Ex.No.3	TEXT PROCESSING USING STRING IN PYTHON	Reg.No: URK24CS1189

3a. Write a python program that accepts a comma separated sequence of words as input and prints the unique words in sorted from (alphanumerically).

Sample Input: red, black, pink, green. Sample Output: black, green, pink, red.

Aim: To implement text processing using string in python.

## **Description / Algorithm:**

Step1: Accept the input from the user, which is a comma-separated sequence of words.

Step2: Split the input string into individual words using the comma delimiter.

Step3: Convert the list of words into a set to remove any duplicate words (if present)>

Step4: Sort the set of unique words alphanumerically.

Step5: Join the sorted words into a comma-separated string.

Step6: Print the result.

## **Program:**

```
#URK24CS1189
ipstring=input("enter a coma seperated sequence of words :")
list=ipstring.split(",")
uniquewords=set(list)
sortedwords=sorted(uniquewords)
op=",".join(sortedwords)
print("unique sorted list :",op)
```

#### **Output:**

```
enter a coma seperated sequence of words :red,black,pink,green
unique sorted list : black,green,pink,red
=== Code Execution Successful ===
```

Result: Thus, a python program that accepts a comma separated sequence of words as input and prints the unique words in sorted from (alphanumerically) has been done successfully.

3b. Write a program to find out if the string contains: alphanumeric characters, alphabetical characters, digits, lowercase and uppercase characters.

Aim: To implement text processing using string in python.

**Description / Algorithm:** 

Step1: Read the string s from the user.

Step2: Initialize a variable l as the length of the string.

Step3: For each of the following checks, initialize a flag variable to False, and then loop through each character in the string:

- Check if there are any alphanumeric characters: If any character is alphanumeric (isalnum()), set the flag to True.
- Check if there are any alphabetic characters: If any character is alphabetic (isalpha()), set the flag to True.
- Check if there are any digits: If any character is a digit (isdigit()), set the flag to True.
- Check if there are any lowercase characters: If any character is lowercase (islower()), set the flag to True.
- Check if there are any uppercase characters: If any character is uppercase (isupper()), set the flag to True.

Step4: For each check, print True if the flag was set to True, otherwise print False.

## **Program:**

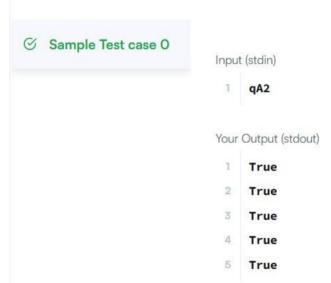
```
Change Theme
 1 \sif __name__ == '__main__':
 2
         s = input()
         l = len(s)
 3
 4
         flag = 0
 5 V
         for i in range(l):
 6 V
             if s[i].isalnum():
 7
                 flag+=1
         if flag == 0:
 8 V
             print("False")
9
10 V
         else:
             print("True")
         flag = 0
         for i in range(l):
14 V
15 V
             if s[i].isalpha():
                 flag+=1
17 V
         if flag == 0:
             print("False")
18
19 V
         else:
             print("True")
         flag = 0
         for i in range(l):
23 V
             if s[i].isdigit():
24 V
                flag+=1
26 V
         if flag == 0:
27
             print("False")
```

```
28 🗸
         else:
             print("True")
         flag = 0
         for i in range(l):
32 V
            if s[i].islower():
33 🗸
34
                 flag+=1
         if flag == 0:
35 ∨
             print("False")
37 ∨
         else:
            print("True")
40
         flag = 0
41 🗸
         for i in range(l):
42 V
            if s[i].isupper():
43
                 flag+=1
         if flag == 0:
44 🗸
            print("False")
45
46 ∨
         else:
             print("True")
```

# **Output:**

# Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.



Result: Thus, a python program to find out if the string contains: alphanumeric characters, alphabetical characters, digits, lowercase and uppercase characters has been done successfully.

```
3c. Write a python program to find characters count of a string which are passed as list.
```

```
I/p: St = "hello welcome" lst = [ "l", "w", "m", "e"]
```

O/p:

l 3 w 1

m 1

e 3

Aim: To implement text processing using string in python.

# **Description / Algorithm:**

**Step1: Initialize input:** 

- Set the string St = "hello welcome"
- Set the list lst = [ "l", "w", "m", "e"]

# **Step2: Iterate through list:**

- For each char in lst.
- Count the occurrences of char in St.
- Print char and it's count.

Step3: Repeat for all characters in the list.

#### **Program:**

```
1 #URK24CS1189
2 st="hello welcome"
3 lst=["l", "w", "m", "e"]
4 for i in lst:
5     count=st.count(i)
6     print(i,count)
```

# **Output:**



Result: Thus, a python program to find characters count of a string which are passed as list has Been done successfully.

3d. Write a python program to find the first appearance of the substring 'not' and 'bad' from a given string, if 'not' follows 'bad', replace the whole 'not'....'bad' substring with 'good'. Return the resulting string.

**Sample Input:** 

The song is not bad!

The song is poor!

**Sample Output:** 

The song is good!

The song is poor!

Aim: To implement text processing using string in python.

**Description / Algorithm:** 

Step1: Set the input string St.

Step2: Check if 'not' comes before 'bad'.

Step3: If true, replace 'not' to 'bad' with 'good'.

Step4: Return the modified string or the original string if no replacement is made.

# **Program:**

```
St=input("enter a string:")

if 'not' in St and 'bad' in St and St.find('not') <St.find('bad'):

St=St.replace('not that bad','good')

print(St)
```

#### **Output:**

```
enter a string:The song is not that bad!
The song is good!
PS C:\Users\URK24CS1188>
```

Result: Thus, a python program to find the first appearance of the substring 'not' and 'bad' from a given string, if 'not' follows 'bad', replace the whole 'not'....'bad' substring with 'good'. Return the resulting string has been done successfully.