  en vergelijkbare notaties worden *uitdrukkingen* genoemd.

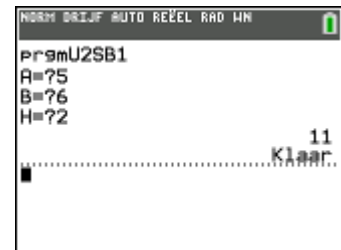
Uitdrukkingen vind je in wiskundige formules. Zo is bijvoorbeeld  $A = \frac{1}{2} \times B \times H$

de **formule** voor de oppervlakte van een driehoek. De **uitdrukking** is  $\frac{1}{2} \times B \times H$ .

Een programma werkt een uitdrukking uit gebruikmakend van de actuele waarden van alle variabelen en geeft het resultaat weer als een getal (numerieke waarde). Uitdrukkingen worden uitgewerkt volgens de *algebraïsche volgorde van bewerkingen*.

Probeer het programma rechts maar eens uit. Dit programma berekent de oppervlakte van een trapezium met twee evenwijdige zijden (kleine en grote basis) **A** en **B** en hoogte **H**.

Je kunt niet variabelen zoals **B1** en **B2** gebruiken. De rekenmachine berekent deze als de uitdrukkingen  $B \times 1$  en  $B \times 2$  en dit kan een fout veroorzaken als wanneer ze niet correct zijn gebruikt. Je kunt ook niet variabelen gebruiken met meer dan één letter zoals **AB**. Dit betekent, zoals eerder gezegd,  $A \times B$ . Dit heet ook wel *impliciete* vermenigvuldiging omdat het vermenigvuldigteken tussen de variabelen impliciet is (het wordt verondersteld).



**Wiskundige uitdrukkingen en uitdrukkingen op de computer**

Hoewel er veel overeenkomsten zijn in het *uiterlijk* van uitdrukkingen in de wiskunde en uitdrukkingen in computerprogramma's zijn er ook belangrijke verschillen. Het belangrijkste verschil is dat in wiskundige uitdrukkingen de variabelen staan voor 'onbekende' waarden en vervangen worden door getallen wanneer dat nodig is. In een uitdrukking in een computerprogramma zijn de variabelen *namen* voor getallen.

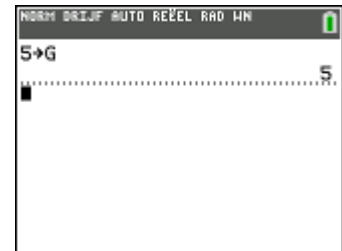
In de wiskunde gebruiken we formules om verbanden tussen grootheden zoals oppervlakte en lengte te beschrijven. In programma's gebruiken we uitdrukkingen om berekeningen uit te voeren en worden de waarden van de variabelen gebruikt om een uitkomst te berekenen. Wanneer we het programma *schrijven* typen we de uitdrukkingen in maar wanneer we het programma *uitvoeren* wordt de uitdrukking 'berekend' en wordt er een resultaat geproduceerd dat later gebruikt kan worden.

**Docenten Tip:** Een van de voor beginners, verwarrende opdrachten in een programma is  $x+1 \rightarrow x$ . Het is tamelijk duidelijk in de syntax van TI-Basic dat 1 wordt opgeteld bij de variabele  $x$  en dat dit vervolgens wordt opgeslagen in variabele  $x$ , dus de  $x$  aan de linkerkant en de  $x$  aan de rechterkant stellen verschillende waarden voor. Deze opdracht staat bekend als een ‘teller’ omdat elke keer dat deze opdracht wordt uitgevoerd (in een lus) de waarde van  $x$  met 1 wordt opgehoogd. Maar in andere programmeertalen zoals B.A.S.I.C en Lua (en vele andere) ziet deze opdracht er uit als  $x=x+1$ , wat duidelijk een onware bewering is in wiskundige zin; het is echter volstrekt juist binnen een programma.

**Waarden opslaan in variabelen: toewijzingsopdracht**

De **operator** `[STO]` wordt gebruikt om het resultaat van een uitdrukking **op te slaan** in (toe te wijzen aan) een variabele. Drukken op de toets `[STO]` geeft als resultaat altijd het symbool  $\rightarrow$

Nadat je op **E** hebt gedrukt toont het beginscherm het resultaat van de uitdrukking *en* bevat de variabele **G** nu de **5**. Voor het pijltje moet een **waarde** staan of een **uitdrukking** die een waarde produceert. Dit heet de *toewijzingsopdracht* omdat deze een waarde toewijst aan een variabele. Het symbool achter het pijltje moet een **variabele** zijn.



