Homework 3 - Iterations

CS 1301 - Intro to Computing - Spring 2020

Important

- Due Date: **Tuesday, January 28th, 11:59 PM.**
- This is an individual assignment. High-level collaboration is encouraged, **but your submission must be uniquely yours.**
- Resources:
  - TA Helpdesk
  - Email TA's or use class Piazza
  - How to Think Like a Computer Scientist
  - CS 1301 YouTube Channel
- Comment out or delete all function calls. Only import statements, global variables, and comments are okay to be outside of your functions.
- **Read the entire document before starting this assignment.**

Responsible Coding: Some functions in this assignment will have a 'Responsible Coding' badge next to them. These functions are written to help you think about how you can use your skills as a programmer to analyze problems related to *sustainability*, such as ethics, health, security, and the environment.

Hidden Test Cases: In an effort to encourage debugging and writing robust code, we will be including hidden test cases on Gradescope for some functions. You will not be able to see the input or output to these cases. Below is an example output from a failed hidden test case:

```
Test failed: False is not true
```


Written by Anthony Zheng (azheng47@gatech.edu) & Arvin Poddar (apoddar32@gatech.edu)

String Product

Function Name: stringMultiply
Parameters: sentence (`str`), num (`int`)
Returns: product (`int`)
Description: You're texting your friend when you notice that they replace many letters with numbers. Out of curiosity, you want to find the product of the numbers. Write a function that takes in a string **sentence** and an int **num**, and find the product of only the first **num** numbers in the sentence. If **num** is 0, return 0. If **num** > 0 but there aren't any numbers in the string, return 1. If **num** is greater than the amount of numbers in the sentence, return the product of all the numbers. (Hint: the **.isdigit()** method may be helpful.)

```
>>> sentence = "h311o n1c3 2 m33t u"
>>> num = 6
>>> print(stringMultiply(sentence, num))
18

>>> sentence = "4nthony 4nd arv1n were h3r3"
>>> num = 10
>>> print(stringMultiply(sentence, num))
144
```

Scrabble

Function Name: longestWord
Parameters: sentence (`str`)
Returns: a string (`str`)
Description: After texting your friend, you invite them over for a game of Scrabble. You don't really know the rules, so you try to find the longest word you can make. Write a function that takes in a sentence and returns the **longest word** in it. All words will be surrounded by one space on each side. If multiple words have the same longest length, return the one that appears last in the sentence. You may **not use .split()** or any similar methods.

```
>>> sentence = " run that back turbo "
>>> print(longestWord(sentence))
turbo

>>> sentence = " jetson made anotha one "
>>> print(longestWord(sentence))
anotha
```

Draw Triangle

Function Name: drawTriangle
Parameters: height (`int`)
Returns: `None`
Description: After destroying your friend in Scrabble, you decide to unwind and watch some Bob Ross together. Unfortunately, you can't paint, so you decide to draw triangles instead! Write a function that takes in a **height** and prints a right triangle of that height to the Shell. Each line should be its line number printed that many times. Each number should be separated by spaces, but there should no extra spaces at the end of each line. See examples below:

```
>>> drawTriangle(3)
1
2 2
3 3 3

>>> drawTriangle(5)
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

Population Growth Responsible Coding

Function Name: populationGrowth
Parameters: starting population (`int`), growth rate (`float`), expected population (`int`)
Returns: years (`int`)
Description: City planners often have to make major decisions based on the anticipated growth of a city's population. These decisions can relate to anything from public policy to safety regulations. Given a city's **starting population**, an annual **growth rate** (as a %), and an **expected population**, help the city planners find out how many years it will take before the city exceeds that expected population. Write a function that returns this number of years. You must solve this using loops, and not any population growth formulas.

```
>>> start = 135434
>>> rate = 4
>>> end = 246776
>>> print(populationGrowth(start, rate, end))
16
```

```
>>> start = 345936
>>> rate = 1.5
>>> end = 584790
```

```
>>> print(populationGrowth(start, rate, end))
36
```

MARTA Map Responsible Coding

Function Name: marta
Parameters: path (`str`), start (`str`), end (`str`)
Returns: distance (`int`)
Description: As part of your efforts to use more public transport to reduce your carbon footprint, you decide to take a MARTA to get around. You have a MARTA path map: a string of letters, each representing a stop, separated by dashes between each stop. Each dash represents a mile. You also know what stop to get on the train, and which stop to get off at. Write a function that returns the distance between these stops. If the start and end stop are the same, return 0. You may assume that both stops exist in the path, and the ending stop will always come after the starting stop.

```
>>> path = "a---b-c-----d"
>>> start = "b"
>>> stop = "c"
>>> print(marta(path, start, stop))
1

>>> path = "a-b-c---d--e--f"
>>> start = "a"
>>> stop = "f"
>>> print(marta(path, start, stop))
9
```

Grading Rubric

Function	Points
stringMultiply()	20
longestWord()	25
drawTriangle()	20
populationGrowth()	15
marta()	20
Total	**100**

Provided

The `HW03.py` skeleton file has been provided to you. This is the file you will edit and implement. All instructions for what the functions should do are in this skeleton and this document.

Submission Process

For this homework, we will be using Gradescope for submissions and automatic grading. When you submit your `HW03.py` file to the appropriate assignment on Gradescope, the autograder will run automatically. The grade you see on Gradescope will be the grade you get, unless your grading TA sees signs of you trying to defeat the system in your code. You can resubmit this assignment an unlimited number of times until the deadline; just click the "Resubmit" button at the lower right-hand corner of Gradescope. You do not need to submit your `HW03.py` on Canvas.