

2nd part -Programming Questions:

16. Write a Program that reads a string from the user and displays the count of uppercase, lowercase letters, digits and spaces in the read string.

```
str=input("enter a string:")
uc=0
lc=0
d=0
s=0
for i in range(len(str)):
    if(str[i].isupper()):
        uc+=1
    if(str[i].islower()):
        lc+=1
    if(str[i].isdigit()):
        d+=1
    if(str[i].isspace()):
        s+=1
print("uppercase:" ,uc, "lowercase:" ,lc, "digit:" ,d, "space:" ,s)
```

```
enter a string: Welcome to Python webinar 123
uppercase: 2 lowercase: 20 digit: 3 space: 4
```

17. Write a Program that swaps the cases of the read string without using inbuilt method

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

enter a string: hello world

HELLO WORLD

```
str1=""
```

```
for i in range(len(str)):
```

```
    if(str[i].islower()):
```

```
        str1+=str[i].upper()
```

```
    else:
```

```
        str1+=str[i].lower()
```

```
print(str1)
```

18. Write a Program to check whether a given sentence is pangram or not [Example:The five boxing wizards jump quickly is a pangram]

```
import string
import sys

s = input('Enter the string:')
alphabets = list(string.ascii_lowercase)

#pangarm is a string containing every letter of the
alphabet

for ch in alphabets:
    if ch not in s.lower():
        print (s,'is not pangram string')
        sys.exit()

print (s, 'is pangram string')
```

Enter the string:The five boxing wizards jump quickly is a pangram
The five boxing wizards jump quickly is a pangram is pangram string

Enter the string:The five
The five is not pangram string

19. Write a Python program to check the validity of a password (input from user). Validation :

- ❑ At least 1 letter between [a-z] and 1 letter between [A-Z].
- ❑ At least 1 number between [0-9].
- ❑ At least 1 character from [\$#@].
- ❑ Minimum length 6 characters and Maximum length 16 characters.

```
l, u, p, d = 0, 0, 0, 0
```

```
s = input("enter a password:")
```

```
if (len(s) >= 6 and len(s) <= 16):
```

```
    for i in s:
```

```
        # counting lowercase alphabets
```

```
        if (i.islower()):
```

```
            l+=1
```

```
        # counting uppercase alphabets
```

```
        if (i.isupper()):
```

```
            u+=1
```

```
        # counting digits
```

```
        if (i.isdigit()):
```

```
            d+=1
```

```
        # counting the mentioned special characters
```

```
        if(i=='@' or i=='$' or i=='#'):
```

```
            p+=1
```

```
if (l>=1 and u>=1 and p>=1 and d>=1 and l+p+u+d==len(s)):
```

```
    print("Valid Password")
```

```
else:
```

```
    print("Invalid Password")
```

enter a password:Pooja@123

Valid Password

enter a password:Hello123

Invalid Password

20. Write a Python Program to Reverse words in a given String in Python. [Input : str = "python quiz practice code."
Output : str = "code. practice quiz python"]

```
# initializing the string  
string = input("enter a string:")
```

```
# splitting the string on space  
words = string.split( )
```

```
# reversing the words using reversed() function  
words = list(reversed(words))
```

```
# joining the words and printing  
print(' '.join(words))
```

```
enter a string : python quiz practice code.  
code. practice quiz python
```

21. Write a Python program to create a Caesar encryption. Note: In cryptography, a Caesar cipher, It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on.

```
realText=input("enter a text:")
step=int(input("enter a step value:"))
def caesar_encrypt(realText, step):
    outText = []
    cryptText = []

    uppercase = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
    lowercase = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

    for eachLetter in realText:
        if eachLetter in uppercase:
            index = uppercase.index(eachLetter)
            crypting = (index + step) % 26
            cryptText.append(crypting)
            newLetter = uppercase[crypting]
            outText.append(newLetter)
        elif eachLetter in lowercase:
            index = lowercase.index(eachLetter)
            crypting = (index + step) % 26
            cryptText.append(crypting)
            newLetter = lowercase[crypting]
            outText.append(newLetter)

    return outText

code = caesar_encrypt(realText, step)
print(code)
```

enter a text : abc
enter a step value:2
['c', 'd', 'e']

enter a text:ABC
enter a step value:3
['D', 'E', 'F']

enter a text:AB
enter a step value:3
['D', 'E']

22 Write a Python to display a number in left, right and center aligned of width 10

```
n=input("Enter the number:")  
print("Left aligned:")  
print(n.ljust(10))  
print("right aligned:")  
print(n.rjust(10))  
print("center aligned:")  
print(n.center(10))
```

```
Enter the number:30  
Left aligned:  
30  
right aligned:  
    30  
center aligned:  
   30
```

23. Write a program that simulates Password Manager Software

```
entry=1
manager={}
while(True):
    entry=int(input("Press 1 to store p/w 0 to exit:"))
    if(entry==0):
        break
    website=input("Website:")
    password=input("password:")
    manager[website]=password
```

```
print(manager)
```

Output

```
Press 1 to store p/w 0 to exit:1
Website:abc
password:123
Press 1 to store p/w 0 to exit:1
Website:def
password:passwd
Press 1 to store p/w 0 to exit:0
{'abc': '123', 'def': 'passwd'}
```


24. Write a program to display the presence of given substring in main string

```
string=input("Enter the string:")
sub=input("Enter the substring:")
if(string.find(sub) ==-1):
    print("Substring not present")
else:
    print("Substring is present.")
```

Enter the string:royal oak
Enter the substring:yal
Substring is present.

