CS 115 - Introduction to Programming in Python

Lab 02

Lab Objectives: Strings, Loops, Nested Loops, Numerical Programs

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 on Moodle before the end of your lab session. Use the following naming convention: **SS_Lab02_Surname_FirstName.zip** where SS is the section number 01, 02, 03, ..., & and Surname is your family name, & FirstName is first name. 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 14).

Important Note: You must be present in the lab from the beginning of your lab sessions. **Otherwise, you will get 0 from this lab assignment.**

1. Write a program that implements a *Caesar cipher* (*rotation cipher)*, which is a simple system of encoding strings by shifting every letter forward by a given number. Your program should prompt the user to type a message and an encoding "key" (number of places to shift each character) and display the shifted (uppercase) message. For example, if the shift amount is 3, then the letter A becomes D, and B becomes E, and so on. Letters near the end of the alphabet wrap around for a shift of 3, X becomes A, and Y becomes B, and Z becomes C. Sample runs below show the user input in red.

   Useful functions:

   - **`ord( ch )`**: takes a single character as a parameter and returns an integer representing Unicode code point for the given Unicode character. Ex: `ord('a')` returns 97.
   - **`chr( val )`**: takes an integer value as a parameter and returns a single character (string) whose Unicode code point is the integer. Ex: `chr( 97 )` returns `'a'`.

Sample Run 1:	**Sample Run 1:**
Your message? abc defgh xyz. Encoding key? 3 DEF GHIJK ABC.	Your message? def ghıjk abc. Encoding key? -3 ABC DEFGH XYZ.
Sample Run 3:	**Sample Run 4:**
Your message? def ghıjk abc. Encoding key? -3 ABC DEFGH XYZ.	Your message? 1aabbcc 2defgaaa xyzbbxyz! Encoding key? 4 1EEFFGG 2HIJKEEE BCDFFBCD!

2. Write a program that prompts the user for a positive integer *n* and displays a new integer whose value is similar to *n*'s but with each pair of digits swapped in order. For example, for the input 123456, it displays 214365. Notice that the 5and 6 are swapped, as are the 3 and 4, and the 1 and 2. If the number contains an odd number of digits, leave the leftmost digit in its original place. For example, for the input 1234567, the program displays 1325476.

   Note: Do not use strings to solve this problem. A sample run below shows the user input in red.

   Sample Run:

   int? 5

   5

   new int? 12

   21

   new int? 123

   132

   new int? 1234

   2143

   new int? 12345

   13254

   new int? 123456

   214365

   new int? 12345678

   21436587

   int? 123456789

   132547698

   new int? 21436587

   12345678

   new int? 12345678010101

   21436587101010

   new int? 0

3. Write a program that asks the user how many rectangles they want and then prompts them for a width and height for each rectangle. It then outputs all the rectangles made of ascii stars and their total area. Assume that the user will provide valid positive integers for each value when prompted. Hint: first try to do it for one rectangle.

Note: Do not use string repetition operator to solve this problem. A sample run below shows the user input in red.

Sample Run:

```
How many rectangles? 3
Width 1? 5
Height 1? 3
*****
*****
*****

Width 2? 6
Height 2? 4
******
******
******
******

Width 3? 3
Height 3? 5
***
***
***
***
***
Total area: 54
```