Boolean Logic in BASIC#

Reprinted from the A.C.E.C. BBS (614)-471-8559

Boolean logic is a very useful, but often unused, function of most BASIC languages. Boolean logic is a true/false comparison operation. Because it only returns a true or false value, it returns a 1 or a 0.

If you enter PRINT 1=1 the computer will print "1" on the screen. Why? Because 1=1 is a true statement. The value 1 does equal the value 1. A true statement is identified by a 1. If you enter PRINT 2=3 the computer will print a "0" on the screen. This is becase 2=3 is NOT a true statement. A false comparison will return a value of 0.

You can also make variable assignments with boolean logic. If you say A=1=1 then A=1 because 1=1 would return a 1. A short table might be helpful at this point:

Statement	**Result**	
A=1=1	A=1	
A=1=2	A=0	
A=2=1	A=0	
A=2=2	A=1	

This alone is useful in programming but we are allowed to use even more complex boolean logic. Right now we can say A=B=C. But what about A=B=C=D? Or A=B=C=D=E? How does that work? DOES it work? The answer is yes, it does work.

Long boolean operations such as this can be confusing unless you break it down. So that is what we will do. A=B=C=D is the same as A=B=(C=D). Lets try a few:

```
B=1, C=1, D=0: A=1=(1=0)
```

so A=1=(0) so A=1=0 so A=0

```
B=0, C=0, D=0: A=0=(0=0)
```

so A=0=(1) so A=0=1 so A=0

```
B=0, C=0, D=1: A=0=(0=1)
```

so A=0=(0) so A=0=0 so A=1

It is really simple. Just evaluate the comparison within the paranthesese first, just like a normal problem. In the last example, the comparison in the parans is (0=1). This is a false statement so we can put a 0 in the place of the parans. Still quite easy, right?

Now how about A=B=C=D=E? Getting a bit more complex here. What this amounts to is A=B=(C=(D=E)). Let's try a few:

```
B=1, C=0, D=1, E=0: A=1=(0=(1=0))
```

so A=1=(0=0) so A=1=(1) so A=1=1 so A=1

```
B=0, C=1, D=0, E=1: A=0=(1=(0=1))
```

so A=0=(1=0) so A=0=(0) so A=0=0 so A=1

As you can see, it is a little more complex but the idea is the same. Just do the parans first and then work out to the next set of parans. Again, just like normal math operations: Parans come first. So it is very simple:

Statement	**Equivalent To**	
A=B=C=D	A=B=(C=D)	
A=B=C=D=E	A=B=(C=(D=E))	
A=B=C=D=E=F	A=B=(C=(D=(E=F)))	
A=B=C=D=E=F=G	A=B=(C=(D=(E=(F=G))))	

I think you can see the pattern here.

It should be noted that while simple boolean logic like A=B=C is useful, complex boolean logic like A=B=C=D=E=F is really too complex to be useful in a real programming enviroment. In most cases there would be an easier, more memory efficient way to accomplish the same thing. But knowing how to use boolean logic will be very useful. Once you have completely mastered it you will find yourself using it often.

Craig Steiner