**NAME – TAUSIF KHAN**

**ROLL NO- 12**

**SUBJECT- OSTL**

**S.E COMPS- B**

**Experiment No- 1**

**Aim:** WAP to demonstrate Strings in Python:

a) A python program to access each element of a string in forward and reverse orders

and check if the entered string is a palindrome.

b) A Python program to find the length of a string without using len() function.

c) A Python program to know whether a sub string exists in main string or not.

d) A Python program to insert a sub string in a string in a particular position.

**Tools Used:** Python 3.4.3, Terminal

**Theory:**

1. Explain Strings in Python

Ans: Strings in Python are arrays of bytes representing unicode characters. However, Python does not have a character data type, a single character is simply a string with a length of 1. Square brackets can be used to access elements of the string.

1. Explain and describe different cases in String

Ans: Sometimes, we might have a problem like this in which we need to convert the odd character of string to upper case and even positioned characters to lowercase.

Method #1 : Using upper() + lower() + loop  
This task can be performed in brute force method in a way that we iterate through the string and convert odd elements to uppercase and even to lower case using upper() and lower() respectively.

Method #2 : Using list comprehension  
This is the shortened one liner approach to this problem. It uses same logic as above but in much more compact way.

1. Explain and describe different methods to find a sub string in main string.

Ans: Python String find()

The find() method returns the lowest index of the substring if it is found in given string. If its is not found then it returns -1.

**Syntax :**

str.find(sub,start,end)

sub : It’s the substring which needs to be searched in the given string.  
start : Starting position where sub is needs to be checked within the string.  
end : Ending position where suffix is needs to be checked within the string.

NOTE : If start and end indexes are not provided then by default it takes 0 and length-1 as starting and ending indexes where ending indxes is not included in our search.

**CODES & OUTPUTS :-**

**A python program to access each element of a string in forward and reverse orders and check if the entered string is a palindrome.**

**Code:**

x=input("Enter the String: ")

print("The String is " ,x)

x=x.strip()

rev=x[-1 : : -1]

print("The Reverse String is",rev)

l=len(rev)

rev=rev.lower()

x=x.lower()

if rev==x:

print("Is a palindrome")

else:

print("Not a palindrome")

**Output:**

Enter the String: Apple

The String is Apple

The Reverse String is elppA

Not a palindrome

**A Python program to find the length of a string without using len() function.**

**Code:**

s="THIS IS PYTHON"

c=0

w=1

for i in s:

c+=1

if(i==' '):

w+=1

print("THE STRING IS:",s)

print("THE NUMBER OF LETTERS ARE:",c)

print("THE NUMBER OF WORDS ARE:",w)

**Output:**

THE STRING IS: THIS IS PYTHON

THE NUMBER OF LETTERS ARE: 14

THE NUMBER OF WORDS ARE: 3

**A Python program to know whether a sub string exists in main string or not.**

**Code:**

s1="THIS IS PYTHON"

s2="IS"

n=s1.find(s2,0,len(s1))

if n==-1:

print("THE STRING IS ABSENT")

else:

print("THE STRING IS PRESENT")

print("THE POSITION IS",n)

**Output:**

THE STRING IS PRESENT

THE POSITION IS 2

**A Python program to insert a sub string in a string in a particular position.**

**Code:**

s1="TUESDAY IS A DAY"

s2="HOLI"

n=13

new=[]

for i in range(0,n):

new.append(s1[i])

for i in s2:

new.append(i)

for i in range(n,len(s1)):

new.append(s1[i])

new=''.join(new)

print(new)

**Output:**

TUESDAY IS A HOLIDAY

**CONCLUSION** – Thus we have studied how to implement strings in python.