

Assignment-3

Ques1- Write a Python program to count the number of occurrences of each word or character in the string entered by the user. (Count the Number of Occurrences of each character only if the single word is entered by the user).

Ans1-

Code:

```
print("\nQuestion 1")

#input from user
string_name=str(input("Enter the string:"))
list1=string_name.split()
list_l=[]
for e in list1:
    lower_e=e.lower()
    list_l.append(lower_e)
set1=set(list_l)
dic={}
for el in set1:
    count=list_l.count(el)
    dic.update({el:count})
dic2={}
set2={1,2}
set2.clear()
list2=[]
if len(list1)==1:

    for elements in string_name:
        list2.append(elements)

    for el in list2:
        set2.add(el.lower())

    string_l=string_name.lower()
    for elem in set2:

        dic2.update({elem:string_l.count(elem)})

    print("\nDictionary containing 'Letter':'Letter Count' is shown below:")
    print(dic2)
else:
    print("\nDictionary containing {'Word':'Word Count'} is shown below:")
    print(dic)
```

Result:

Question 1

Enter the string:A quick brown fox jumps over the lazy dog

Dictionary containing {'Word':'Word Count'} is shown below:

{'fox': 1, 'dog': 1, 'quick': 1, 'a': 1, 'the': 1, 'jumps': 1, 'brown': 1, 'over': 1, 'lazy': 1}

Question 1

Enter the string:Alternative

Dictionary containing 'Letter':'Letter Count' is shown below:

{'a': 2, 'v': 1, 'r': 1, 'i': 1, 'l': 1, 'e': 2, 't': 2, 'n': 1}

Ques2- Write a python script to print next date of input date. It must meet following conditions(use if-elif)

C1:1<=month<=12

C2:1<=day<=31

C3:1800<=year<=2025

E.g.:

Input: Day - 28

Month -02

Year -1999

Output: Next Date is: 1/03/1999

Ans2-

Code:

```
print("\nQuestion 2")

print("Enter Date below to get date of next day.\n")
# introducing leap year function
def leapyear(x):
    # leap year condition
    if (x%400)==0 or ((x%100!=0) and (x%4==0)):
        return True
    else:
        return False
#input from user
day=int(input("Enter Day [1-31]:"))
month=int(input("Enter Month [1-12]:"))
year=int(input("Enter Year [1800-2025]:"))
#condition 1
if day<1 or day>31 or month<1 or month>12 or year<1800 or year>2025:
    condition1=False
else:
    condition1=True

#condition 2
month_31 = [1, 3, 5, 7, 8, 10, 12] #List containing month with 31 days
month_30 = [2, 4, 6, 9, 11] #List containing month with 30 days
#for day entered 31 does not present in that month
cla= day==31 and (month in month_30)
#for day entered greater than 29 in month february i.e 2
clb= day>29 and month==2
#for day entered greater than 28 in february in non leapyear
clc= day==29 and (not leapyear(year)) and month==2
if cla or clb or clc :
    condition2=False
else:
    condition2=True
#printing error when date entered is not correct
if cla:
    print(f"\nError\n{day} day can't be in month {month}")
elif clb:
    print(f"\nError\n{day} day can't be in month {month}")
elif clc:
    print(f"\nError\n{day} day can't be in month {month} as the year {year} is not leapyear")
elif condition1==False:
    print(f"\nError\nPlease enter date in range day[1-31], month[1-12], year[1800-2025] ")

#correct data
if condition1==True and condition2==True :
    month_31 = [1, 3, 5, 7, 8, 10, 12] #List containing month with 31 days
    month_30 = [2, 4, 6, 9, 11] #List containing month with 30 days
    #For month with 31 days
```

```

if (month in month_31) == True:
    if day == 31:
        day = 1
        month = month + 1
    elif 1 <= day <= 30:
        day = day + 1
#For month with 30 days
elif (month in month_30) == True:
    if day == 30 and month == 12:
        day = 1
        month = 1
        year = year + 1
    elif day == 30 and month != 12:
        day = 1
        month = month + 1
    elif 1 <= day <= 29:
        day = day + 1
#for february month i.e. month 2
elif month == 2:
    # If year is leap year
    if leapyear(year) == True:
        if day == 29:
            day = 1
            month = month + 1
        elif 1 <= day <= 28:
            day = day + 1
    # If year is not leap year
    elif leapyear(year) == False:
        if day == 28:
            day = 1
            month = month + 1
        elif 1 <= day <= 27:
            day = day + 1
# Printing Next date
print(f"\nNext Date in format Day/Month/Year is: {day}/{month}/{year}")

