CS 115 - Introduction to Programming in Python

Lab 02

Lab Objectives: Strings, Loops, Nested Loops

Instructions: For this assignment, you can use your favorite IDE (Spyder or Jupyter recommended). Upload your solutions as a single .zip file to the Lab01 assignment for your section in Moodle before the end of your lab session. Use the following naming convention: **SS_Lab01_Surname_FirstName.zip** where SS is the section number 01, 02, 03, ..., & and Surname is your family name, & FirstName is first name. You must attend the lab Zoom session. You must show and explain your solutions to your TA during your lab session and must answer their questions to get your grade by the end of your lab session (the week of Oct 18).

Students who do not attend the lab Zoom session but submit will get 0.

1. Write a program, `Lab02_yourname_Q1.py`, that inputs an integer number from the user and prints the factors of that number, as shown in the sample runs below.

Sample Run 1: (User inputs are red)	**Sample Run 2:**
Enter an int: 18 Factors of 18: 1 , 2 , 3 , 6 , 9 , 18	Enter an int: 23 Factors of 23: 1 , 23
Sample Run 3:	**Sample Run 4:**
Enter an int: 36 Factors of 36: 1 , 2 , 3 , 4 , 6 , 9 , 12 , 18 , 36	Enter an int: 125 Factors of 125: 1 , 5 , 25 , 125

2. Write a program, `Lab02_yourname_Q2.py`, that prompts the user to enter a string until an empty string is entered. For each input string, it displays a new string created by collecting even position letters followed by odd position letters from the input string.

Sample Run: (User inputs are red)	
Enter a string: abcdefgh new string is acegbdfh Enter a string: 0123456789 new string is 0246813579 Enter a string: baby new string is bbay	Enter a string: baby bear happy :-) new string is bb erhpyayba ap Enter a string: tea yippee!! new string is taype!e ipe! Enter a string: bye for now... new string is befrnw.y o o.. Enter a string: done!

3. Write a program, `Lab02_yourname_Q1.py`, that continually prompts the user for a desired sum until 0 is entered. For each desired sum input value, it repeatedly rolls two six-sided dice until their sum is the desired sum and reports the number of rolls to achieve the desired sum. It should validate input for the desired sum. For example, 1 cannot be a valid desired sum.

Sample Run: (User inputs are red)

```
Desired dice sum: -4
Invalid dice sum, try again...

Desired dice sum: 1
Invalid dice sum, try again...

Desired dice sum: 6
4 rolls

Desired dice sum: 2
58 rolls

Desired dice sum: 10
4 rolls

Desired dice sum: 5
11 rolls

Desired dice sum: 12
11 rolls

Desired dice sum: 13
Invalid dice sum, try again...

Desired dice sum: 20
Invalid dice sum, try again...

Desired dice sum: 4
1 rolls

Desired dice sum: 4
9 rolls

Desired dice sum: 0
bye!
```