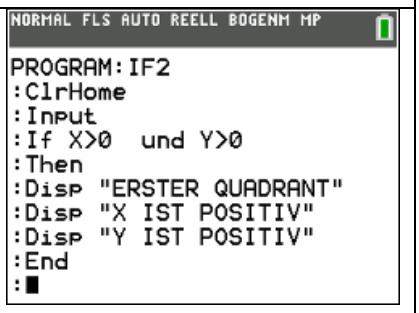
Lektion 3: Befehle mit Bedingungen	Übung 2: If…Then…End und verbundene Bedingungen
In der zweiten Übung von Lektion 3 geht es um den verbesserten **If** – Befehl und zusammengesetzte Bedingungen.	**Lernziele:** • Untersuchung der **If…Then…End** Struktur. • Zusammengesetzte Bedingungen mit den logischen Operatoren herstellen. • Schreiben eines Programmes mit der **If…Then…End** Struktur, dass Koordinaten einem Quadranten zuordnet.

Die If…Then…End Struktur

TI Basic verwendet in der Struktur **If…Then** das Schlüsselwort **End**, um damit das Ende einer Reihe von Anweisungen zu kennzeichnen, die ausgeführt werden, wenn die Bedingung wahr ist. Aufbau:

If <*Bedingung*>
Then
<Anweisungen die ausgeführt werden, wenn die Bedingung wahr ist>
End



Hinweis:
***If** wird immer gefolgt von einer Bedingung.*
Then** steht in einer* eigenen *Zeile unmittelbar unter **If.
Es folgen die Befehle, die ausgeführt werden, wenn die Bedingung wahr ist.
***End** gehört zu **Then** und grenzt damit die auszuführenden Befehle ab.*
***End** steht wieder in einer eigenen Zeile.*
***End ist* nicht *das Ende des Programmes, sondern nur das Ende der If…Then…End Struktur**.

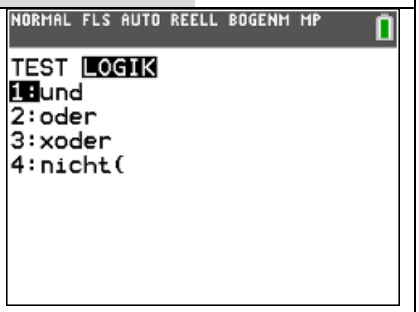
Hinweis: If…Then…End sollte gegenüber dem einfachen **If** bevorzugt angewendet werden, weil dann das Programm leichter zu lesen ist. Die Gruppe der Anweisungen zwischen **Then** und **End** kann auch nur aus einer einzelnen Anweisung bestehen oder auch einen weiteren **If** – Befehl enthalten, der dann natürlich wieder ein eigenes **End** benötigt. Doch dazu später mehr.

Zusammengesetzte Bedingungen

Zusammengesetzte Bedingungen enthalten mehr als einen relationalen Ausdruck, der mit den logischen Operatoren **und**, **oder**, **xoder** und **nicht(** aus dem Menü [TEST] LOGIC gebildet wird.

Beispiele:

- X>0 **und** Y>0 wahr, wenn X und Y beide positiv sind
- X>0 **oder** Y>0 wahr, wenn X oder Y oder beide positiv sind
- **nicht(**X>0 **und** Y>0) wahr, wenn X oder Y nicht positiv sind gleichbedeutend mit **X<=0 oder Y<=0**
- X>0 **xoder** Y>0 wahr, wenn X oder Y positiv sind, aber nicht, wenn beide es sind, gleichbedeutend mit **X>0 oder Y>0 und nicht(X>0 und Y>0)**



xoder ist das Exklusiv-Oder, das wahr ist, wenn einer der beiden Teile wahr ist, aber nicht beide zusammen.
Man kann relationale Operatoren nicht verketten wie in der Mathematik. **2<A<3** bedeutet „A liegt zwischen 2 und 3“ und muss im Programm geschrieben werden als **2<A und A<3.**
Logische Operatoren haben auch eine Rangfolge wie die Rechenarten:

A<0 oder A<5 und A>2 bedeutet, dass A negativ sein kann oder zwischen 2 und 5 liegt.
und wird vor **oder** ausgeführt, ähnlich wie die Multiplikation vor der Addition.

