Homework 3 - Loops and Iteration

CS 1301 - Intro to Computing - Fall 2021

Important

- Due Date: **Tuesday, September 14th, 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.**

The goal of this assignment is to give practice on simple string indexing problems and iteration using for or while loops. The homework will consist of 5 functions for you to implement. You have been given `HW03.py` skeleton file to fill out. Please read this PDF thoroughly as you will find more detailed information to complete your assignment.

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 Kathleen Butler (kbutler36@gatech.edu), Joshua Tabb (jtabb6@gatech.edu), and Michael Sullivan (msullivan67@gatech.edu)

Username Entry

Function Name: userReplace()
Parameters: startingString (`str`), username (`str`)
Returns: replacedString (`str`)
Description: You are building a new website, and you want to customize it to your users. Given a string, replace any '$' characters in the string with the given username. Then, return the fixed string.

```
>>> userReplace('Hello, $, welcome to my website.', 'CoolCat13')
"Hello, CoolCat13, welcome to my website."
```

```
>>> userReplace("You have visited $'s page a lot recently.", "TheRealShawnMendes")
"You have visited TheRealShawnMendes's page a lot recently."
```

Special Characters

Function Name: specialChar()
Parameters: characterString (string)
Returns: sumOfIndices (int)
Description: You want to know the sum of the indices of the special chracters in a given string Given a string that includes letters, numbers, and special charachters, return the sum of the indices that have special charatcers.

Note: The characters that count as special chararcters are "!@#$%^&*."

```
>>> specialChar("ab*na&cao21$")
18
```

```
>>> specialChar("uaja%$$@#23ac")
30
```

Football Game

Function Name: footballGame()
Parameters: score1 (`str`), score2 (`str`), team1 (`str`), team2 (`str`)
Returns: winningTeam (`str`)
Description: You missed the football game last weekend, but you need to figure out who won the match. Given 2 strings that contain points scored during 4 different quarters of a football game, separated by '_ ', add up the points to total up each team's score at the end of the game, and return the winner of the game. In the event of a tie, return the string `"It's a tie!"` . **Assume the scores will always be a single digit.**

Note: score1 corresponds to team1, and score2 corresponds to team2.

```
>>> footballGame('7_7_1_7', '0_7_7_7', 'NIU', 'GT')
"NIU"
```

```
>>> footballGame('0_7_3_0', '0_0_0_3', 'Georgia', 'Clemson')
"Georgia"
```

Buy The Dip

Function Name: buyTheDip()
Parameters: currentPrice (`float`), finalPrice (`float`), growth (`float`)
Returns: days (`int`)
Description: You are trying to build your investment portfolio, so you have decided to purchase a bunch of stocks! You've gotten some intel that the value of Snapchat is going to go down before the end of the year, so you're going to purchase the stock during the dip. Write a function that takes in the current price of the stock, the final price that you want to buy at, and the daily growth rate of the stock **given as a percent**. Return how many days it will take for Snapchat to reach or exceed the final price.

Note: The growth rate will be negative, but the current price and final price will be positive.

```
>>> buyTheDip(75.83, 10, -5.85)
34
```

```
>>> buyTheDip(5025, 450, -2.3)
104
```

Question Maker

Function Name: questionMaker()
Parameters: questions (`str`), subQuestions (`str`)
Returns: combinedQuestions (`str`)
Description: You've been chosen to write the questions for the next homework. Each question has multiple sub questions. For example, question 1 might have 3 sub questions labeled 1a, 1b, 1c. Write a function that takes in a string of numbers representing the number of questions to write and a string of letters representing the sub questions for each question. Return a string with all the questions in order with the correspoding sub questions attached to each question.

Note: If there are no sub questions, the output should be an empty string.
Note: Each question will have the same number of sub questions.

```
>>> questionMaker("123", "abc")
"1a1b1c2a2b2c3a3b3c"
```

```
>>> questionMaker("1234", "ab")
"1a1b2a2b3a3b4a4b"
```

Grading Rubric

Function	Points
userReplace()	20
specialChar()	20
footballGame()	20
buyTheDip()	20
questionMaker()	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 re-submit 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.