* Assignment No-2 *

AIM:-

To implement the string operation in python.

OBJECTIVE:-

(1) To understand the use standard library function for string operations.
(2) To perform string operations.

Input:-

one or two strings.

Output:-

Resulting string after performing string operations.

Theory:-

String:-

It is defined as an array of characters or a pointer to charaters. It simply a sequence of characters.

A charater is simply a symbol. for example, the english alphabet has 26 charaters, Computer only deals with binary numbers. Even though you can may see charaters on your screen, internally it is stored and manipulate as a combination of 0's and 1's. this convesion of charaters to a number is called encoding and the reverse process decoding. ASCII on unicode are same of the popular encoding used.

In python, a string is a sequence of

unicode characters unicode was introduced to include every character in all language and bring uniformily in encoding.

Null - terminated Strings:-

String are terminated by special character which called as null terminator or null parameter (10) so when you define string you should be sure to have suffivient space for the null terminator, the null terminator has value 0.

Declearing String:-

As in string definition we have two ways to declare a string. the way i.e we declare an array of characters all follows:-

```
char s[] = "string"
        or
char str[20];
```

String operations:-

1) To display word with the longest length
2) to determines the frequency of occurance of particular characters in a string.
3) to check whether the given string palindrome or not.
4) to display index of first apperance of the substring
5) To count the occurance of each word is given string.

Algorithm :-

step 1 : Start

step 2 : Define the required function for,

(1) longest word check
(2) Count occurance
(3) is pallindrome.
(4) substring
(5) frequency of words.

step 3 : Assign flags to given the same choice

step 4 : take string as input.

step 5 : perform the choosen option / operation, with the help of flags.

step 6 : Exit.

* Pseudo code for function —

(1) to display word with logest length.
- define a function
- check if current character is not end of current word.
- use max function to determine the maximum length if the end of word is found.
- Return the word which has maximum length.

(2) to determine the frequency of occurance of particular character in the string.
- Iterate the entire string for that particular charactr and counter, when the encounter, the particular character.

(3) to check whether the given string is pallindrome or not - check the reverse of the given string.

- If the given string & reverse string are equal then it is a pallindrome.

(4) to display the apperance of substring
- Iterate the string.
- when the substring is encountered return that index.

(5) to count the occurance of each word
- Break the given string into list of word.
- Run a loop & counter for all words.

Flowchart –

start

Input s (string)

ung = 0 , flag = 0

while
s[leng,q] != Num.

leng ++

i = 0, j = leng - 1

while (leng 12)
+1 . flag = 1

if s[i] = s(j)

flag = 0

i++, j--

if flag = = 0

print pallindrome

int
allindrome.

stop

pseudo code -

(1) longer word length

```
def longest word. length (str)
        i in range (0, len (str))
            current_len = 0
    if (str[i] != ' '):
            longest t = str [i]
            current_len = +1
    else:
        res = max (res, (curr_len). 10.
```

(2) count occurance :-

```
def occurance (str, char):
    cout = 0
    for i in str;
    if i == char
    Count = count + 1
    return char, Count
```

(3) Is pallindrome

```
def is pallindrome (str):
    for i in range (0, len (str)/2):
    if str (i) != str [len (str) - ]
        return false,
        return true.
```

(4) for substring

```
def substr (main str, sub)
    res ( )
    Flag = 0
    K = 0
```

```
for i in range (0, len
                (main str);
    K = 1
    flag = 0
if flag: break K = K+1
if Flag = = 0;
    res. append (:)
    return rg.
```

Application:

(1) to display word with the longest length.

step 1: Start
step 2: Break sentence into array a[]
step 3: initialize counter variable i = 0
step 4: iterate loop through each in array
step 5: Get length of each word update
step 6: Return the counts.
step 7: Exit.

(2) to determine the frequency of occurance particular character in the string.

step 1: Start.
step 2: S = String
step 3: Define a() (arry) frequency with some size of string.
step 4: Two loops will be used to count the frequency of a charactu.
step 5: If a match found in arrent element
step 6: Display the charactu & their occurance.
step 7: Exit.

(3) to check whether a given string is pallindrome or not.

step 1: Define a string str.
step 2: Define another string for reversed string str2.

Step 3: Initialize variable flag = 1

Step 4: If (str1 == str2)

print ("It is palindrome)

else,

print +("It is not");

step 6: flag = 1, print i, Exit.

Conclusion :-

By this way we can perform string operation successfully.