Enter the string:royal oak
Enter the substring:ok
Substring not present

25. Write a python program to read a sentence from the user and display the longest word of that sentence along with its length.

```
def longest_word(sentence):  
    ls=sentence.split()  
    longest= max(ls, key=len)  
    return longest,len(longest)
```

```
sentence=input("enter the sentence:")  
res1,res2=longest_word(sentence)  
print("the longest word is", res1, "with length", res2)
```

enter the sentence: welcome to the world of python
the longest word is welcome with length 7

26. Program to get a string from a given string where all occurrences of its first character have been changed to “\$” except the first character (Example : restart to be changed as resta\$t)

```
s=input("Enter the string:")  
s1=s.replace(s[0],'$')  
print(s[0]+s1[1:])
```

Enter the string:papaya
pa\$aaya

27. Program to add “ing” at the end of the given string, if it ends with “ing” then add “ly”. If the string length of the given string is less than 3 leave it unchanged

```
def string_end (str):  
    length=len (str)  
    if length > 2:  
        if str.endswith('ing')==True:  
            str+='ly'  
        else:  
            str+='ing'  
  
    return str
```

```
word=input("Enter the word:")  
print(string_end(word))
```

Enter the word:call
calling

Enter the word:disturbing
disturbingly

Enter the word:am
am

28. Write a Program to count the number of vowels in a given string

```
str=input("enter a string:")  
s="AaEeliOoUu"  
count=0  
for ch in str:  
    if(ch in s):  
        count+=1  
  
print(count)
```

```
enter a string:pooja  
3
```

29. Write a Program that creates a function which checks whether the given string is palindrome or not

```
def palindrome(word):  
    reverse=word[::-1]  
    print(reverse)  
    if word == reverse:  
        print("Palindrome")  
    else:  
        print("Not a Palindrome")  
word=input("Enter a word: ")  
palindrome (word)
```

output

Enter a word: python
nohtyp
Not a Palindrome

Enter a word: abccba
abccba
Palindrome

30. Write a function that takes 2 arguments: string and a character, which counts the number of occurrences of character in the string

```
def count_char(string,ch):  
    count=0  
    for i in string:  
        if(ch==i):  
            count=count+1  
        else:  
            continue  
    return count
```

```
string=input ("Enter a string: ")  
ch=input ("Enter a character: ")
```

```
res=count_char(string,ch)
```

```
print("Number of times", ch, "is present in ", string, "=",res)
```

Enter a string: hello world

Enter a character: l

Number of times l is present in hello world = 3

31. Write a function to check whether two strings are anagram or not [LISTEN and SILENT are anagrams]

```
str1=input("enter the 1st string:")  
str2=input("enter the 2nd string:")
```

```
def areAnagram(str1, str2):
```

```
    n1=len(str1)
```

```
    n2=len(str2)
```

```
    if (n1!=n2):
```

```
        return 0
```

```
    str1= sorted(str1)
```

```
    str2= sorted (str2)
```

```
    if (str1==str2):
```

```
        return 1
```

```
    else:
```

```
        return 0
```

```
if areAnagram(str1, str2):
```

```
    print("The two strings are anagrams of each other")
```

```
else:
```

```
    print ("The two strings are not anagrams of each other")
```

enter the 1st string:listen

enter the 2nd string:silent

The two strings are anagrams of each other

enter the 1st string:camera

enter the 2nd string:camel

The two strings are not anagrams of each other

32. Write a Program to remove all duplicate characters in a string and prints the string with unique characters

```
original_string=input("Enter a string: ")
```

```
unique_str=""
```

```
for i in original_string:
```

```
    if(i not in unique_str):
```

```
        unique_str+=i
```

```
    else:
```

```
        continue
```

```
print("The unique string is:" ,unique_str)
```

Enter a string: hello everyone

The unique string is: helo vryn

33. Consider the string „brontosaurus“. Write Pythonic code that implements and returns the functionality of histogram using dictionaries for the given string.

```
import pprint
string='brontosaurus'
freq={}
for ch in string:
    if ch not in freq:
        freq[ch]=1
    else:
        freq[ch]+=1

pprint.pprint(freq)
```

```
{'a': 1, 'b': 1, 'n': 1, 'o': 2, 'r': 2, 's': 2, 't': 1, 'u': 2}
```