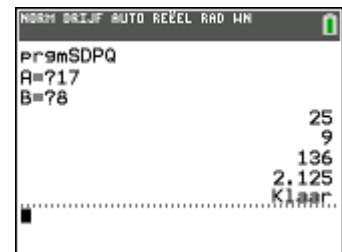
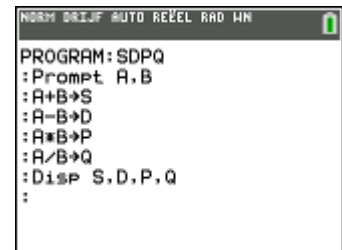
**Programmeren met toewijzingsopdrachten**

Laten we een programma schrijven dat je vraagt om twee getallen in te voeren, de som, het verschil, het product en het quotiënt ervan opslaat in vier variabelen en vervolgens het resultaat laat zien op het scherm.

1. Begin een nieuw programma. De naam van ons programma is **SDPQ**.
2. Vraag invoer voor twee variabelen **A** en **B**.
3. Sla de vier uitdrukkingen op in vier *andere* variabelen **S**, **D**, **P**, en **Q**.
4. Laat **S**, **D**, **P** en **Q** zien.
5. Voer het programma uit.

Voer een waarde in voor **A** en een andere waarde voor **B**.

Zorg dat je getallen gebruikt waarvoor je met zekerheid kunt vaststellen dat de berekeningen correct zijn uitgevoerd! Dit staat bekend als het ‘testen’ van het programma.



*Opmerking: De toewijzingsopdracht van TI Basic is uniek omdat de uitdrukking eerst komt, dan de ‘opslag’-operator en dan de variabele. Dit maakt dat de opdracht makkelijk van links naar rechts te lezen is. In de meeste andere programmeertalen is de volgorde omgekeerd, zoals  $S=A+B$ . Dit is ‘achterstevoren’ omdat de computer de uitdrukking aan de rechterkant eerst uitwerkt en deze vervolgens opslaat in de variabele aan de linkerkant.*

**Docenten Tip:** het opslaan van waarden in variabelen heeft veel voordelen. Het gebruik van variabelen maakt het eenvoudiger het resultaat weer te geven. Het maakt het ook mogelijk om de variabele te gebruiken in volgende berekeningen. Als je merkt dat je hetzelfde getal op veel verschillende plekken in je programma gebruikt dan kun je deze waarde opslaan in een variabele en deze overal gebruiken waar dat voorkomt in je programma.



## 10 minuten programmeren

### TI-84 PLUS SERIE

#### Verbeteringen aanbrengen

Je kunt het programma verbeteren door **Output**( te gebruiken in plaats van **Disp** om de oorspronkelijk ingevoerde waarden *én* de vier resultaten met *passende labels* weer te geven. Probeer het maar!

Rechts zie je een stukje van de programmacode en een scherm in het Engels van het gedeeltelijk voltooide programma dat wordt uitgevoerd. Er is hier nog wat werk te doen dus we laten het aan jou over om het programma af te maken.

Vergeet niet om een **Pause** opdracht toe te voegen aan het eind van de opdracht **Output** om te voorkomen dat het bericht 'Klaar het schermbeeld verpest.

Bedenk dat **TI-84 Plus** en de **TI-84 Plus C/CE** beginschermen hebben met verschillende afmetingen (en ook grafiekschermen met verschillende afmetingen) dus bereid dit goed voor.

Het beginscherm van de TI-84 Plus heeft 16 tekens per regel en 8 regels, terwijl het beginscherm van de TI-8 Plus C/CE 26 tekens per regel en 10 regels heeft.

## MODULE 2: OEFENBLAD 3

### DOCENTENHANDLEIDING

```
NORM DRJF AUTO REEL RAD MN
PROGRAM:SDPQ
:Prompt A,B
:A+B→S
:A-B→D
:A*B→P
:A/B→Q
:WdsHome
:Output(2,2,"A=")
:Output(2,4,A)■
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
A=18
B=7
SUM=25
```

**Docenten Tip:** Daag je leerlingen uit om te experimenteren met andere wiskundige formules en toewijzingsopdrachten te gebruiken om de resultaten op te slaan om het eenvoudiger te maken de resultaten weer te geven met **Display** of **Output**. In de toepassingsopdracht voor deze module krijgen de leerlingen drie wiskundige formules om mee te rekenen.