```

Result:

Question 2

Enter Date below to get date of next day.

Enter Day [1-31]:3

Enter Month [1-12]:8

Enter Year [1800-2025]:2023

Next Date in format Day/Month/Year is: 4/8/2023

|

Ques3- Write a Python program to create a list of tuples with the first element as the number and Second element as the square of the number.

E.g. list = [3, 9, 10]

Output: [(3, 9), (9, 81), (10, 100)]

Ans3-

Code:

```
print("\nQuestion 3")

#input list
list1=list(map(int,input("Enter the numbers separated by space:").split()))
#blank list
list2=[]
for e in list1:
    t=(e,e*e)
    list2.append(t)
# Printing the final result
print("\nList containing (n,n^2) is shown below:")
print(list2)
|
```

Result:

Question 3

Enter the numbers separated by space:2 4

List containing (n,n^2) is shown below:

[(2, 4), (4, 16)]

Ques4- Write a program to prompt the user for a grade between 4 and 10. If the grade is out of range print error message. If the grade is between 4 and 10 Print the grade and the performance using the following:

Letter Grade	Performance	Grade Points
A+	Outstanding	10
A	Excellent	9
B+	Very Good	8
B	Good	7
C+	Average	6
C	Below Average	5
D	Poor	4

E.g.: For Grade 9 Output is:

Your Grade is 'A' and Excellent Performance.

Ans4-

Code:

```
print("\nQuestion 4")

grade_point=int(input("Enter grade points:"))
dic={10:"Your grade is 'A+' and Outstanding performance.",
      9:"Your grade is 'A' and Excellent performance.",
      8:"Your grade is 'B+' and Very Good performance.",
      7:"Your grade is 'B' and Good performance.",
      6:"Your grade is 'C+' and Average performance.",
      5:"Your grade is 'C' and Below Average performance.",
      4:"Your grade is 'D' and Poor performance."}
#Applying Conditions for correct grade range
if 4<=grade_point<=10:
    statement=dic.get(grade_point)
    print(statement)
else:
    print("\nError\nPlease enter grade in range [4,10]")
```

Result:

```
Question 4
Enter grade points:8
Your grade is 'B+' and Very Good performance.
|
```

Ques5- Write a python program to print following pattern.

ABCDEFGHJK

ABCDEFGHI

ABCDEFG

ABCDE

ABC

A

Ans5-

Code:

```
print("\nQuestion 5")

string = "ABCDEFGHJK"
j = 0
while len(string)-j*2 >= 1:
    print(" "*j, string[0 : len(string) - j*2])
    j += 1
|
```

Result:

```
Question 5
ABCDEFGHIJK
ABCDEFGHI
ABCDEFG
ABCDE
ABC
A
```

Ques6- Write a python script that repeatedly ask user to enter name and SID of students (use 'Y' or 'N'). Store them in a dictionary whose keys are SID's and values are student's name. Perform the following operations on Dictionary :

- a. Print students details stored in the dictionary.**
- b. Sort dictionary using student names.**
- c. Sort dictionary using SID.**
- d. Search a student details with SID and print name of that student.**

