Page No.:
Date: / /

Name- Sushilkumar D. Dhamane.
Class- SE1 Subj- DSL
Roll NO- 21123.

Problem statement -
Write a python program to compute following operation on string
a) To display word with the longest word.
b) To determine the frequency of occurance of a particular character in the string.
c) To check whether given string is palindrome or not.
d) To display index of first apperance of the substring.
e) To count the occurance of each word in a given string.

Objectives:
1) To understand the concepts of operations on string.
2) To understand primitive function of list data structure in python.

Outcomes:
1) To implement string operation using list data structure in python.
2) To write menu driven, modular program in python.
3) To implement user defined function in python.
4) To find time & space complexity of an algorithm.

Hardware & software Requirement:
1) operating system - 64bit.
2) programming language - Python
3) programing tool / IDE - pycham

Theory:

String operations:

A string is a datatype used in programming such as an integer & floating point unit, but is used to represent text rather than numbers It is comprised of a set of characters that can also contain space & Numbers.

Basic operations:

1) On a computer, string matching algorithm is used to search files/folders with a particular name.
2) This string matching algorithm is also used in detecting plagiarism particularly in case of reasearch papers.
3) Strings are important in bioinformatics for dealing with sequence of information.

Algorithm.

1) Algorithm for longest length.

1. start
2. read the string from user name it has str1.
3. split the str1 and name it as list.
4. initialize n=0 for particular word.
5. compare the first word with every element in list then
6. Read the longest word from list.
7. Repeat step 5 & 6 for every word in list.
8. write (longest word)
9. stop.

2) Algorithm for frequency for occurance:

1. Start.
2. Read the string and character form user name it has str1 and char.
3. Initialize the count variable as zero.
4. compare the character with every element in the string
5. and then count the character
6. write count.

3) Algorithm for Palindrome:

1. read the word for which the palindrome to be found.
2. Reverse the string which is entered by user.
3. Compare the reverse string with entered string.
4. If it's equal then it's palindrome else not palindrome.

4) Algorithm for indexing the character:

1. Read the string as str1 and read substring
2. Initialize index as zero.
3. compare the Character of substring with the str1
4. If they are equal then move to the other element
5. If not then write zero.
6. If the other element is also compared then count index.
7. Recipeat step 3 to 6 for every element in substring.
8. write index

5) Algorithm for occurance of each word:

1. Read string from user and split it.
2. Read the each word to the string which is spilted.
3. Compare each word with other element with other if its Repeat the count.
4. If not then write one.

Page No.:
Date: / /

5. Repeat step 3 & 4 for each word in splited list.
6. write (word, count)

Pseudocode:

1) Pseudocode for longest length.

```
str1 = Input ("Enter the string")
list = str1.split()
print(list)
for i in range (len(list))
        if n < len(list):
            n = len(list)
        print(list[i])
```

2) Pseudocode for palindrome.

```
Str = Input("Enter word")
m = str[::-1]
IF   str == m then
        print ("It's palindrome")
  else:
        ("It's not palindrome")
```

3) Pseudocode for frequency for occurance.

```
Str1 = Input ("Enter string")
char = Iput ("Enter character")
print (str1)
count = 0
for i in range (len(str1))
    if str[i] = = char :
            count = count + 1
    print (char, count)
```

4) Pseudocode for Indexing of character:

```
str1 = Input("Enter string")
sub_string = Input("Enter substring")
sub len = len(sub_string)
index = 0
j = 0
for i in range(len(str1)):
    if sub_string == str1[i]:
        j = j + 1
        if (j == sublen):
            index = i - (sublen-1)
            break
        else:
            j = 0
Print(Index)
```

5) Pseudocode for Occurance of each word:

```
Str1 = Iput("Enter the string")
Str1 = str1.spilt()
sprint(str1)
j = 0
while (i<len(str1)):
    count = 0
    For j in str1:
        if str1[i] == j:
            count = count + 1
    print(count, times, str1[i])
```

Application:

1) String manipulation can be used for manupating the input recived from various online form.
2) This can be further used for Natural Language processing.
3) These problen statements also help in improving problem solving skills.

