uOttawa

ITI1120 F - 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 Sunday, March 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. Assignments must be submitted in a notebook format (file extension .ipynb) using free resources: either google Colab or Jupyter. The following information must be included at the beginning of your assignment program:

   # Course: IT1 1120

   # Assignment number: 2

   # Due Date: Sunday March 08, 2020 11:30 PM

   # 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. Make sure you use the provided master notebook file to document all answers to all questions. File name must be formatted a2_xxxxxx.ipynb (where xxxxxx is replaced with your student number). You must have minimum one code cell to answer each question. Your submission should contain AT LEAST 9 cells. Your program must run without syntax errors. In particular, when grading your assignment, TAs will first open your file a2_xxxxxx.ipynb with either 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: Wassim

Wassim'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!)

Question #5: (10 marks)

Write a function is_prime(n) that returns a Boolean value indicating whether an integer is prime or not. your function should take the positive integer number as an input and print if the number is prime or not.

For example:

Enter positive integer number: 9

Number 9 is not Prime

Question #6: (10 marks)

Write function vowelCount() that takes a string as input and counts and prints the number of occurrences of vowels in the string.

vowelCount('Le Tour de France')

a, e, i, o, and u appear, respectively, 1, 3, 0, 1, 1 times.

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".