34. Write a Program that has a dictionary of names of students and a list of marks in 4 subjects. Create another dictionary from this dictionary that has the name of the students and their total marks. Find out the topper and his/her score

```
student1={}    #will store names and their marks in 4 subjects
student2={}    #will store names and total marks
n=int(input("Enter the no of students:"))
print("Enter their names and marks scored in 4 subjects:")
total=[0]*n    # will store total marks in 4 subject
for i in range(n):
    name=input() #name
    marks=[0]*4  #list storing marks in 4 subjects
    for j in range(4):
        marks[j]=int(input())
        total[i]+=marks[j]
    student1[name]=marks

student2=dict.fromkeys(student1.keys(),0) #will store the keys in student1 to student2

i=0
for keys in student2:
    student2[keys]=total[i]
    i+=1

max_mark=0
for key in student2:
    if(student2[key]>max_mark):
        max_mark=student2[key]
        topper=key

print(topper," is the topper with ",max_mark,"Marks.")
print("Student_1=",student1)
print("Student_2=",student2)
```

Enter the no of students:4

Enter their names and marks scored in 4 subjects:

jyo

4

3

7

9

poo

4

3

5

6

srima

10

9

6

8

vaish

10

9

9

8

vaish is the topper with 36 Marks.

Student_1= {'jyo': [4, 3, 7, 9], 'poo': [4, 3, 5, 6], 'srima': [10, 9, 6, 8],
'vaish': [10, 9, 9, 8]}

Student_2= {'jyo': 23, 'poo': 18, 'srima': 33, 'vaish': 36}

35. Write a Python program to get the maximum and minimum value in a dictionary.

```
d={"Apple":10,"Cherry":50,"Kiwi":2,"Orange":25}
d_sorted=sorted(d.items(),key=lambda x:x[1])
print(d_sorted)
print(d_sorted[0])
print(d_sorted[len(d)-1])
```

```
[('Kiwi', 2), ('Apple', 10), ('Orange', 25), ('Cherry', 50)]
('Kiwi', 2)
('Cherry', 50)
```

36. Write a Python program to get the top three items in a shop.

Sample data: {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}

Expected Output:

item4 55
item1 45.5
item3 41.3

```
shop={'item1':45.50,"item2":35,"item3":41.30,"item4":55,"item5":24}
sorted_shop=sorted(shop.items(),key=lambda
x:x[1],reverse=True)
print(sorted_shop)
for i in range(3):
    print("top 3 items in shop:",sorted_shop[i])
```

```
[('item4', 55), ('item1', 45.5), ('item3', 41.3), ('item2', 35),
('item5', 24)]
top 3 items in shop: ('item4', 55)
top 3 items in shop: ('item1', 45.5)
top 3 items in shop: ('item3', 41.3)
```

37. Write a python program to print the frequency of characters in a string. (Hit: Use a dictionary with the character as keys and the counter as values.)

```
s=input("Enter a string:")
```

```
d=dict()
```

```
for ch in s:
```

```
    d[ch]=d.get(ch,0)+1
```

```
print(d)
```

Enter a string:hello

{'h': 1, 'e': 1, 'l': 2, 'o': 1}

38. Write a program to store data about your friends' birthdays.(Hit: Use a dictionary with the names as keys and the birthdays as values.)

```
birthdays = {'Alice': 'Apr 1', 'Bob': 'Dec 12', 'Carol': 'Mar 4'}  
while True:  
    print('Enter a name: (blank to quit)')  
    name = input()  
    if name == '':  
        break  
    if name in birthdays:  
        print(birthdays[name] , ' is the birthday of ' , name)  
    else:  
        print('I do not have birthday information for ' , name)  
        print('What is their birthday?')  
        bday = input()  
        birthdays[name] = bday #add the information to the dictionary  
        print('Birthday database updated.')  
print(birthdays)
```

Enter a name: (blank to quit)

Alice

Apr 1 is the birthday of Alice

Enter a name: (blank to quit)

Mamatha

I do not have birthday information for Mamatha

What is their birthday?

Mar 8

Birthday database updated.

{'Alice': 'Apr 1', 'Bob': 'Dec 12', 'Carol': 'Mar 4',
'Mamatha': 'Mar 8'}

Enter a name: (blank to quit)

11:33 am

39. Consider a dictionary with strings as keys and numbers as values. Write a program to sort the elements of this dictionary based on keys.

```
import pprint  
d={"Tom":3456,"Jerry":464467,"Mickey":5657}  
pprint.pprint(d)
```

Output*

```
{'Jerry': 464467, 'Mickey': 5657, 'Tom': 3456}.
```


40. Read a string from keyboard input. Create a list containing tuples, where each tuple represents a word in the input string and length of that string. Write a program sort the words in descending order of their length

```
str=input("Enter the string:").split()
final=[]
for word in str:
    t=(word,len(word))
    final.append(t)
print(final)
```