uOttawa

ITI 1120 E - Assignment 3

The purpose of this assignment is to practice concepts that you have seen in class so far. This assignment does not require the use of recursion and global variables. Assignment 3 is 5% of the final mark. The deadline for all regular and SASS students is **March 4, 2023, at 11:59 pm**. No late assignments will be accepted.

1. This work should be done **individually.**
2. Submit a zip file in BrightSpace **a3_xxxx.zip** where the xxxx corresponds to your student id. The zip file should contain the files a3_xxxx.py, and a3_xxxx.txt, and declaration-YOUR-FULL-NAME.txt
3. Put all the required functions in a file a3_xxxx.py and the results of your tests are a file a3_xxxx.txt.
4. Your grade will partially be determined by automatic (unit) tests that will test your functions.
5. All the specified requirements below are mandatory (including **function names**, and the behavior implied by examples). Any requirement that is specified and not met will result in a deduction of points. A function that does not have the required name will get zero points.
6. You MUST add comments for pre-conditions and assumptions in addition to the **doctoring** for each function (with a short description, the pre-conditions, and the type contract).
7. **If your file a3_xxxx.py gives a syntax error, the mark will be zero.**
8. Note that we will be using a plagiarism detection tool. In case two assignment solutions are identical or very similar, the mark will be zero for both students.

About declaration-YOUR-FULL-NAME.txt file:

It needs to be a plain text file and it must contain references to any code you used that you did not write yourself, including any code you got from a friend, internet, social media/forums or any other source or person. The only exclusion from that rule is the code that we did in class or as part of the lab work and code from official python.org documentation. So here is what needs to be written in that file. In every question where you used code from somebody else, you must write:

1. question number

2. copy-pasted parts of the code that were written by somebody else. That includes the code you found/were given that you then slightly modified. No copying from a friend!

3. whose code it is: name of a person or place on the internet/book where you found it. While you may not get points for that part of the question, you will not be in a position of being accused of plagiarism.

You have an option not to include this file, however, not including declaration-YOUR-FULL-NAME.txt will be taken as you declare that all the code in the assignment was written by you. Any student caught in plagiarism will receive zero for the whole assignment and will be reported to the dean. Finally, showing/giving any part of your assignment code to a friend also constitute plagiarism and the same penalties will apply. *You are not allowed to consult any tutoring website such as Chegg, Coursehero, (including Stack Overflow and discord), etc.*

Section1: A Library of Functions

Question 1. (6 points) Implement a Python function called **triples** that takes a string and returns True if there is at least one sequence of 3 consecutive characters of the same value, and False otherwise. Make sure the function is efficient, meaning it returns the boolean value as soon as the result is known (as soon as 3 identical characters were found).

Question 2. (6 points) Implement a Python function called **countMe** that takes a string of characters and returns a new string that contains the characters one time each, in the same order, with their number of repetitions. For example, if the string is 'bbcggggeeeee' the new string is 'b2c1g4e5'.

Question 3. (6 points). Write a function called **sum_odd_divisors** that takes a integer n as input. If n is zero, sum_odd_divisors returns None. Else, sum_odd_divisors returns the sum of all the postive odd divisors of n.

Question 4: (7 points) You want to send a note to your friend in class, and because you want to be respectful in class, you don't want to whip out your phone and send a text or an email (or any other digital communication). Instead, you choose to go old school and write it out on a piece of paper and pass it along to your friend. The problem is, you don't want anyone else to read the note as they pass it along. Luckily, your friend and you have come up with an encryption system so that nobody else can understand your message. Here's how it works: you write out your message backwards (so, Hello, world becomes dlrow ,olleH). But you don't stop there, because that's too easy to crack -anyone can gure that out! Now that you've written in backwards, Then you start on either side of the string and bring the characters together. So the first and the last characters become the first and the second character in the encrypted string, and the second and the second-last characters become the third and the fourth characters in the string, and so on. Note: all punctuation, special characters, spaces, etc are all treated the same) and 0123456789 becomes 9081726354. **Write a function called encrypt, that has one parameter s where s is a string and encrypt returns a string which is the encrypted version of s. Hint:** Think of how slicing and reversing of strings can help with this problem.

Section2: Testing

Test all the functions in the Python interpreter and add the results in the file a3_xxxx.txt

Copy and paste is the file a3_xxxx.txt when you test each function in the interpreter. Your file a3_xxxx.txt should look something like this (test with other values too):

```
#test Q1
triples("abc")
False
triples("abbbcdeeggggg")
True
triples("abc2eee")
True


#test Q2
countMe("a")
'a1'
countMe("aabbbccccx")
'a2b3c4x1'
countMe("aaa1111")
'a314'


#test Q3
sum_odd_divisors(-9)
13
sum_odd_divisors(1)
1
sum_odd_divisors(2)
1
sum_odd_divisors(3)
4
sum_odd_divisors(7)
8
sum_odd_divisors(-2001)
2880


#test Q5
encrypt("Hello, world")
'dHlerlolwo ,'
encrypt("1234")
'4132'
encrypt("12345")
'51423'
encrypt("1")
'1'
encrypt("123")
'312'
encrypt("12")
'21'
encrypt(",'4'r")
"r,''4"
```