CSE110 Lab Assignment 3

This Assignment is to help you develop your concept of Strings in Python.

Write your name, student ID and CSE110 theory section number below:

In [ ]:

```
#STUDENT NAME: M. ASADUZ ZAMAN
#STUDENT ID: 21101050
#CSE110 THEORY SECTION: 10
```

Write the code in Python to do the following tasks:

[MUST MAINTAIN VARIABLE NAMING CONVENTIONS FOR ALL THE TASKS]

Task 1

Write a Python program that will ask the user to input a string (containing exactly one word). Then your program should print subsequent substrings of the given string as shown in the examples below.

==

Example 1:\ Input: BANGLA\ Output:\ B\ BA\ BAN\ BANG\ BANGL\ BANGLA

==

Example 2:\ Input: DREAM\ Output:\ D\ DR\ DRE\ DREA\ DREAM

==

Hints(1): Need to use "for loop" for this task.

Hints(2): Need to use print() function for printing newlines.

For example:\ print(1)\ print(2)

Output:\ 1\ 2

==

We need use print(end = "") to skip printing the additional newline.

For example:\ print(1, end =" ")\ print(2)

Output:(prints the following output right next to the previous one)\ 12

==

In [4]:

```python
string= input("enter a string")
length = len(string)
for row in range (length):
    for col in range(row+1):
        print(string[col],end="")
    print()
```

```
enter a stringDREAM
D
DR
DRE
DREA
DREAM
```

Task 2

Write a Python program that will ask the user to enter a word as an input.

- If the length of the input string is less than 4, then your program should print the same string as an output.
- If the input string’s length is greater than 3, then your program should add "er" at the end of the input string.
- If the input string already ends with "er", then add "est" instead.
- If the input string already ends with "est", then your program should print the same input string as an output.

===

Example 1:\ Input: strong\ Output: stronger

===

Example 2:\ Input: stronger\ Output: strongest

===

Example 3:\ Input: strongest\ Output: strongest

===

Example 4:\ Input: abc\ Output: abc

===

In [1]:

```python
string=input()
if len(string)<4:
  print(string)
else:
  if string[len(string)-2:len(string)] == 'er':
    print(string[:-2]+"est")
  elif string[len(string)-3:len(string)]=='est':
    print(string)
  else:
    print(string+"er")
```

```
strong
stronger
```

Task 3

Write a Python program that will ask the user to input a string (containing exactly one word). Then print the ASCII code for each character in the String using the ord() function.

To check if your program is working correctly or not, you can find a list of all correct values from the following website. Look at “Dec” and “Char” columns only, ignore other columns.\ link: http://www.asciitable.com/ (http://www.asciitable.com/)

===

Example 1:\ Input: Programming\ Output:\ P : 80\ r : 114\ o : 111\ g : 103\ r : 114\ a : 97\ m : 109\ m : 109\ i : 105\ n : 110\ g : 103

===

Example 2:\ Input: hunger\ Output:\ h : 104\ u : 117\ n : 110\ g : 103\ e : 101\ r : 114

===

In [1]:

```
string = input("enter any string: ")
for c in string :
    print((c),ord(c))
```

```
enter any string: hunger
h 104
u 117
n 110
g 103
e 101
r 114
```

Task 4

Take a string as a input from the user with all small letters. Then print the next alphabet in sequence for each alphabet found in the input.

===

Hint: You need to use functions ord() and chr(). The ASCII value of ‘a’ is 97 and ‘z’ is 122.

===

Example 1:\ Input: abcd\ Output: bcde

===

Example 2:\ Input: the cow\ Output: uif!dpx

===

Example 3: [Must fulfil this criteria]\ Input: xyzabc\ Output: yzabcd

===

In [1]:

```
text=input("enter a text: ")
for i in text:
    if ord(i)==122:
        print("a",end="")
    else :
        print(chr(ord(i)+1),end="")
```

```
enter a text: abcd
bcde
```

In [2]:

```
text=input("enter a text: ")
for i in text:
    if ord(i)==122:
        print("a",end="")
    else :
        print(chr(ord(i)+1),end="")
```

```
enter a text: the cow
uif!dpx
```

In [2]:

```
text=input("enter a text: ")
for i in text:
    if ord(i)==122:
        print("a",end="")
    else :
        print(chr(ord(i)+1),end="")
```

```
enter a text: xyzabc
yzabcd
```

Task 5

Write a Python program that will ask the user to enter two strings (s1, s2) as an input. Then create a mixed string with alternative characters from each string. Any leftover chars will be appended at the end of the resulting string.

==

Hint: For adding the leftover characters you can use string slicing.

