Date

Page

Assingment3 Module 3.

1. Count Occurences of all characters within a String:

```
while True:
    print ("Enter 'x' for exist.")
    string = input ("Enter any string:")
    if string = = 'x':
        break
     else:
       char = input ("Enter a character to count:")
       val = string. count (char)
       print ( val, "\n").
```

2. Sum and Average of the digits that appear in a String:

```
number = int (input ("please Enter any Number: "))
total = 0
for value in range (1, number +1):
  total = total +Value
  average = total / Number.
print ("The sum of Natural Numbers from 1 to {0} = {1}", format
                                          (number, total))
print ("Average of Natural Numbers from 1 to {0}= {1}", format
                                          (number, Average)).
```

(OR)

Scanned with CamScanner

```
def sum_digits_string(str1):
    sum_digit=0
    for x in str1:
        if x.isdigit()== True:
            z=int(x)
            sum_digit=sum_digit+z
            return sum_digit
print (sum_digits_string("14ab37rb5"))
```

3. string input count all lower case, upper case, digits and special symbols.

```
def Count(str):
    upper, lower, number, special=0,0,0,0
     for i in range(len(str)):
         if str[i].isupper():
                upper +=1
         elif str[i].islower():
                lower +=1
         elif str[i].isdigit():
                number +=1
         else:
                special +=1
print ('Upper case letters:', upper)
print ('Lower case letters:', lower)
print ('Number:', number)
print ('special characters:', special)
  str = "$Geek5BOOk@593A"
  Count (str)
```

4. String Characters such that lowercase letters should come first.

```
inputStr = "pyNative"
words = inputStr. split()
lower= []
Upper= []
for char in inputStr:
  if char. islower():
    lower. append (char)
  else:
      Upper. append (char)
  Sorted String=' '.join (lower + Upper)
print ("In arranging characters given precedence to lowercase letters:")
print (Sortedstring)
```

5. Q. S1 and S2, create a new string by appearing S2 in the middle of S1.

```
def appendMiddle (S1, S2):
   middle Index = int( len(s1)/2)
   print (" Original Strings are", S1, S2)
  middleThree = S1 [:middleIndex - 1:] + S2 + S1[ Middle Index 1:]
  print (" After appending new string in middle", middle Three)

appendMiddle (" Apple ", "EatEverymorning")
```