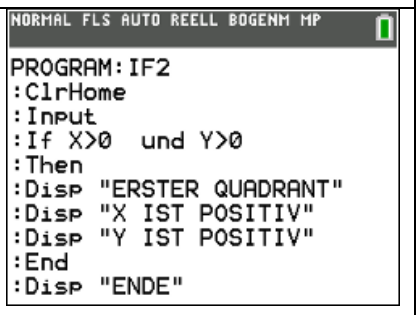
Hinweis: Man sollte sich mit den logischen Bedingungen vertraut machen. Die Wahrheitstabelle hilft dabei:

A	B	A und B	A oder B	A xoder B	nicht(A)
Wahr	Wahr	Wahr	Wahr	Falsch	Falsch
Wahr	Falsch	Falsch	Wahr	Wahr	Falsch
Falsch	Wahr	Falsch	Wahr	Wahr	Wahr
Falsch	Falsch	Falsch	Falsch	Falsch	Wahr

Programmieren mit der Befehlsgruppe If…Then…End

Das Programm **IF2** ist ein erstes Beispiel.

*Hinweis: **Input** ist ohne Variable, eine spezielle Möglichkeit des TI-Basic. In Lektion 2 wurde gezeigt, dass dann der GRAFIK-Bildschirm erscheint, auf dem man den Cursor frei bewegen kann. Drückt man [ENTER] so werden die angezeigten Werte für **X** und **Y** in das Programm übernommen.*
***und** findet man im Menü [TEST] LOGIC.*
Then** hat eine eigene Zeile direkt unter **If.
***End** schließt die Gruppe der bei* wahr *auszuführenden Befehle ab. Es ist nicht das Ende des Programmes! **If**, **Then** und **End** findet man im Menü [prgm] STRG.*



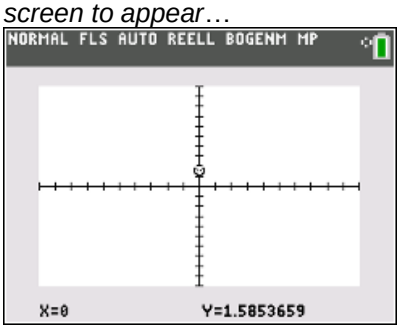
Vervollständigung des Programmes

Durch das Koordinatensystem wird die Zeichenebene in sogenannte Quadranten unterteilt: I, II, III und IV. Es soll nun ein Programm geschrieben werden, bei dem der Benutzer einen Punkt auf dem GRAPH-Schirm auswählt. Das Programm ermittelt dann den Quadranten und zeigt ihn und die Punktkoordinaten auf dem Bildschirm an.
Hier ist der Programmanfang:

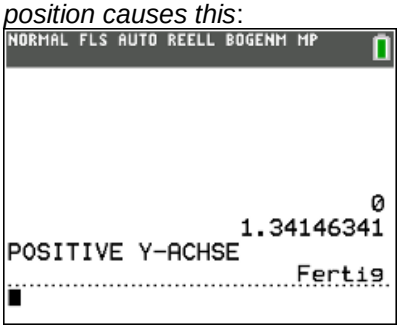
```
Input          (Achtung, keine Variable !)
ClrHome
Disp X,Y
If X>0 und Y>0
Then
Disp "ERSTER QUADRANT"
End
If X=0 und Y>0
Then
Disp "POSITIVE Y-ACHSE"
End
If X<0 und Y>0
Then
Disp "ZWEITER QUADRANT"
End
.
```

Das Programm enthält 8 **If**-Befehle, je 4 für die Quadranten und die Halbachsen.

Running the program cause this screen to appear…



…and pressing enter at that position causes this:



Beispielprogramm:

```
Input
ClrHome
Disp X,Y

If X>Ø und Y>Ø
Then
Disp "ERSTER QUADRANT"
End

If X=Ø und Y>Ø
Then
Disp "POSITIVE Y-ACHSE"
End

If X<Ø und Y>Ø
Then
Disp "ZWEITER QUADRANT"
End

If X<Ø und Y=Ø
Then
Disp "NEGATIVE X-ACHSE"
End

If X<Ø und Y<Ø
Then
Disp "DRITTER QUADRANT"
End

If X=Ø und Y<Ø
Then
Disp "NEGATIVE Y-ACHSE"
End

If X>Ø und Y<Ø
Then
Disp "VIERTER QUADRANT"
End

If X>Ø und Y=Ø
Then
Disp "POSITIVE X-ACHSE"
End
```