==

Example 1:\ Input:\ "ABCD"\ "efgh"

Output:\ AeBfCgDh

==

Example 2:\ Input:\ "ABCDENDFGH"\ "ijkl"

Output:\ AiBjCkDlENDFGH

==

Example 3:\ Input:\ "ijkl"\ "ABCDENDFGH"

Output:\ iAjBkClDENDFGH

==

In [3]:

```
s1=input("enter a text= ")
s2=input("enter another text= ")
i=0
j=0
sum=''
while i<len(s1) or j<len(s2):
    if i<len(s1):
        sum=sum+s1[i]
        i+=1
    if j<len(s2):
        sum=sum+s2[j]
        j+=1
print(sum)
```

```
enter a text= ABCDEF
enter another text= ijkl
AiBjCkDlEF
```

Task 6

Given a string, create a new string with all the consecutive duplicates **removed**.

===

Hint: You may make a new string to store the result. You can check whether the current character and the next character are the same, then add that character to the new string.

===

Sample Input:\ AAABBBBCDDBBECE

Sample Output:\ ABCDBECE

===

In [7]:

```
s=input("enter a String")
s1=""
for i in range (len(s)):
  if i==0:
    s1+=s[i]
  else:
    if s[i-1]!=s[i]:
      s1+=s[i]
print(s1)
```

```
enter a Stringaaaabbbbbbbbccccccc
abc
```

Task 7

Write a python program that takes 2 inputs from the user, where the first input is a string with length greater than 1. The second input is the index of the first given string from where you have to start reversing. Then print the new string back to the user.

==

Sample Input 1:

72418

4

Sample Output 1:

81427

Explanation: Our second input, index '4' is the last index of our first input String '72418', hence the entire string is reversed giving us '81427'.

==

Sample Input 2:

12345

2

Sample Output 2:

32145

Explanation: The second input is '2' so we have to reverse from index 2 of our first input. The 2nd index of our first input String is '3', index 1 is '2' and index 0 is '1'. Hence, if we reverse indexes 0 to 2, we get '321'. Index 3 and 4 which is '4' and '5' respectively remains unchanged hence our final output is '32145'.

==

Sample Input 3:

aBcd1234defg

5

Sample Output 3:

21dcBa34defg

Explanation: From our first input String 'aBcd1234defg',

index 0 = 'a'

index 1 = 'B'

index 2 = 'c'

index 3 = 'd'

index 4 = '1'

index 5 = '2'

index 6 = '3'

Since our second input is 5, index 0 to index 5 is reversed and we have '21dcBa' and the rest is unchanged from indexes 6 to 11 ('34defg'). Therefore, we have '21dcBa34defg' finally.

===

In [6]:
```
text=input()
n=int(input())
print(text[n::-1]+text[n+1::])
```

```
aBcd1234defg
5
21dcBa34defg
```

Task 8

Write a python program that splits a given string on a given split character. The first input is a String and the second input is the character that will be used to split the first String. [You cannot use the built-in split function]

==

Sample Input 1:
This-is-CSE110
-

Sample Output 1:
This
is
CSE110

Explanation: The second input which is the character '-', is used to split or divide the first input String 'This-is-CSE110' into 'This', 'is' and 'CSE110' which are printed individually in seperate lines.

==

Sample Input 2: tom@gmail,harry@yahoo,bob@gmail,mary@gmail

,

Sample Output 2:

tom@gmail

harry@yahoo

bob@gmail

mary@gmail

==

In [5]:

```python
text=input()
x=input()
for i in text:
    if i==x:
        text=text.replace(i,'\n')
print(text)
```

```
this-is-my-car
-
this
is
my
car
```

Task 9

Suppose you are given two strings, s1, and s2. Now, print a new string made up of the last characters and then the first characters of the input strings.

===

Sample Input 1:

s1 = new

s2= string

Sample Output 1:

gwsn

Explanation: The last character of the String s2 is 'g'. The last character of the String s1 is 'w'. The first character of the String s2 is 's'. The first character of the String s1 is 'n'. Together they give us the ouput we want 'gwsn'.

===

Sample Input 2:

s1 = abcd

s2= efgh

Sample Output 2:

hdea

Explanation: The last characters of the Strings s2 and s1 is 'h' and 'd' respectively while the first characters of the Strings is 'e' and 'a' respectively. Together they give us the ouput we want 'gwsn'.

===

In [4]:

```python
x=input("enter first string ")
y=input("enter second string ")
print(y[-1]+x[-1]+y[0]+x[0])
```

```
enter first string new
enter second string car
rwcn
```

Task 10

Write a Python program that takes a String as input from the user, removes the characters at even index and prints the resulting String in uppercase.

