Unit 3: Conditional Statements

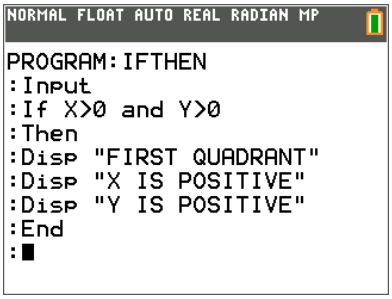
Skill Builder 2: If…Then…End and Compound Conditions

In this second lesson for Unit 3 you will learn about a much better conditional structure and compound conditions.

Objectives:

- Examine the **If…Then…End** structure.
- Make compound conditions with the logical operators.
- Write a program using the **If…Then…End** structure that examines the regions of the coordinate plane.

The If…Then…End structure



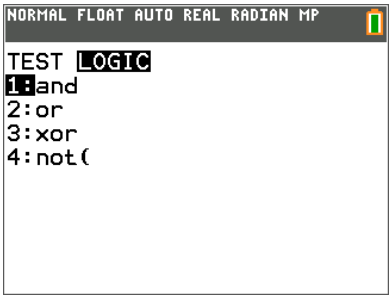
TI Basic has a unique **If…Then** structure that makes use of the **End** keyword to control the statements that form the block of code that will be processed when the condition is true. It looks like this:

If *<condition>*
Then
<true block: do these statements when the <condition> is true
End

Note:
***If** is followed by some <condition>.*
***Then** is immediately below **If,** set on a line by itself.*
There are one or more statements in the <true block>.
***End** indicates the end of the **Then** block and the statements below **End** will be processed.*
***End is not the end of the program**! It is the End of the **If…Then…End** structure.*

Teacher Tip: This statement is preferred over the previous **If** statement because it is easier to read. The 'block' can be a single statement (or even no statement at all!). **Then** and **End** appear on their own lines in the program. The 'block' can also include another **If** statement. Each **If Then** requires a corresponding **End**. But we discuss nested structures later on.

Compound Conditions



Compound conditions involve more than one relational expression. The logical operators **and**, **or**, **xor** and **not(** are found on the [TEST] LOGIC menu. These operators allow you to build compound conditions.

Examples:

- X>0 **and** Y>0 is true when both X and Y are positive
- X>0 **or** Y>0 is true when either X or Y is positive (or both)
- **not(**X>0 **and** Y>0) is true when either X or Y is not positive
  it means the same as **X<=0 or Y<=0**
- X>0 **xor** Y>0 is true when either X or Y is positive *but not both*
  it means the same as… **X>0 or Y>0 and not(X>0 and Y>0)**

xor stands for 'exclusive or' and is true when either part is true but not both parts.
You cannot 'string together' the relational operators: 2<A<3 is interpreted to mean "A is between 2 and 3" and must be coded as **2<A and A<3.** The logical operators have an order of operations just like the arithmetic operators +, - , *, and /.

A<0 or A<5 and A>2 means A can be negative or between 2 and 5.
and is processed before **or** (similar to 'multiplication before addition').

Teacher Tip: Some practice evaluating logical conditions would be helpful here. If you've never done it, truth tables make a good learning activity:

A	B	A and B	A or B	not(A)
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Programming with If…Then…End Statements

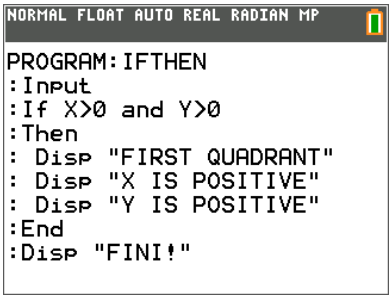
Try the **IFTHEN** program to the right.

*Note: **Input** has no variable. This is a special feature of TI-Basic. Recall from Unit 2 that the GRAPH screen will appear so that you can move the cursor anywhere and press [ENTER] to set values for **X** and **Y**.*

*'**and**' is on the [TEST] **LOGIC** menu.*

Then** is on a line by itself right below **If

***End** is the bottom of the 'true' block (the set of statements that are executed when the condition is true). It is not the end of the program.*



Complete the Program

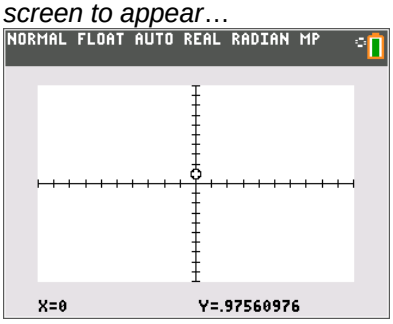
A graph has several named regions: Quadrants I, II, III, and IV and the positive and negative *x* and *y* axes. Let's write a program that allows the user to select a point on the GRAPH screen and then the program will tell where the point lies using those names.

We'll start you off with a few **If** statements and you can finish the rest:

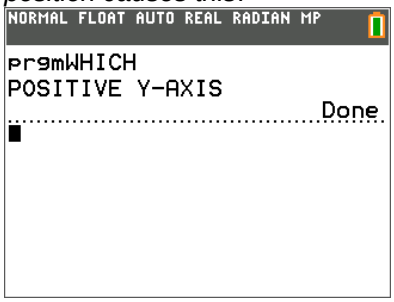
```
Input          notice, no variable!
Disp X,Y
If X>0 and Y>0
Then
Disp "FIRST QUADRANT"
End
If X=0 and Y>0
Then
Disp "POSITIVE Y-AXIS"
End
If X<0 and Y>0
Then
Disp "SECOND QUADRANT"
End
.
.
.
```

You should have eight **If** structures (for the four quadrants and the four half-axes).

Running the program cause this screen to appear…



…and pressing enter at that position causes this:



Sample Answer:

```
Input                    [Notice, no 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
Disp "FOURTH QUADRANT"
End
```