DATA STRUCTURES LAB
CSE - 2018-22
Level 2 :Exercises in Python

1. Write a Python program to calculate the length of a string

2. program to count the number of characters (character frequency) in a string.

Sample String : 'OpenLabInPython'
Expected Result :
{'a': 1, 'o': 1, 'b': 1, 'e': 1, 'P': 1, 'I': 1, 'h': 1, 'L': 1, 'O': 1, 'n': 3, 'p': 1, 't': 1, 'y': 1}

3. program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

Sample String: 'OpenLabInPython'
Expected Result : Opon

Sample String: 'In'
Expected Result : InIn

Sample String: 'I'
Expected Result : *Empty String*

4. program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself

Sample String : 'restart'
Expected Result : 'resta$t'

5. Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string
Sample String : 'abc', 'xyz'
Expected Result : 'xyc abz'

6. Write a Python function that takes a list of words and returns the length of the longest one.

Sample Input: ["PHP", "Exercises", "Backend"]
Expected Result : Exercises

7. Write a Python script that takes input from the user and displays that input back in upper and lower cases.

8. Write a Python function to reverses a string if it's length is a multiple of 4.

9. Write a Python program to create a Caesar encryption.

Note : In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by

A, E would become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence.

10.Write a python function to find maximum depth of nested parenthesis in a string
Sample Input: "(((X)) (((Y))))"

To Do: Find the maximum depth of balanced parenthesis, like 4 in above example. Since 'Y' is surrounded by 4 balanced parenthesis.

If parenthesis are unbalanced then return -1.