Boolean Shortcut Evaluations

Work to do

We want to write a little program to help us determine how the C language evaluates conditional expression composed of two terms (e.g. left to right, right to left, with shortcut evaluations…).

Here's what we **think** we know about C on this topic;

- When a conditional expression has multiple terms (e.g. (a < 3) && (c < 9)), these are evaluated from left to right
- && stops evaluating after figuring out that the leftmost term is false, the final result is false regardless of the second term
- || stops evaluating after figuring out that the leftmost term is true, the final result is true regardless of the second term

Write a little program which will establish these facts for us. Look at the Hints for help by thinking on the hints provided to you, you should be able to come up with an original way to use printf in a program in order to validate the above-mentioned hypothesis.

Example(s)

n/a

Hints

- The printf function actually returns a value which is the number of characters which were displayed successfully on the screen. E.g. printf("hi"); would return the value 2.
- Printf can be used as a term in a conditional expression. It will return an integer value which can then be considered true or false by C.
- Printf, regardless of what it returns, will always display something on the screen BEFORE to return a value. This is called a function with a "side effect".

Testing

n/a

Input	Results	
	Expected	Observed