uOttawa

ITI1120 G- Introduction to Computing 1 – Winter 2020
Assignment 2

Section A: Important Instructions

1. Read and follow the instructions below carefully.
2. This assignment is worth 5% of your grade.
3. Submit your assignment by 11:30 PM Saturday, February 29, 2020 8th, 2020 via Brightspace. Refer to the course syllabus to understand the policy over late assignment submissions. You can make multiple submissions, but only the last submission will be graded.
4. The goal of this assignment is to learn and practice (via programming) the concepts that we have learned so far, in particular: Strings (including indexing, slicing and string methods), control structures (if statements and for-loops), use of range function, function design and function calls.
5. Before you can start this assignment, you need to know how to use google Colab or Jupyter notebook to develop and test your code. Refer to lecture or lab material for help.
6. The only collections you can use are strings and lists. You may not use any other collection (such as a set, tuple, or dictionary) in this assignment. Using any of these in a solution to a question constitutes changing that question. Consequently, that question will not be graded.
7. Feel free to use any python interpreter i.e., IDLE, Google Colab, Jupitar….However, the following information must be included at the beginning of your assignment program:

   # Course: IT1 1120

   # Assignment number: 2

   # Due Date: 11:30 PM Saturday, February 29, 2020 8th, 2020

   # Family name, Given name

   # Student number

8. This is an individual assignment, NOT a group effort. Review and adhere to course policies and the university Plagiarism and Academic Integrity policy presented during the first lecture.

9. The assignment has 9 questions totalling 100 marks. Each question asks you to design, implement, and test a function (s). Make sure you've written good docstrings that include:

    a. Type contract

    b. Function description, including parameter names.

    c. Preconditions (if any).

10. File name must be formatted a2_xxxxxx.py (where xxxxxx is replaced with your student number). Your program must run without syntax errors. In particular, when grading your assignment, TAs will first open your file a2_xxxxxx.ipynb with either IDLE, google Colab or Jupyter notebook and press Run Module. If pressing Run Module causes any syntax error for any cell, the grade for that question will be zero.

11. For each of the functions below, test example(s) are provided to test your functions. To obtain a partial mark your function may not necessarily give the correct answer on these tests. But if your function gives any kind of python error when run on the tests provided below, that question will be marked with zero points.

12. To determine your grade, your functions will be tested both with examples provided in each question and with extra examples.

Section B: Assignment Questions

Question #1: (10 marks)

Write a function called print_factors that takes an integer n as a parameter and prints out all the factors of n, returning True if 2 is a factor of n and returning False otherwise. Recall that a factor is a number between 1 and n that goes evenly into n.

For example:
- if you enter integer: 15, you should get:

    Factors of 15 = 1 3 5 15

    False
- Or, if you enter integer: 15, you should get:

    Factors of 12 = 1 2 3 4 6 12

    True.

Question #2: (10 marks)

Write a function called triangle that takes an integer size as a parameter and prints a (size * 2 – 1) wide by size tall triangle of numbers.

For example, if you enter integer: 5, you should get:

```
123456789

 2345678

  34567

   456

    5
```

Question #3: (10 marks)

Write a function approxPi() that takes as input a float-value error and approximates constant π within error by computing the preceding sum, term by term, until the difference between the current sum and the previous sum (with one less term) is no greater than error.

- The constant π is an irrational number with value approximately 3.1415928 . . .
- The precise value of π is equal to this infinite sum: π = 4/1 − 4/3 + 4/5 − 4/7 + 4/9 − 4/11 + . . .
- We can get a good approximation of π by computing the sum of the first few term.

For example if you entre 0.01, you should get:

```
3.1465677471829556
```

Question #4: (10 marks)

Write a function named longest_name that reads number of names (as an input), names typed by the user (as an input(s)) and prints the longest name (the name that contains the most characters):

- Your method should accept an integer n as a parameter and should then prompt for n names.
- The longest name should be printed with its first letter capitalized and all subsequent letters in lowercase, regardless of the capitalization the user used when typing in the name.
- If there is a tie for longest between two or more names, use the tied name that was typed earliest.
- Also print a message saying that there was a tie, as in the right log below.
- It's possible that some shorter names will tie in length, such as John and Izzy in the left log below; but don't print a message unless the tie is between the longest names.
- You may assume that n is at least 1, that each name is at least 1 character long, and that the user will type single-word names consisting of only letters.

  Example #1:

    Enter numer of names: 2

Enter Name #1: John

Enter Name #2: Mohammad

Mohammad's name is the longest

Example #2:

Enter number of names: 3

Enter Name #1: Izzy

Enter Name #2: John

Enter Name #3: Dan

Izzy's Name is the Longest

(There was a Tie!)

Example #3

```
longest_name(7)
name #1? PeTer
name #2? eric
name #3? RAFAEL
name #4? brian
name #5? sarina
name #6? LIOR
name #7? EmIllo
Rafael's name is longest
(There was a tie!)
```

Question #5: (10 marks)

Write a function named is_fib_like that takes a list of integers as a parameter and that returns whether or not the sequence matches the pattern of the Fibonacci sequence (True if it does, False if it does not). The Fibonacci sequence begins with the number 1 followed by the number 1 and each successive value is the sum of the two previous values: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, and so on. It is possible to follow this pattern with different starting values. For example, Lucas numbers start with the values 2 and 1 but otherwise follow the Fibonacci pattern. Your function should determine whether each value after the first two is the sum of the previous two values in the sequence, returning True if the sequence has that pattern and returning False if it does not. If the list has two or fewer values, your function should return True. Below are sample lists and the value that should be returned for each:

• Contents of list passed to is_fib_like	• Value returned by is_fib_like
• []	• True
• [42]	• True
• [18, 42]	• True
• [1, 1, 1]	• False
• [1, 2, 3]	• True
• [0, 0, 0, 0, 0]	• True
• [1, 1, 2, 3, 5, 8, 13, 21]	• True
• [2, 1, 3, 4, 7, 11, 18, 29]	• True
• [1, 1, 2, 3, 5, 12, 17]	• False

Question #6: (10 marks)

Write a function called gcd that accepts two integers as parameters and returns the greatest common divisor (GCD) of the two numbers. The GCD of two integers a and b is the largest integer that is a factor of both a and b. One efficient way to compute the GCD of two numbers is to use Euclid's algorithm, which states the following:

$GCD(a, b) = GCD(b, a \% b)$

$GCD(a, 0) = Absolute\ value\ of\ a$

Question # 7: (15 marks)

Write one function called test_password() that takes any user input password and verify if the password meets the following requirements or not: length between 8 and 16 alphanumeric characters, and needs to include at least, one lower case letter, one upper case letter, number, and special characters (only @, #, $, or % are accepted).

your function should print statements tells if the password is acceptable or not Example 1:

    Enter your password: Ottawa#2020

Great, your password meets all requirements

Example 2:

Enter your password: Ottawa2020

Try again, your password does not meet all requirements.

Question # 8: (15 marks)

Write a simple function called encrypt_string( ) that takes input string (i.e. that meets the password requirements in Question #7 above) and it should returns new string (encrypted version of input string).

Hint: You can use some built-in functions on the standard Python library, e.g. chr( ) and and it's inverse; ord( ).

To validate your solution, if you use input "Ottawa@2020", your function must return "Tyyf|fE7575"

Question # 9: (10 marks)

Write a simple function called decrypt_string( ) that takes input string (the encrypted one in question #8 above) and it should returns the original encrypted string.

To validate your solution, if you use input "Tyyf|fE7575", your function must return "Ottawa@2020".