Homework 3 - Loops and Iteration

CS 1301 - Intro to Computing - Fall 2022

Important

- Due Date: **Tuesday, September 13th, 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 Parul Srivastava (srivastava61@gatech.edu) & Josh Tabb (jtabb6@gatech.edu)

Product

Function Name: product()
Parameters: nums (`str`)
Returns: product (`int`)
Description: Given a string containing only the numbers 0-9, calculate the product of all the numbers in the string and return this value

```
>>> product("14123")
24
```

```
>>> product("586930")
0
```

Recipe Count

Function Name: recipeCount()
Parameters: recipe (`str`)
Returns: totalTeaspoons (`int`)
Description: Given a string containing the ingredients from a recipe, with each ingredient measured in a whole amount of teaspoons (which will only be one digit), write a function that returns the total amount of teaspoons needed for this recipe. If you are given an empty string, return 0.

Note: The only time digits will appear in the recipe string is for the amount of teaspoons.

```
>>> recipeCount("1 teaspoon of brocolli, 3 teaspoons of lettuce")
4
```

```
>>> recipeCount("4 teaspoons of kashmiri chili powder, 4 teaspoons of gochujang")
8
```

De-Instagram-inator

Function Name: instagraminator()
Parameters: usernames (`str`)
Returns: decodedUsernames (`str`)
Description: You are going through your list of sign-ups for your club, and realized that many people signed up using their instagram username, including symbols such as `'@'`, `'_'`, and `'$'`. In order to find the real names of these individuals, you need to create a function which takes in a string containing instagram usernames, and return a new string with these 3 symbols removed.

```
>>> instagraminator("@josh_The_TA, @$Queen$Parul$, @Party_$_Paige")
"joshTheTA, QueenParul, PartyPaige"
```

```
>>> instagraminator("@$$$Cash$$$Craig, @A_Lot_Of_Aryan_")
"CashCraig, ALotOfAryan"
```

Student Loan Forgiveness

Function Name: studentLoans()
Parameters: loanAmount (`int`)
Returns: forgivenessCount (`int`)
Description: Happy Loan Forgivness everyone! As a college student, you want to figure out how many times you would have to be forgiven of your student loan debt before you no longer are in debt (i.e., your debt is less than or equal to 0). Assuming each loan forgiveness is taking away $10,000 from the total amount of student loans, write a function which takes in an integer amount of total student loan debt you have, and returns a count of how many times you will have to be forgiven in order for you to be debt free.

Hint: You might find while loops to be more useful than for loops for this problem.

```
>>> studentLoans(42000)
5
```

```
>>> studentLoans(200001)
21
```

Team Playoffs

Function Name: playoffs()
Parameters: team1name (`str`), team2name (`str`), scoreCount (`str`)
Returns: winningTeam (`str`)
Description: You are watching the sportsball playoffs, and want to write a function which will calculate the winner of the game. Given the names of the two teams (team 1 and team 2), as well as a string made up of 1's and 2's, where the numbers represent each time team 1 and team 2 scores a point respectively, write a function which will sum up the amount of points scored by each team, and returns a string in the form of:

```
"{team1name} has won the game!"
```

if team 1 scores more points, or

```
"{team2name} has won the game!"
```

If a tie occurs, your function should return the string

```
"It was a tie :("
```

```
>>> playoffs("Parul", "Josh", "1111112221")
"Parul has won the game!"
```

```
>>> playoffs("Liquid", "OG", "1222112221")
"OG has won the game!"
```

Grading Rubric

Function	Points
product()	20
recipeCount()	20
instagraminator()	20
studentLoans()	20
playoffs()	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.