

## **09 – Dictionary**

vEx. No. : 9.1

Date: 4/5/24

Register No.: 231501053

Name: Gokulakkannan P

---

## Uncommon words

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.

### PROGRAM

```
s1 = input()
```

```
s2 = input()
```

```
words1 = s1.split()
```

```
words2 = s2.split()
```

```
count = {}
```

```
for word in words1 + words2:
```

```
    if word in count:
```

```
        count[word] += 1
```

```
    else:
```

```
        count[word] = 1
```

```
uncommon_words = [word for word in count if count[word] == 1]
```

```
print(uncommon_words)
```

## Output:

	Input	Expected	Got	
✓	this apple is sweet this apple is sour	sweet sour	sweet sour	✓
✓	apple apple banana	banana	banana	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**Ex. No. : 9.2**

**Date:** 4/5/24

**Register No.:** 231501053

**Name:** Gokulakkannan P

---

## **Sort Dictionary by Values Summation**

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

**Input :** test\_dict = {'Gfg' : [6, 7, 4], 'best' : [7, 6, 5]}

**Output :** {'Gfg': 17, 'best': 18}

**Explanation :** Sorted by sum, and replaced.

**Input :** test\_dict = {'Gfg' : [8,8], 'best' : [5,5]}

**Output :** {'best': 10, 'Gfg': 16}

**Explanation :** Sorted by sum, and replaced.

Sample Input:

2

Gfg 6 7 4

Best 7 6 5

Sample Output

Gfg 17

Best 18

**For example:**

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

### **PROGRAM**

```
n=int(input())
l1=[]
l2=[]
l3=[]
d=dict()
for i in range(n):
    l1=input().split(" ")
    l2.append(l1[0])
    l1.pop(0)
    l3.append(l1)
for j in range(len(l3)):
    sum=0
    for k in l3[j]:
        sum+=int(k)
    l3[j]=sum
for i in range(len(l2)):
    d[l2[i]]=l3[i]
dn=dict(sorted(d.items(),key=lambda items:items[1]))
for key,value in dn.items():
    print(f'{key} {value}')
```

### **Output:**

	Input	Expected	Got	
✓	2 Gfg 6 7 4 Best 7 6 5	Gfg 17 Best 18	Gfg 17 Best 18	✓
✓	2 Gfg 6 6 Best 5 5	Best 10 Gfg 12	Best 10 Gfg 12	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

Ex. No. : 9.3

Date: 4/5/24

Register No.: 231501053

Name: Gokulakkannan P

---

## **Winner of Election**

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 Johnny get maximum votes. Since John is alphabetically smaller, we print it. Use dictionary to solve the above problem

### **Sample Input:**

10  
John  
John  
Johnny  
Jamie  
Jamie  
Johnny  
Jack  
Johnny  
Johnny  
Jackie

### **Sample Output:**

Johnny

**For example:**

Input	Result
10 John John Johnny Jamie Jamie Johnny Jack Johnny Johnny Jackie	Johney

**PROGRAM**

```
n=int(input())
l1=[]
d=dict()
for i in range(n):
    l1.append(input())
for j in l1:
    d[j]=l1.count(j)
n1=max(d)
m=d[n1]
d.pop(n1)
n2=max(d)
if(m== d[n2]):
    if(len(n1)<len(n2)):
        print(n1)
    else:
        print(n2)
else:
    print(n1)
```

Output:

	Input	Expected	Got	
✓	10 John John Johny Jamie Jamie Johny Jack Johny Johny Jackie	Johny	Johny	✓
✓	6 Ida Ida Ida Kiruba Kiruba Kiruba	Ida	Ida	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

v



**Ex. No. : 9.4**

**Date: 4/5/24**

**Register No.: 231501053**

**Name: Gokulakkannan P**

---

## **Student Record**

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 has 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

Sample input:

4

James 67 89 56

Lalith 89 45 45

Ram 89 89 89

Sita 70 70 70

Sample Output:

Ram

James Ram

Lalith

Lalith

### **PROGRAM**

```
n=int(input())
list1=[]
list2=[]
list3=[]
list4=[]
list5=[]
for i in range(n):
    student_data=input()
    student_input_parts=student_data.split()
    name=student_input_parts[0]
    marks=[int(mark) for mark in student_input_parts[1:]]
    average=(marks[0]+marks[1]+marks[2])/3
    list1.append(name)
    list2.append(marks[0])
    list3.append(marks[1])
    list4.append(marks[2])
    list5.append(average)
l1=[]

for i in range(n):
    if list5[i]==max(list5):
        l1.append(list1[i])
for i in l1:
    print(i,end=' ')
print()
l2=[]
for i in range(n):
    if list3[i]==max(list3):
        l2.append(list1[i])
for i in l2:
    print(i,end=' ')
print()
l3=[]
for i in range(n):
    if list4[i]==min(list4):
        l3.append(list1[i])
l3.reverse()
for i in l3:
    print(i,end=' ')
print()
l4=[]
for i in range(n):
    if list5[i]==min(list5):
        l4.append(list1[i])
for i in l4:
    print(i,end=' ')
```

Output:

	Input	Expected	Got	
✓	4 James 67 89 56 Lalith 89 45 45 Ram 89 89 89 Sita 70 70 70	Ram James Ram Lalith Lalith	Ram James Ram Lalith Lalith	✓
✓	3 Raja 95 67 90 Aarav 89 90 90 Shadhana 95 95 91	Shadhana Shadhana Aarav Raja Raja	Shadhana Shadhana Aarav Raja Raja	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

Ex. No. : 9.5

Date: 4/5/24

Register No.: 231501053

Name: Gokulakkannan P

---

## Scramble Score

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.

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.

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

Sample Input

REC

Sample Output

REC is worth 5 points.

### PROGRAM

```
s=input()
```

```
l1=list(s)
```

```
d=dict([(1,['A','E','I','L','N','O','R','S','T','U']),  
(2,['D','G']),  
(3,['B','C','M','P']),  
(4,['F','H','V','W','Y']),  
(5,['K']),  
(8,['J','X']),  
(10,['Q','Z'])])
```

```
sum=0
```

```
for j in l1:
```

```
    for key,value in d.items():
```

```
        if j in value:
```

```
            sum+=key
```

```
print(f'{s} is worth {sum} points.')
```

Output:

	Input	Expected	Got	
✓	GOD	GOD is worth 5 points.	GOD is worth 5 points.	✓
✓	REC	REC is worth 5 points.	REC is worth 5 points.	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.