

# Programming Problems

For each of these problems (unless otherwise specified), write the needed code directly in the Python file, in the corresponding function definition.

All programming problems may also be checked by running 'Run File As Script' on the starter file, which calls the function `testAll()` to run test cases on all programs. Before submitting, make sure your code runs without raising an error message. Any syntax or runtime errors left in the code will result in a deduction on the assignment grade. You should do this for all future programming assignments as well.

## #1 - `numSign(x)` - 10pts

*Can attempt after Booleans, Conditionals, and Errors lecture*

Write a function **`numSign(x)`** that takes a number as a parameter and returns a string representing its sign. More specifically, the function should return "positive" if the number is positive, "negative" if it is negative, and "zero" otherwise.

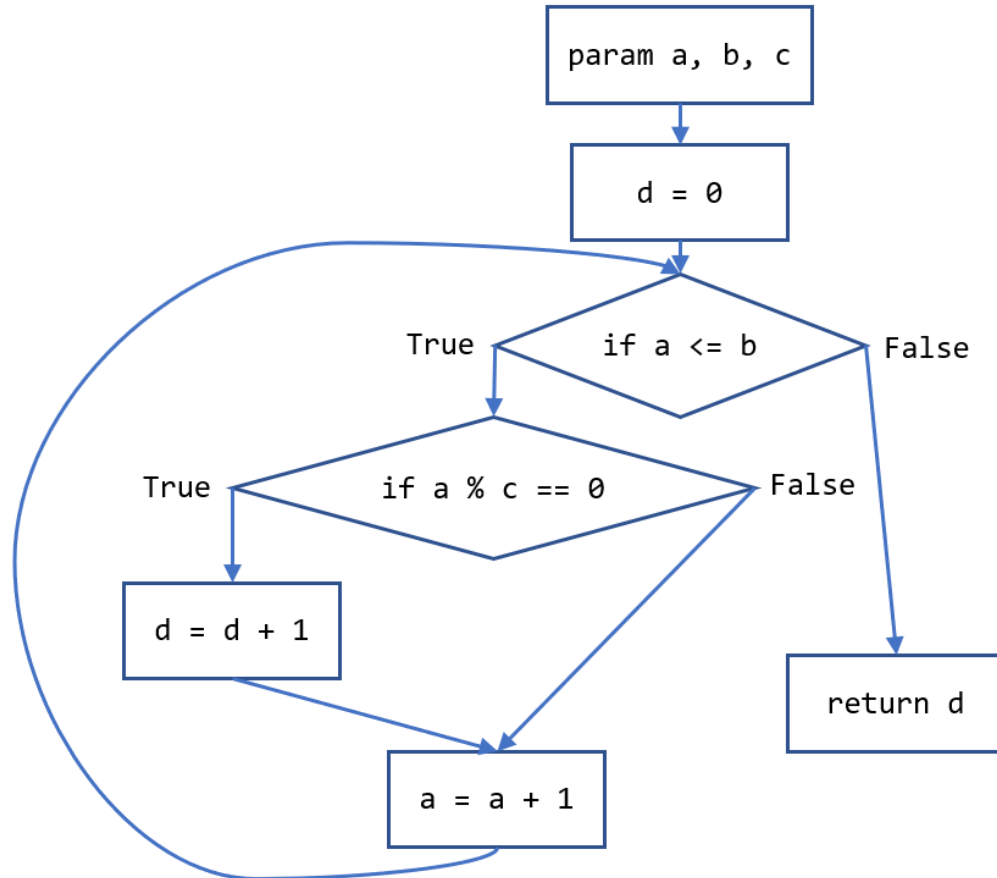
For example, `numSign(12)` should return "positive", `numSign(-0.5)` should return "negative", and `numSign(0)` should return "zero".

You are guaranteed that the function will only be called on ints and floats.

## #2 - Flow Chart to Program - 15pts

*Can attempt after While Loops lecture*

Given the control flow chart below, write a function `mysteryFunction(a, b, c)` that implements the control flow chart correctly.



### #3 - Interactive Program - 15pts

*Can attempt after Booleans, Conditionals, and Errors lecture*

In the function `interactiveProgram`, use the `input` function and conditionals to set up a short interactive program of your own design. This could be a very short choose-your-own-adventure story, or a BuzzFeed-style quiz, or whatever else you'd like! The only requirements are:

1. You must use the `input` function to collect information from the user at least three times.
2. The `interactiveProgram` function should take no parameters
3. You must use **conditionals** somewhere in your code. There should be at least two `if` statements and at least one `elif` or `else` statement.
4. All the code for your interactive program must be in the `interactiveProgram` function (or helper functions that `interactiveProgram` calls).

**Note:** we won't officially cover `input` until the While Loops lecture