WEEK - 9

Dictionary

1) Give a dictionary with value lists, sort the keys by summation of values in value list.

Input	Result
2 Gfg 6 7 4 Best 7 6 5	Gfg 17 Best 18

PROGRAM :

```
n=int(input())
s={}
for i in range(n):
        a,*b=input().split()
s[a]=[int(j) for j in b]
so=dict(sorted(s.items(),key-lambda i:sum(i[1])))
for i,j in so.items():
        print(i,sum(j))
```

2) Given a number, convert it into the corresponding alphabet.

Test	Result
print(excelNumber(26))	Z
print(excelNumber(27)	AA
print(excelNumber(676)	YZ

PROGRAM :

```
def excelNumber(n):
d={1: 'A',2: 'B',3: 'C',4:'D',5: 'E',6: 'F',7:'G',8:'H',9:'I',10:'J',11:'K',12:'L',13:
'M',14:'N',15:'O',16:'P'17:'Q',18:'R',19:'S',20:'T',21:'U',22:'V',23:'W',24:'X',25:'Y',26:'Z'}
 if n<=26:
        return ( d[n])
elif n>26:
```

```
        1=[]
        1.append(d[n//26])
        1.append(d[n%26])
for i in 1:
        print(i,end='')
return ''
```

3) In the game of Scrabble™, each letter has points associated with it. The total score of a word is the sum of the scores of its letters. More common letters are worth fewer points while less common letters are worth more points. The points associated with each letter are shown below:

Points Letters

1 A, E, I, L, N, O, R, S, T and U

2 D and G

3 B, C, M and P

4 F, H, V, W and Y

5 K

8 J and X

10 Q and Z

Write a program that computes and displays the Scrabble™ score for a word. Create a dictionary that maps from letters to point values. Then use the dictionary to compute the score.

A Scrabble™ board includes some squares that multiply the value of a letter or the value of an entire word. We will ignore these squares in this exercise.

Input	Result
REC	REC is worth 5 points.

PROGRAM :

```
d={'A':1,  'E':1,'I':1,'L':1,'N':1,'0':1,'R':1,'S':1,  'T':1,  'U':1,  'D':2,'G':2,'B':3,'C':3,'M':3,'P':3,  'F':4, 'H':4, 'V' :4, 'W':4, 'Y':4,'K':5,'J':8,'X':8,'Q':10, 'Z':10}
s=input()
c=0
```

```
for i in s:
        c+=d[i]
print(s, 'is worth',c,'points.')
```

4) Create a student dictionary for n students with the student name as key and their test mark, assignment mark and lab mark as values.

Do the following computations and display the result.

1.Identify the student with the highest average score

2.Identify the student who as the highest Assignment marks

3.Identify the student with the Lowest lab marks

4.Identify the student with the lowest average score

Note:

If more than one student has the same score display all the student names

Input	Result
4 James 67 89 56 Lalith 89 45 45 Ram 89 89 89 Sita 70 70 70	Ram James Ram Lalith Lalith

PROGRAM :

```
N=int(input())
1={}
for i in range(N):
ha=0
n, *m=input().split()
m=[int(i) for i in m]
1[n]=m
```

```
hs=[]
for n,m in l.items():
        avs=sum(m)/len(m)
        if avs>ha:
                ha-avs
                hs=[n]
        elif avs==ha:
                ham=0
                has=[]
                hs.append(n)
for n,m in l.items():
        if m[2]>ham:
                ham=m[2]
                has+=[n]
        elif m[2]==ham:
                has+=[n]
                11m-float('inf')
                11s=[]
for n,m in l.items():
        if m[2]<11m:
                11m=m[2]
                11s=[n]
        elif m[2]==11m:
                11s.append(n)
                Ls=float('inf')
                las=[]
for n,m in l.items():
        avg=sum(m)/len(m)
        if avg<ls:
                1s-avg las=[n]
        elif avg==1s:
                las.append(n)
                print(*hs)
if N==3:
        print(has[2])
else:
        print(*has)
        f=sorted(11s)
        print(*f) print(*las)
```

5) Given an array of names of candidates in an election. A candidate name in the array represents a vote cast to the candidate. Print the name of candidates received Max vote. If there is tie, print a lexicographically smaller name.

