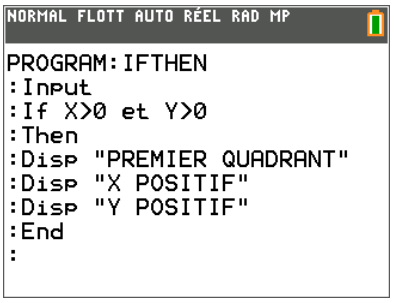
Dans cette seconde leçon de l'Unité 3 vous allez étudier une meilleure structure conditionnelle et des conditions complexes.

Objectifs :

- Examiner la structure **If…Then…End**.
- Utiliser des conditions complexes avec les opérateurs logiques.
- Écrire un programme utilisant la structure **If…Then…End** qui examine des régions du plan.

La structure If…Then…End



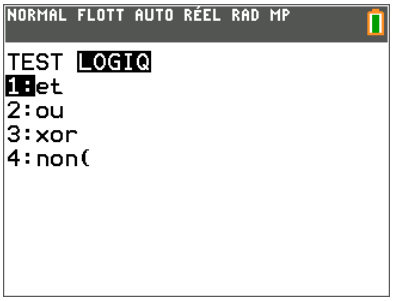
Le TI-Basic a une unique structure **If…Then** qui utilise le mot-clé **End** pour contrôler les instructions qui forment le bloc de code qui sera exécuté quand la condition est vraie. Ça ressemble à ceci :

If <*condition*>
Then
< *bloc_vrai : instructions exécutées quand la <condition> est vraie*>
End

Note :
***If** est suivi d'une <condition>.*
***Then** est immédiatement sous le **If,** sur une nouvelle ligne.*
Il y a une ou plusieurs instructions dans le <bloc_vrai>.
***End** indique la fin du bloc **Then** et les instructions dans les lignes après **End** seront traitées.*
***End n'est pas la fin du programme** ! C'est uniquement la fin de la structure **If…Then…End**.*

Indication : Cette structure est préférée au simple **If** car elle est plus lisible. Le 'bloc' peut être réduit à une seule instruction (ou même à pas d'instruction du tout !). **Then** et **End** apparaissent sur leurs propres lignes dans le programme. Le 'bloc' peut aussi inclure une autre structure conditionnelle **If**. Chaque **If Then** nécessite un **End** correspondant. Mais nous verrons les structures imbriquées plus tard.

Conditions complexes



Les conditions complexes impliquent plus d'une expression. Les opérateurs logiques **ou**, **et**, **xor** et **non(** se trouvent dans le menu LOGIQ de [tests]. Ces opérateurs vous permettent de construire des conditions complexes.

Exemples :

- X>0 **et** Y>0 est vraie quand X et Y sont tous les deux positifs
- X>0 **ou** Y>0 est vraie quand X est positif ou Y est positif (l'un ou l'autre ou les deux en même temps)
- **non(**X>0 **et** Y>0) est vraie quand X ou Y est non positif
  ça signifie la même chose que : **X<=0 ou Y<=0**
- X>0 **xor** Y>0 est vraie quand l'un ou l'autre est positif *mais pas les deux à la fois.*
  Ça signifie la même chose que : **X>0 ou Y>0 et non(X>0 et Y>0)**

xor représente le 'ou exclusif' qui est vrai quand l'un des deux est vrai mais pas les deux en même temps.
Vous ne pouvez pas fusionner deux opérateurs relationnels pour obtenir : 2<A<3 pour "A compris entre 2 et 3" vous devez le coder comme ceci : **2<A et A<3.** Les opérateurs logiques ont un ordre d'opérations comme les opérateurs arithmétiques +, - , *, et /.

