School of Information Systems
Academic Year 2019/20
Term I

IS111: Introduction to Programming

Lab Exercise 03: For Loops and Strings

Optional submission deadline: **Sun 08 Sep 2019 @ 2359 hrs**

INSTRUCTIONS

You are expected to read through the accompanying lesson 03 notes before attempting lab exercise 03. You should attempt the questions using **Visual Studio Code**. Each question should be submitted on a different file.

Please note that submission instructions have changed. Please **do not** submit Level 1 questions; use the discussion forum for that.

To submit, you should:

1. Name your 3 files as **week3-qn2-1.py**, **week3-qn2-2.py** and **week3-qn3-1.py**.
2. Submit the 3 .py files to **eLearn → Assignments** by the stipulated deadline.

Level 1 Questions (no submission required; use the discussion forum)

1.1 Negative Numbers

Write a code snippet that asks the user for a **positive** integer *n*, and then displays the numbers from -*n* (negative *n*) to 0, separated by a space ' ' between them. Here is an example of how the output may look like.

```
Input a positive integer: 10
-10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
```

1.2 Decreasing Numbers

Write a code snippet that asks the user for a **positive** integer *n*, and then displays the numbers from *n* to 0 (i.e., decreasing order), separated by a space `' '` between them. Here is an example of how the output may look like.

```
Input a positive integer: 10
10 9 8 7 6 5 4 3 2 1 0
```

1.3 Sum of Square of Numbers

Write a code snippet that asks the user for a **positive** integer *n*, and then displays the sum of the square of numbers from 1 to *n*. For instance, if n=6, then the sum is given by $1^2+2^2+3^2+4^2+5^2+6^2=91$. Here is an example of how the output may look like.

```
Input a positive integer: 6
Sum of square of numbers from 1 to 6 is 91
```

1.4 Factorial

The factorial of a number *n* is given by *n*`!=1x2x3...`*n*.

Write a code snippet that asks the user for a positive integer *n*, and then displays the factorial of *n*. Here is an example of how the output may look like.

```
Input a positive integer: 5
5! is 120
```

1.5 String Swap

Write a code snippet that asks the user for a string and returns a new string with all the '*z*' swapped to '*s*', and vice versa. The upper/lower casing of each character should be preserved, as well as all other character types (e.g., numbers and punctuation marks).

Here is an example of how the output may look like.

```
Input a string: The quick brown fox jumps over the lazy dog.
The quick brown fox jumpz over the lasy dog.
```

Level 2 Questions

2.1 Palindrome Checker

A palindrome is a string that reads the same, either forward or backward. Some examples of palindromes are '*madam*', '*noon*', '*Was it a car or a cat I saw*', and '*step on no pets*'.

Write a code snippet that asks the user for a string `s` and returns `True` if `s` is a palindrome, and `False` otherwise (if `s` is not a palindrome).

Here is an example of how the output may look like.

```
Input a string: Was it a car or a cat I saw
True
```

Here is another example of the output.

```
Input a string: Is this exercise difficult?
False
```

📌 You may want to try out `s[::-1]` (string slicing with a negative step/stride) and explore how it works; it might be useful in your solution.

💬 You do not need to cater for punctuation in this exercise. However, you will need to take care of the whitespaces.

2.2 String Rotation

Write a code snippet that asks the user for a string *s* and an integer *n*. You should then rotate the string *s* by *n* characters in a clockwise manner if *n* is positive, and an anti-clockwise manner if *n* is negative.

Here are some examples of how the output may look like with positive values of *n*.

```
Input a string: alphabets
Input an integer: 4
betsalpha
```

```
Input a string: The questions are always more important than the answers.
Input an integer: 3
rs.The questions are always more important than the answe
```

Here is another example of how the output may look like with negative values of *n*.

```
Input a string: alphabets
Input an integer: -4
abetsalph
```

```
Input a string: The questions are always more important than the answers.
Input an integer: -5
uestions are always more important than the answers.The q
```

3 Brownie Points 🍪

3.1 Password Strength

A strong password should have the following criteria:

- Be at least 8 characters long
- Contain a lower case letter (a-z)
- Contain an upper case letter (A-Z)
- Contain a number (0-9)
- Contain a special character (!@#$)

Write a code snippet that asks the user for a password, and then checks for the strength of the password. You should print `True` if the password is strong, and `False` otherwise.

Here is an example of how the output may look like with a strong password.

```
Input a password: ThisIsAStrongPassword123!
True
```

Here is an example of how the output may look like with a weak password.

```
Input a password: ThisIsAStrongPassword
False
```

💬 You should be able to complete your solution in 30 lines of code (or less).

💾 02 Sep 2019