Homework 4 - Strings, Indexing, & Lists

CS 1301 - Intro to Computing - Fall 2020

Important

- Due Date: **Tuesday, September 22nd, 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 homework is for you to enhance your understanding of strings, indexing, and lists. The homework will consist of 5 functions for you to implement. You have been given `HW04.py` skeleton file to fill out. However, below you will find more detailed information to complete your assignment. Read it thoroughly before you begin.

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 Alexa O'Reilly (aoreilly@gatech.edu) & Brian Chuo (bchuo3@gatech.edu) & Craig Nyazema (cnyazema3@gatech.edu)

Find the Maximum

Function Name: findMax()
Parameters: list of numbers (`list`), start index (`int`), stop index (`int`)
Returns: highest number (`int`)
Description: Write a function that finds (and returns) to highest number in the list in between the two indices given. The start index and stop index are both inclusive and will always be in range. Assume there will be no repeats in the list.

Note: You are not allowed to use the `max()` function for this problem

```
>>> findMax([1, 8, 3, 2, -4], 2, 4)
3
```

```
>>> findMax([3, 0, 7, 3, 2], 0, 4)
7
```

Fruit Pie

Function Name: fruitPie()
Parameters: fruits (`list`), minimum quantity (`int`)
Returns: fruits for pie (`list`)
Description: You are tired of eating take-out and decide to get some fruit to make some pies. However, when you get to the store, you see a bunch of different types of fruit in different quantities and need to figure out which fruit has enough supply to make a pie! Write a function that takes in a list of fruit and their quantities and figures out which types of fruit have quantities greater than or equal to the minimum quantity passed in. Return a list of these fruits. If the list is empty, return an empty list. The quantities will never be negative.

Note: The list will be formatted as follows:

```
[fruit1, quantity1, fruit2, quantity2, . . .]
```

You can assume that every fruit will be followed by its quantity.

```
>>> fruitPie(["apple", 2, "cherry", 10, "watermelon", 5], 4)
["cherry", "watermelon"]
```

```
>>> fruitPie(["peach", 55, "orange", 32, "pineapple", 2], 50)
["peach"]
```

Replace Word

Function Name: replaceWord()
Parameters: initial sentence (`str`), replacement word (`str`)
Returns: corrected sentence (`str`)
Description: Write a function that takes in a sentence and a replacement word. Every word within the sentence with a length of 5 or greater should be replaced with the replacement word provided.

Hint: The string method `.split()` might be useful
Note: You are not allowed to use the `.replace()` method

```
>>> replaceWord("I used to rule the world","seas")
'I used to rule the seas'
```

```
>>> replaceWord("I miss the old Kanye","coding")
'I miss the old coding'
```

Highest Sum

Function Name: highestSum()
Parameters: list of strings (`list`)
Returns: index of string with the highest sum (`int`)
Description: Write a function that takes in a list of strings and computes the sum of the numbers in each string. If a string contains no numbers, the sum of that string is 0. Return the first index of the string with the highest sum.

Hint: You may find the `.isdigit()` string method useful
Note: You may use the `max()` function and the `.index()` method for this problem

```
>>> myList = ["3lf", "bg_73e", "001!0", "gg9./"]
>>> highestSum(myList)
1
```

```
>>> myList = ["py1h0n", "1s", "v3ry", "fun!!11!!!111"]
>>> highestSum(myList)
3
```

Sublist

Function Name: sublist()
Parameters: alist (`list`), blist (`list`)
Returns: True or False (`bool`)
Description: Write a function that checks to see if a **blist** is a sublist of **alist**. **blist** is a sublist of **alist** if all of its elements are in **alist** in the same consecutive order. Return whether **blist** is a sublist of **alist**. If two lists are equal, they are sublists of one another. Additionally, if a list is empty, it is always a sublist of the other.

```
>>> alist = [6, 2, 3, 4, 5]
>>> blist = [6, 3]
>>> sublist(alist, blist)
False
```

```
>>> alist = ['a', 'b', 'd', 'e', 't']
>>> blist = ['b', 'd', 'e']
>>> sublist(alist, blist)
True
```

```
>>> alist = ["The", "Houston", "Astros", "are", "cheaters"]
>>> blist = ["The", "Houston", "Astros", "are", "cheaters"]
>>> sublist(alist,blist)
True
```

Grading Rubric

Function	Points
findMax()	15
fruitPie()	20
replaceWord()	20
highestSum()	20
sublist()	25
Total	**100**

Provided

The `HW04.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 `HW04.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 "Re-submit" button at the lower right-hand corner of Gradescope. You do not need to submit your `HW04.py` on Canvas.