Conclusion:

1) All the string operations were performed successfully.
2) I learnit how to write code for built in function that I was using before.

```
print("Write a python program to compute following operation in string:")
print("a)to display word with the longest length")
print("b)to determine the frequency of occurrence of particular character in the string")
print("c)to check whether the string is palindrome or not")
print("d)to display index of first appearance of the substring")
print("e)to count the occurrence of each word in a given string")



##b)) to dtermine the letter how many times is present
#---> c))) palindrome means== consider the string "aba" if we read it will be same as aba
# similary for "madam" it will be same from rear side if we read ,

def longest():       #to print longest length of word in given string
   str1=input("Enetr the string : ")
   list=str1.split()
   print(list)
   n = 0         # n is the length of particular word after spliting.
   for i in range(len(list)):
      if n<len(list[i]):
         n=len(list[i])
   print("longest length word is : ",list[i])

def frequency():     #to calculate the single char how many times is present ,for that we need to
   # compare the char with given string(str1)
   str1=input("Enter the string:")
   char=input("Enter the character to be count:")
   print("Given string:",str1)
   count=0
   for i in range(len(str1)):
```

```
        if char==str1[i]:       #compare the char with each letter in str1(for identify the each letter)
            count=count+1
    print(char,"is",count,"times present in given string.")

def palindrome():     #the revese word must be same eg. madam,dad
    str=input("Enter the word:")
    m=str[::-1]
    if str==m:
        print("It's a palindrome")
    else:
        print("It's not a palindrome")

def apperance():
    str1=input("Enter the string:")
    sub_string=input("Enter the sub_string:")
    sublen=len(sub_string)
    index=0      #index denotes the indexing of str1.
    j=0          # j denotes the indexing of sub_string.
    for i in range(len(str1)):
        if (sub_string[j]==str1[i]):
            j=j+1
            if (j==sublen):
                index=i-(sublen-1)
                break
        else:
            j=0
    print("sub_string index:",index)

def occurance():
```

```
    str1 = input("enetr the string:")
    str1 = str1.split()
    print(str1)
    i = 0
    while (i<len(str1)):
        count = 0
        for j in str1:
            if str1[i]==j:
                count=count+1
        print(str1[i],count,"times")
        i = i + 1



while(True):
    choice=int(input("Enter the choice:"))
    if (choice==1):
        longest()
    elif (choice==2):
        frequency()
    elif (choice==3):
        palindrome()
    elif (choice==4):
        apperance()
    elif (choice==5):
        occurance()
    else:
        break
    stop = input("would you like to continue(y/n):")
```

```
    if (stop == "n"):
        print("THANK YOU!!")
        break
```

OUTPUT:

Write a python program to compute following operation in string:

a)to display word with the longest length

b)to determine the frequency of occurrence of particular character in the string

c)to check whether the string is palindrome or not

d)to display index of first appearance of the substring

e)to count the occurrence of each word in a given string

Enter the choice:1

Enetr the string : pune institue of coputer tecnology

['pune', 'institue', 'of', 'coputer', 'tecnology']

longest length word is : tecnology

would you like to continue(y/n):y

Enter the choice:2

Enter the string:pict ranks second from pune

Enter the character to be count:p

Given string: pict ranks second from pune

p is 2 times present in given string.

would you like to continue(y/n):y

Enter the choice:3

```
Enter the word:Madam
It's not a palindrome
would you like to continue(y/n):y
Enter the choice:3
Enter the word:Hello
It's not a palindrome
would you like to continue(y/n):y
Enter the choice:4
Enter the string:my self sushilkumar dhamane
Enter the sub_string:s
sub_string index: 3
would you like to continue(y/n):y
Enter the choice:5
enetr the string:hello everyone its sushil
['hello', 'everyone', 'its', 'sushil']
hello 1 times
everyone 1 times
its 1 times
sushil 1 times
would you like to continue(y/n):n
THANK YOU!!
```