15-110 Check2 - Written Portion

Name:

AndrewID:

Complete the following problems in the fillable PDF, or print out the PDF, write your answers by hand, and scan the results. Also complete the programming problems in the starter file check2.py from the course website.

When you are finished, upload your check2.pdf to **Check2 - Written** on Gradescope, and upload your check2.py file to **Check2 - Programming** on Gradescope. Make sure to check the autograder feedback after you submit!

Written Problems

#1 - Evaluating Boolean Expressions - 10pts

Can attempt after Booleans, Conditionals, and Errors lecture

For each of the following Boolean expressions, determine whether it evaluates to `True`, `False`, or an error. Select just one answer - the **best** answer.

`("foo" == "foo") and (4 > 5)`

- □ `True`
- □ `False`
- □ Error

`(10 < 0) or (0 == 1/0)`

- □ `True`
- □ `False`
- □ Error

`not (False and True)`

- □ `True`
- □ `False`
- □ Error

`(2 >= 5) and (4 + "a" == "4a")`

- □ `True`
- □ `False`
- □ Error

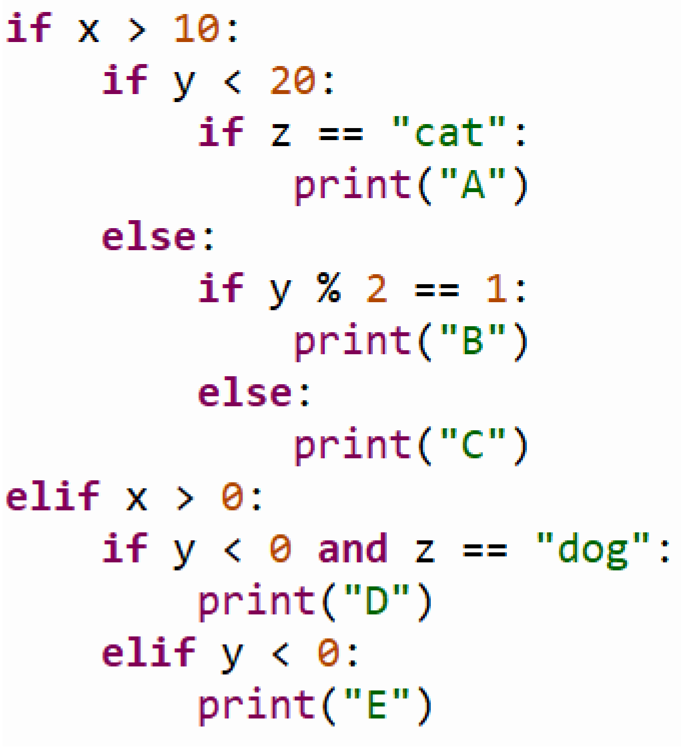
`(1/0 == 0) or (0 < 1)`

- □ `True`
- □ `False`
- □ Error

#2 - Code Tracing Conditionals - 15pts

Can attempt after Booleans, Conditionals, and Errors lecture

Given the following block of code, choose specific values for x, y, and z that would lead to the code printing A, B, C, D, or E. Note that there are multiple correct answers for some of these. Fill out your answers in the table below.



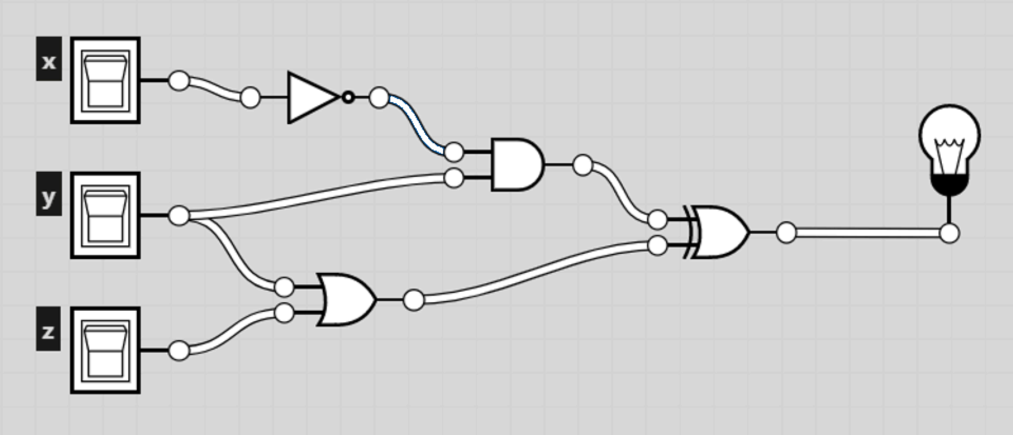
```
if x > 10:
    if y < 20:
        if z == "cat":
            print("A")
    else:
        if y % 2 == 1:
            print("B")
        else:
            print("C")
elif x > 0:
    if y < 0 and z == "dog":
        print("D")
    elif y < 0:
        print("E")
```

Printed Result	x value	y value	z value
A			
B			
C			
D			
E			

#3 - Circuit to Truth Table and Expression - 20pts

Can attempt after Circuits and Gates lecture

Fill out the truth table below to show the resulting outputs given the input combinations for the circuit shown here.



x value	y value	z value	output value
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Write a Boolean expression which is equivalent to the circuit above in the box below.
Note: it's fine to use normal Boolean terms (and/or/not/xor) instead of circuit operators.