Ans6-

Code:

```

print("\nQuestion 6")

repeat="Y"
dic={}
dic2={}
#List containing Y or N
liste=["Y","y","n","N"]
while repeat=="Y" or repeat=="y":

    name = str(input("Enter student name:"))
    sid = int(input("Enter student SID:"))
    if sid<0:
        print("\nError\nSID can't be negative\n")
    else:

        dic.update({sid: name})

        dic2.update({name:sid})

        repeat = str(input("Enter Y to continue or N to end:"))
    if repeat=="N" or repeat=="n":
        break
    elif (not (repeat in liste)):
        print("\nError\nPlease enter valid input['Y' or 'N']")
        repeat=str(input("\nEnter Y to continue or N to end:"))

# a
#printing the dictionary
print("\nQ.6(a)")
print("The Dictionary containing {'SID':'Name'} is shown below")
print(dic)
# b
#sorting according to name
print("\nQ.6(b)")
list_k=sorted(dic2)
dic3={}
for ele in list_k:
    dic3.update({dic2.get(ele):ele})
print("The Dictionary after sorting according to name:")
print(dic3)

# c
#sorting according to SID
print("\nQ.6(c)")
sort_dic = sorted(dic)
dic4 = {}
for va in sort_dic:
    dic4.update({va: dic.get(va)})

print("The Dictionary after sorting according to SID:")
print(dic4)
# d
print("\nQ.6(d)")
# Taking input SID to be searched
enter_sid = int(input("Enter SID to get name of student:"))
# Searching for sid as key in dic
output_name = str(dic.get(enter_sid))
# printing name with key sid
print(f"Name of student with SID {enter_sid} is {output_name}.")

```

Result:

Question 6

```
Enter student name:Rahul
Enter student SID:22103020
Enter Y to continue or N to end:Y
Enter student name:Mohan
Enter student SID:22103021
Enter Y to continue or N to end:Y
Enter student name:Sonu
Enter student SID:22103050
Enter Y to continue or N to end:N
```

Q.6(a)

The Dictionary containing {'SID':'Name'} is shown below
{22103020: 'Rahul', 22103021: 'Mohan', 22103050: 'Sonu'}

Q.6(b)

The Dictionary after sorting according to name:
{22103021: 'Mohan', 22103020: 'Rahul', 22103050: 'Sonu'}

Q.6(c)

The Dictionary after sorting according to SID:
{22103020: 'Rahul', 22103021: 'Mohan', 22103050: 'Sonu'}

Q.6(d)

```
Enter SID to get name of student:22103050
Name of student with SID 22103050 is Sonu.
```

Ques7- Write a python program to print Fibonacci sequence also print average of the resultant Fibonacci series.

Ans7-

Code:


```

print("\nQuestion 7")

#input from user
n=int(input("Enter number of elements N in fibonacci series:\n[N must be positive Integer]: N="))
#printing error message when N<=0
if n<=0 :
    print("\nError\nNumber of elements in fibonacci series must be integer and greater than zero.")
#code for fibonacci series
else:
    #code for fibonacci series for first 2 elements
    if n == 1:
        print("\nThe fibonacci series with 1 element is shown below\n[1]")
        print("\nAverage of given fibonacci series is", 1)

    elif n == 2:
        print("\nThe fibonacci series with 2 element is shown below\n[1,1]")
        print("\nAverage of given fibonacci series is", 1)
    #General code for fibonacci for next N-2 elements
    else:
        # List of fibonacci elements with 1,1 as initial elements
        list1 = [1, 1]
        #Building logic to get fibonacci series
        a = 1
        b = 1
        # For loop
        for i in range(n - 2):
            s = a + b
            list1.append(s)
            a = b
            b = s
        # printing the final fibonacci series
        print(f"\nThe fibonacci series with {n} elements is shown below:")
        print(list1)
        # Finding average of fibonacci series
        sum = 0 #intial sum=0
        # finding sum of all elements in list1
        for num in list1:
            sum = sum + num
        average = (sum / n)
        # Till two decimal place
        two_decimal = "{:.2f}".format(average)
        # printing average
        print(f"\nAverage of given fibonacci series is {two_decimal}")