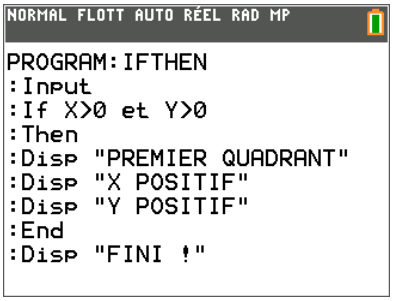
A<0 ou A<5 et A>2 signifie que A peut être négatif ou entre 2 et 5.
et est traité avant **ou** (de même que la 'multiplication avant l'addition').

Indication : Certaines pratiques d'évaluation des conditions logiques serait utile ici. Si vous ne l'avez jamais fait, les tables de vérité constituent une bonne activité d'apprentissage :

A	B	A et B	A ou B	non(A)
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

Programmation avec la structure If…Then…End

Essayez le programme avec la structure **IF…THEN…End** à droite.
*Notez : **Input** n'a pas de variable. Ceci est une fonctionnalité spéciale du TI-Basic. Rappel de l'Unité 2 : l'écran GRAPH s'affiche de sorte que vous pouvez déplacer le curseur et appuyer sur* [entrer] *pour affecter des valeurs à **X** et **Y**.*
*'**et**' est dans le menu* **LOGIQ** *de* [tests].
Then** est sur la ligne juste sous **If
***End** est à la fin du 'bloc_vrai' (ensemble des instructions exécutées quand la condition est vraie). Ce n'est pas la fin du programme.*



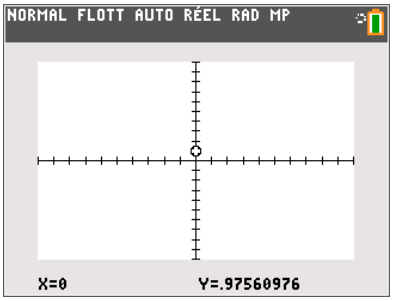
Complétez le programme

Le plan rapporté à un repère a plusieurs régions nommées : Quadrants I, II, III, et IV et les demi-axes *x* et *y* positif et négatif. Écrivons un programme qui permet à l'utilisateur de sélectionner un point de l'écran GRAPH, puis le programme indiquera dans quelle région se trouve le point.

Nous vous donnons le début avec quelques instructions **If**, vous pouvez ensuite compléter le programme :

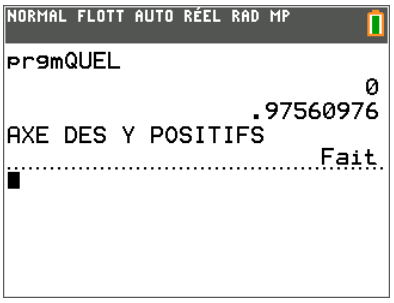
```
Input          notez, pas de variable !
Disp X,Y
If X>0 et Y>0
Then
Disp "PREMIER QUADRANT"
End
If X=0 et Y>0
Then
Disp " AXE DES Y POSITIFS"
End
If X<0 et Y>0
Then
Disp "SECOND QUADRANT"
End
```

L'exécution du programme entraîne l'affichage de cet écran…



…et la pression sur [entrer] *en cette position provoque ceci* :

Vous devrez avoir huit structures **If** (pour les quatre quadrants et les quatre demi-axes).



Exemple de réponse :

```
Input                    [Notez, pas de variable]
Disp X,Y

If X>Ø et Y>Ø
Then
Disp "PREMIER QUADRANT"
End

If X=Ø et Y>Ø
Then
Disp "AXE DES Y POSITIFS"
End

If X<Ø et Y>Ø
Then
Disp "SECOND QUADRANT"
End

If X<Ø et Y=Ø
Then
Disp " AXE DES X NEGATIFS"
End

If X<Ø et Y<Ø
Then
Disp "TROISIEME QUADRANT"
End

If X=Ø et Y<Ø
Then
Disp " AXE DES Y NEGATIFS"
End

If X>Ø et Y<Ø
Then
Disp "QUATRIEME QUADRANT"
End
```