



In deze tweede les van Module 3 leer je een betere manier om structuren met voorwaarden te gebruiken en je leert je over samengestelde voorwaarden.

#### Doelen:

- De **If...Then...End**-structuur onderzoeken.
- Samengestelde voorwaarden maken met behulp van de logische operatoren.
- Een programma schrijven met de **If...Then...End** structuur dat de gebieden van het assenstelsel onderzoekt.

### De If...Then...End structuur

TI Basic bevat een unieke **If...Then** structuur die gebruik maakt van het belangrijke woord **End** om de opdrachten die het blok programmeercode vormen aan te sturen, de code wordt alleen uitgevoerd als de voorwaarde 'waar' (true) is. Het ziet er als volgt uit:

**If** <voorwaarde>

**Then**

<blok 'waar': voer deze opdrachten uit wanneer <voorwaarde> 'waar' is

**End**

*Opmerking:*

Na **If** komt er een <voorwaarde>.

**Then** staat direct onder **If**, op een verder lege regel.

Er staan één of meerdere 'opdrachten' in het <blok 'waar'>.

**End** geeft het einde van het **Then**-blok aan en de statements onder **End** worden wel uitgevoerd.

**End** is niet het einde van het programma! Het is alleen het einde van de **If...Then...End** structuur.

```

NORM DRIJF AUTO REEEL RAD MN
PROGRAM: IFTHEN
: Input
: If X>0 en Y>0
: Then
: Disp "EERSTE KWADRANT"
: Disp "X IS POSITIEF"
: Disp "Y IS POSITIEF"
: End
:

```

**Docenten Tip:** Deze opdracht heeft de voorkeur boven de voorgaande **If** –opdracht, omdat deze veel makkelijker te lezen is. Het 'blok' kan ook uit slechts één opdracht bestaan (of zelfs helemaal geen opdracht bevatten!). **Then** en **End** staan op hun eigen regels in het programma. Het 'blok' kan ook nog een andere **If-opdracht** bevatten. Elke **If Then** heeft een bijbehorende **End** nodig. Maar we bespreken geneste structuren pas later nog.

### Samengestelde voorwaarden

Samengestelde voorwaarden bestaan uit meer dan één relationele uitdrukking. De logische operatoren **and (en)**, **or (of)**, **xor (xof)** en **not( (niet)** zijn te vinden in het **Ô** menu LOGIC. Met deze operatoren kan je Samengestelde voorwaarden in elkaar zetten.

Voorbeelden:

- $X > 0$  **en**  $Y > 0$  is 'waar' wanneer X en Y beide positief zijn
- $X > 0$  **of**  $Y > 0$  is 'waar' wanneer X of Y positief is (of allebei)
- **niet**( $X > 0$  **en**  $Y > 0$ ) is 'waar' wanneer X of Y niet positief is  
Dit betekent hetzelfde als  $X \leq 0$  **of**  $Y \leq 0$
- $X > 0$  **xof**  $Y > 0$  is 'waar' wanneer of X, of Y positief is *maar niet beide*  
Dit betekent hetzelfde als...  $X > 0$  **of**  $Y > 0$  **en niet**( $X > 0$  **en**  $Y > 0$ )

```

NORM DRIJF AUTO REEEL RAD MN
TEST LOGISCH
1:en
2:of
3:xof
4:niet(

```



**xof** staat voor 'exclusief of' en is alleen 'waar' wanneer één van de twee delen 'waar' is, maar niet wanneer ze beide 'waar' zijn.

Je kan de relationele operatoren niet 'aan elkaar rijgen', bijvoorbeeld: Met  $2 < A < 3$  wordt bedoeld dat A tussen 2 en 3 ligt, maar dit moet gecodeerd worden als  **$2 < A$  en  $A < 3$** . De logische operatoren hebben een volgorde waarin ze worden uitgevoerd, net als de rekenbewerkingen +, -, \* en /.

