Mini-Exercise #1

Part 1: Mod and Integer Division

Question: How can I use mod (%) and integer division (//) to figure out the nth digit of an integer?

In **part1.py,** complete the code so that after it gets a value for "number" and "n" from the user, it prints out the nth digit (from the right) of the given number.

Again, your solution should use mod (%) and integer division (//) to figure out this nth digit of the integer. Think about how you can do so.

As an example, if I have number = 123456, I want to be able to use mod and division in some way to fetch the 1st digit of this number (which is 6), or 2nd (which is 5), or 3rd (which is 4), etc. using some equation that involves the values of "number" and "n".

Part 2: XOR

Question : How can I use the logical operators we discussed to check for xor (exclusive or)? Remember, a xor b should be True if and only if just one of either a or b is True (the other has to be False) if both a and b are True, or both a and b are False, then a xor b will be False.

In **part2.py,** complete the code so that after it gets a value for "a" and "b" from the user, it prints out the value of **a xor b**, using one simple logical expression. You may assume that both of the values of a and b are going to be either True or False -- although the user **must enter them as integers 0 or 1**, the starter code converts 0 to False and 1 to True for you.

The code you add should look like this, in only one line: print(...your expression here...) and the value of your expression should be True iff a xor b is True, and False iff a xor b is False. By "expression" I mean a logical expression such as "a and b" or "a and (b or c)", or some sort of statement like that which only uses brackets (if needed), the variables "a" and "b", and logical operators.

As a more concrete example: If I wanted to write an expression for a different problem, let's say I wanted an expression that only equals to True if both a and b are False, then a possible answer would be the expression '(not a) and (not b)'. Why?