```

Result:

```

Question 7
Enter number of elements N in fibonacci series:
[N must be positive Integer]: N=24

The fibonacci series with 24 elements is shown below:
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181,
, 6765, 10946, 17711, 28657, 46368]

Average of given fibonacci series is 5058.00

```

Ques8- Given the sets below, write python statement to:

Set1= {1, 2, 3, 4, 5}

Set2= {2, 4, 6, 8}

Set3= {1, 5, 9, 13, 17}

a. Create a new set of all elements that are in Set1 and Set2 but not both.

b. Create a new set of all elements that are in only one of the three sets Set1, Set2 and Set3.

c. Create a new set of elements that are exactly two of the sets Set1, Set2 and Set3.

d. Create a new set of all integers in the range 1 to 10 that are not in Set1.

e. Create a new set of all integers in the range 1 to 10 that are not in Set1, Set2 and Set3.

Ans8-

Code:

```
print("\nQuestion 8")

#Given Sets
set1= {1, 2, 3, 4, 5}
set2= {2, 4, 6, 8}
set3= {1, 5, 9, 13, 17}
#printing the given sets
print(f"Set1= {set1}\nSet2= {set2}\nSet3= {set3}")
#a
print("\nQ.8(a)")
print("\nA new Set of all 'elements that are in Set1 and Set2 but not both' is:")
set_sym_dif=set1.symmetric_difference(set2)
print(set_sym_dif)
#b
#set1 Union set2
print("\nQ.8(b)")
print("\nA new set of all elements that are 'in only one of the three sets Set1,Set2 and Set3' is:")
set_ul=set1.union(set2)

#set1 Union set2 Union set3
set_uf=set_ul.union(set3)

#set1 intersection set2
set_il=set1.intersection(set2)

#set1 intersection set2 intersection set3
set_if=set_il.intersection(set3)

#set1 intersection set2
set_l2=set1.intersection(set2)

#set2 intersection set3
set_23=set2.intersection(set3)

#set3 intersection set1
set_l3=set1.intersection(set3)

#final required set
set_fl=set_uf-set_l2-set_23-set_l3
print(set_fl)
#c
print("\nQ.8(c)")
set_c1=set_l2.union(set_23)
set_c2=set_c1.union(set_l3)
set_final=set_c2-set_if
print("\nA new set of elements that are 'exactly in two of the sets Set1, Set2 and Set3' is:")
print(set_final)
#d

#forming a set containing values from 1 to 10
print("\nQ.8(d)")
set_d={1,2}
set_d.clear()
for i in range(1,11):
    set_d.add(i)
set_new=set_d-set1
#printing final set
print("\nA new set of all Integers in the 'range 1 to 10' that are 'not in Set1' is:")
print(set_new)
#e
print("\nQ.8(e)")
set_e=set_d-set_uf
print("\nA new set of all Integers in the range 1 to 10 that are not in Set1,Set2 and Set3.")
print(set_e)
```

Result:

Question 8
Set1= {1, 2, 3, 4, 5}
Set2= {8, 2, 4, 6}
Set3= {1, 17, 5, 9, 13}

Q.8(a)

A new Set of all 'elements that are in Set1 and Set2 but not both' is:
{1, 3, 5, 6, 8}

Q.8(b)

A new set of all elements that are 'in only one of the three sets Set1, Set2 and Set3' is:
{3, 6, 8, 9, 13, 17}

Q.8(c)

A new set of elements that are 'exactly in two of the sets Set1, Set2 and Set3' is:
{1, 2, 4, 5}

Q.8(d)

A new set of all Integers in the 'range 1 to 10' that are 'not in Set1' is:
{6, 7, 8, 9, 10}

Q.8(e)

A new set of all Integers in the range 1 to 10 that are not in Set1, Set2 and Set3.
{10, 7}

|