==

Sample Input 1:

String

Sample Output 1:

TIG

Explanation: The characters 'S', 'r' and 'n' are at index positions 0, 2, and 4 respectively. Hence they are removed and the remaining characters 'tig' are capitalized giving us output 'TIG'.

==

Sample Input 2:

abcd

Sample Output 2:

BD

==

In [2]:

```
string = input("enter any string ")
s1=''
for i in range (0,len(string)):
    if i%2!=0 :
        s1+=string[i]
print(s1.upper())
```

```
enter any string abcd
BD
```

Task 11

Write a python program that takes two inputs. The first input is a string and the second input is a number. If the number is even then concatenate the given string two times the given number and if the number is odd then concatenate the given string three times the given number.

===

Sample Input 1:

CSE110

4

Sample output 1:

CSE110CSE110CSE110CSE110CSE110CSE110CSE110CSE110

Explanation: The second input which is the number 4 is even, therefore the first string input 'CSE110' is concatenated(joined together) 4*2 = 8 times.

===

Sample Input 2:

CSE110

3

Sample Output 2:

CSE110CSE110CSE110CSE110CSE110CSE110CSE110CSE110CSE110

===

In [1]:

```
text=input()
n=int(input())
if n%2==0:
    print(text*(n*2))
else:
    print(text*(n*3))
```

```
cse110
5
cse110cse110cse110cse110cse110cse110cse110cse110cse110cse110cse110cse110cse11
0cse110cse110
```

Task 12

Write a python program that takes 2 inputs from the user. The first input is a string and the second input is a letter. The program should remove all occurences of the letter from the given string and print the output. If the letter is not found in the string and the length of string is greater than 3, then remove the first letter and last letter of the given string and print it. Otherwise print the string as it is. [You can assume that all the input will be in lowercase letter]

===

Sample Input 1:

tanjiro kamado

a

Sample output 1:

tnjiro kmdo

Explanation: All 3 instances of the character 'a' is removed from the input String 'tanjiro kamado' to give us output 'tnjiro kmdo'.

===

Sample Input 2:

eren yeager

k

Sample Output 2:

ren yeage

Explanation: The character 'k' is absent in the first input String 'eren yeager' and it's length is 11 which is greater than 3 therefore the first character 'e' and the last character 'r' is removed. Hence, the final String is 'ren yeage'.

===

Sample Input 3:

hi

a

Sample Output 3:

hi

Explanation: The letter 'a' is not found in our first input 'hi', the length of which is 2. Since the character is not present and the length is less than 3, we print the String 'hi' as it is.

===

In [2]:

```
text=input()
letter=input()
if letter in text:
    print(text.replace(letter,''))
elif len(text)>3:
    print(text[1:len(text)-1])
else:
    print(text)
```

```
tanjiro kamado
a
tnjiro kmdo
```

Optional Tasks (13 - 15) [Ungraded]

Task 13

Write a python program that takes a string as an input from the user and then modifies the string in such a way that the string always starts with an uppercase letter and the case of each subsequent letter is the opposite of the previous letter (uppercase character followed by a lowercase character followed by an uppercase character and so on). Finally the modified string is printed to show the user.

Hints: Flags/counters can be used to manage uppercase-lowercase.

==

Sample Input 1:\ Python programming is very easy

Sample Output 1:\ PyThOn PrOgRaMmInG iS vErY eAsY

==

Sample Input 2:\ I   love   Python   Programming

Sample Output 2:\ I   lOvE   pYtHoN   pRoGrAmMiNg

==

Sample Input 3:\ CSE110 Course

Sample Output 3\ CsE110 cOuRsE

==

Sample Input 4:\ c

Sample Output 4:\ C

==

```
In [ ]: #to do
```

Task 14

An anagram is a play on words created by rearranging the letters of the original word to make a new word or phrase. So we can say two words are anagrams if they contain all of the same letters, but in a different order.

Write a python program that takes two strings from the user and tells if they are anagram or not.

===

Sample Input 1:
dusty
study
Sample Output 1:
They are anagram

===

Sample Input 2:
dustyyy
study
Sample Output 2:
They are not anagram

In [ ]:

```
# to do
```

Task 15

Write a python program thats prints the largest consecutive sequence from a given string S. For example, if S = ‘AABBBCCAAAA’, print ‘AAAA’. If more than one largest consecutive sequence exists, print the first one. Print NONE if no consecutive sequence is found.

==

Sample Input 1:
AABBBCCAAAA
Sample Output 1:
AAAA

==

Sample Input 2:
AABBCC
Sample Output 2:
AA

==

Sample Input 3:
ABCD
Sample Output 3:
NONE

==

In [ ]:

```
#to do
```