#4 - While Loop Control Variable Values - 15pts

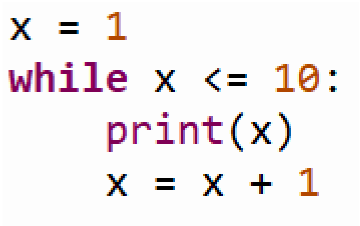
Can attempt after While Loops lecture

Each of the following problem prompts could be implemented using a while loop. Identify the start value, continuing condition, and update action for the loop control variable you would use in that while loop. Assume that the loop control variable will be printed at the beginning of the loop, and no conditional will be used. We've given an example of what this looks like in the first line

Ex) Print the numbers from 1 to 10, inclusive of both.
A) Print all even numbers between 2 and 20, including 2 but not including 20.
B) Print the numbers from 10 to 1, inclusive of both.
C) Print the numbers 3, 9, 15, 21.

Prompt	**Start Value**	**Continuing Condition**	**Update Action**
Ex	1	x <= 10	x = x + 1
A			
B			
C			

Hint: The code for Ex) Print the numbers from 1 to 10, inclusive of both looks like this:



```
x = 1
while x <= 10:
    print(x)
    x = x + 1
```

You only need to fill in the table and do not need to write the actual code, but this example may help you to understand where the print occurs.

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 current 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 a syntax error; any syntax errors remaining in the code will lead to a deduction in your overall grade. Runtime and assertion errors will also affect the grades of individual problems. You should check your code this way 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"`.

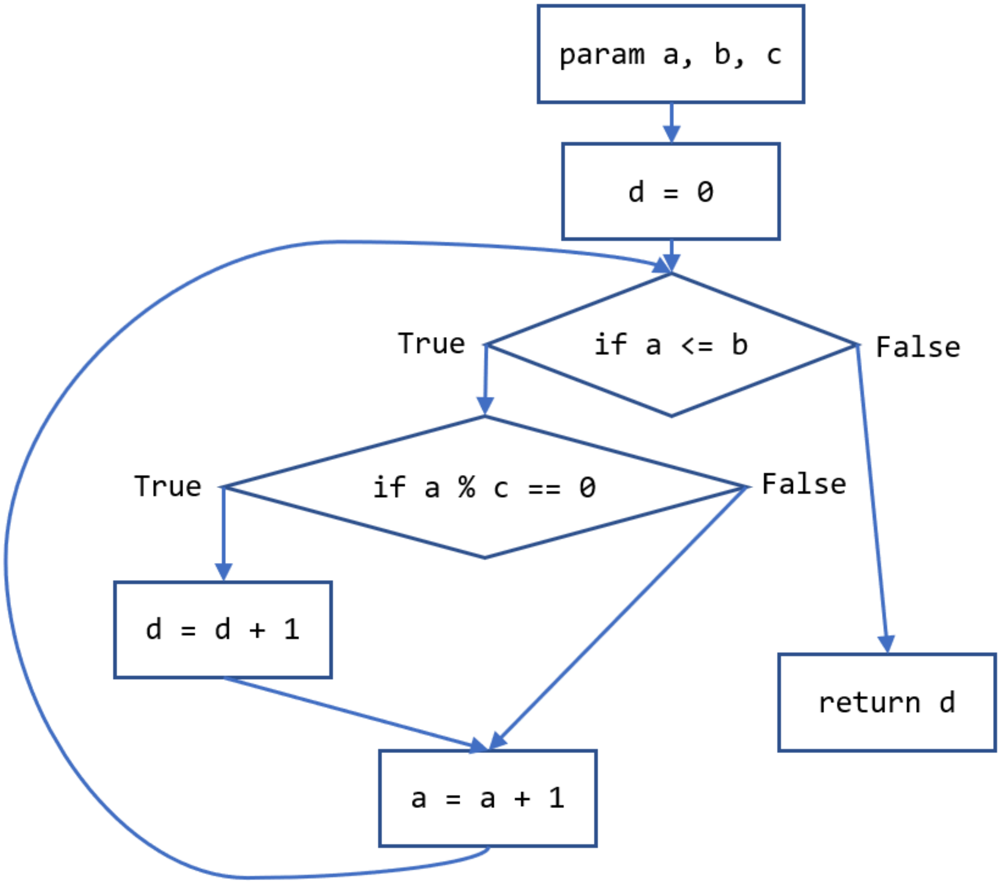
You are guaranteed that the function will only be called on ints and floats.

#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.

Note: make sure to run the test cases to see if you've implemented the code correctly! If the test case fails or your code loops infinitely, check whether your indentation matches what is expected here.



#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 functions that `interactiveProgram` calls). Do not put calls to `input` at the top level (i.e. outside of a function) - **this will break the autograder.**