102-A01 Truth values vs. integer values

Work to do

In C, almost everything can be interpreted as a truth value based on the rule that 0 is false and everything else is true. More specifically, anything that evaluates into a numerical value can be interpreted as true or false. As you can guess, this leads C to interpret things that weren't meant as Boolean expressions as such without even issuing a warning. Learning how to program is also learning how what you type is actually interpreted by your programming language.

This exercise will help you strengthen and test your understanding of truth values in C and how non-boolean value are interpreted as such by the language (think of it, C unlike Java doesn't have any Boolean type does it?).

For each of the following expressions

- 'a' < 'b'
- 'a' > 'b'
- 0
- 1
- -1
- !0
- !30
- !'c'

You will do two things with these expressions;

(1) display their value as an integer with a

```
printf ("%d", INSERT EXPRESSION HERE);
```

(2) display whether the expression results in true or false by using a ternary conditional operator and a printf

```
printf ( "%s", ( INSERT EXPRESSION HERE)? "true" : "false" );
```

BEFORE you run your code, predict what will be the output of both printf statements for each expression. Then only run your code to confirm whether it is what you expected or not. This exercise is meant to raise questions so don't hesitate to voice them on the forums.

Example(s)

n/a

Hints

- n/a

Testing

Input	Output	
	Expected	Observed
n/a		