**$A < 0$  of  $A < 5$  en  $A > 2$**  betekent dat A negatief kan zijn of tussen 2 en 5 ligt.

**en** wordt eerder uitgevoerd dan **of** (net zo als 'vermenigvuldigen' voor 'optellen' gaat).

**Docenten Tip:** Wat oefening om de uitwerking van logische voorwaarden beter te begrijpen zou hier goed van pas komen. Als je dit nog nooit eerder hebt gedaan kan het maken van waarheidstabellen een leeractiviteit zijn:

A	B	A en B	A of B	niet(A)
Waar	Waar	Waar	Waar	Onwaar
Waar	Onwaar	Onwaar	Waar	Onwaar
Onwaar	Waar	Onwaar	Waar	Waar
Onwaar	Onwaar	Onwaar	Onwaar	Waar

### Programmeren met If...Then...End opdrachten

Probeer het **IFTHEN** programma hier rechts uit.

*Opmerking: **Input** heeft geen variabele. Dit is een speciale functie van TI-Basic. Net als in Module 2 zal het scherm GRAPH verschijnen zodat je de cursor overal heen kunt verplaatsen en **e** kunt indrukken om de waardns van X en Y in te stellen.*

*'and' staat in het menu **LOGIC**.*

***Then** staat op zichzelf op een regel recht onder **If***

***End** is het einde van het blok 'waar' (alle opdrachten die worden uitgevoerd als aan de voorwaarde wordt voldaan). Maar het is niet het einde van het programma.*

### Maak het programma compleet

Een grafiek heeft meerdere gebieden, elk gebied heeft een naam zo heb je: kwadranten I, II, III, en IV en de positieve en negatieve x en y assen. We gaan nu een programma schrijven waarmee de gebruiker een punt kan selecteren op het scherm GRAPH en waar vervolgens het programma vertelt waar dit punt ligt in termen van deze gebieden .

We helpen je een eindje op weg met enkele **If**-opdrachten (in een Engels voorbeeld) en dan kan jij de rest doen:

```

Input          let op, geen variabele!
Disp X,Y
If X>0 and Y>0
Then
Disp "FIRST QUADRANT"
End
If X=0 and Y>0
Then

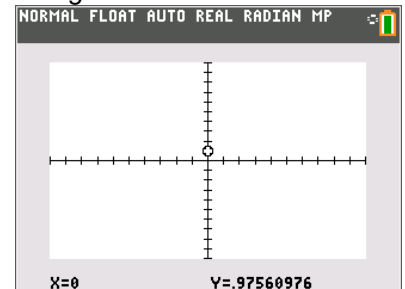
```

```

PROGRAM: IFTHEN
:Input
:If X>0 en Y>0
:Then
:Disp "EERSTE KWADRANT"
:Disp "X IS POSITIEF"
:Disp "Y IS POSITIEF"
:End
:Disp "EINDE"
:

```

*Uitvoeren van het programma brengt dit scherm in beeld ...*



*...en drukken op enter op die positie geeft dit als resultaat.*



# 10 Minuten programmeren

TI-84 PLUS SERIE

```

Disp "POSITIVE Y-AXIS"
End
If X<0 and Y>0
Then
Disp "SECOND QUADRANT"
End
.
.

```

Als het goed is zou je nu acht **If** structuren moeten hebben (voor de vier kwadranten en de vier halve assen).

### Voorbeeld antwoord (in een Engels voorbeeld):

```

Input _____ [Merk op, geen variable]
Disp X,Y

```

```

If X>0 and Y>0
Then
Disp "FIRST QUADRANT"
End

```

```

If X=0 and Y>0
Then
Disp "POSITIVE Y-AXIS"
End

```

```

If X<0 and Y>0
Then
Disp "SECOND QUADRANT"
End

```

```

If X<0 and Y=0
Then
Disp "NEGATIVE X-AXIS"
End

```

```

If X<0 and Y<0
Then
Disp "THIRD QUADRANT"
End

```

```

If X=0 and Y<0
Then
Disp "NEGATIVE Y-AXIS"
End

```

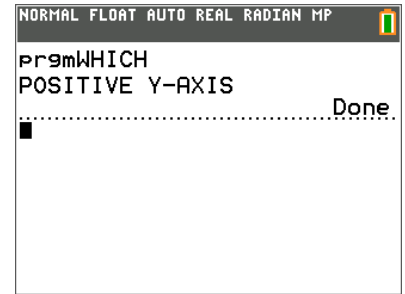
```

If X>0 and Y<0
Then

```

## MODULE 3: OEFENBLAD 2

### DOCENTENHANDLEIDING





## 10 Minuten programmeren

TI-84 PLUS SERIE

Disp "FOURTH QUADRANT"

End

MODULE 3: OEFENBLAD 2

DOCENTENHANDLEIDING