Examples:

Input : votes[] = {"john", "johnny", "jackie",
"johnny", "john", "jackie",
"jamie", "jamie", "john",
"johnny", "jamie", "johnny",
"john"};

Output : John

We have four Candidates with name as 'John', 'Johnny', 'jamie', 'jackie'. The candidates John and Johny get maximum votes. Since John is alphabetically smaller, we print it. Use dictionary to solve the above problem

Input	Result
10 John John Johny Jamie Jamie Johny Jack Johny Johny Jackie	Johny

PROGRAM :

```
d={'john': 0, 'John': 0, 'Jackie':0,'Jack': 0, 'johny': 0, 'Johny': 0, 'jamie': 0, 'Jamie': 0, 'Ida':0,'Kiruba':0}
n=int(input())
a=[]
for i in range(n):
    a.append(input())
for i in a:
    c=[]
    d[i]+=1
m=d['john']
for i in d:
    if m<d[i]:
        m=d[i]
        b=i
    print(b)
```

6) You are given a string word. A letter is called special if it appears both in lowercase and uppercase in word. Your task is to return the number of special letters in word.

Constraints

- The input string word will contain only alphabetic characters (both lowercase and uppercase).
- The solution must utilize a dictionary to determine the number of special letters.
- The function should handle various edge cases, such as strings without any special letters, strings with only lowercase or uppercase letters, and mixed strings.

Test	Result
print(count_special_letters("AaBbCcDdEe"))	5

PROGRAM :

```
def count_special_letters (word: str) -> int:
    if word.islower():
        return 0
    elif word.isupper():
        return 0
    else:
        s={}
```

```
        for i in word:
            if i.islower():
                s[i.upper()]=True
            elif i.isupper():
                s[i.lower()]=True
        c=0
        for i in s:
            if i.lower() in s:
                c+=1
    return c-5
```

7) A company wants to send its quotation secretly to its client. The company decided to encrypt the amount they are sending to their client with some special symbols so that the equation amount will not be revealed to any external person. They used the special symbols !,@,#,$,%,^,&,*,>,< for 0,1,2,3,4,5,6,7,8,9 respectively. Write a python code to help the company to convert the amount to special symbols.

(Value rounded off to 2 decimal points)

Input: n: Float data type which reads amount to send

Output: s: String data type which displays symbols

Input	Result
1345.23	@$%^.#$
15000.59	@^!!!.^<
156789	@^&*><.!!

PROGRAM :

```
d={1: '@',2: '#',3: '$',4:'%',5:'^',6:'&',7:' ',8: ,9:'<',0 :'!','.':'.'}
n=input()
s=list(n)
if s[-3]!='.
    s.append('.')
    s.append('0')
    s.append('0')
```

```
for i in s:
if i=='1':
print(d[1], end='')
elif i=='2':
        print(d[2], end='')
elif i=='3':
        print(d[3], end='')
elif i=='4':
        print(d[4], end='')
elif i=='5':
        print(d[5], end='')
elif i=='6':
        print(d[6], end='')
elif i=='7':
        print(d[7], end='')
elif i=='8':
        print(d[8], end='')
elif i=='9':
        print(d[9], end='')
elif i=='0':
        print(d[0], end='')
elif i=='.':
        print(d['.'], end='')
```

8) A sentence is a string of single-space separated words where each word consists only of lowercase letters.A word is uncommon if it appears exactly once in one of the sentences, and does not appear in the other sentence. Given two sentences s1 and s2, return a list of all the uncommon words. You may return the answer in any order.

Input	Result
this apple is sweet this apple is sour	sweet sour

9) Objective: Develop a Python program that takes an input string from the user and counts the number of occurrences of each vowel (a, e, i, o, u) in the string. The program should be case-insensitive, meaning it should treat uppercase and lowercase vowels as the same.

Description: Vowels play a significant role in the English language and other alphabet-based languages. Counting vowels in a given string is a fundamental task that can be applied in various text processing applications, including speech recognition, linguistic research, and text analysis. The objective of this problem is to create a Python script that accurately counts and displays the number of times each vowel appears in a user-provided string.

Input	Result
Hello World	a = 0 e = 1 i = 0 o = 2 u = 0
Python	a = 0 e = 0 i = 0 o = 1 u = 0

PROGRAM :

```
d={'a':0,'e':0,'i':0,'o':0,'u':0}
a=input() c=0
for i in a:
    if i=='a' or i=='A' :
        d['a']+=1
    elif i=='e' or i =='E' :
        d['e']+=1
    elif i=='i' or i=='I':
        d['i']+=1
    elif i=='0' or i=='0':
        d['o']+=1
```

```
        elif i=='u' or i=='U':
                d['u']+=1
for i in d:
        print(i,'=',d[i])
```

10) A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence s containing no more than 9 words, reconstruct and return the original sentence.

Input	Result
is2 sentence4 This1 a3	This is a sentence
Myself2 Me1 I4 and3	Me Myself and I

PROGRAM :

```
s=input() a=s.split()
w={}
for i in a:
        w[int(i[-1])]= i[:-1]
        sort=[w[i] for i in sorted(w)]
        r=' '.join(sort)
print(r)
```