**09–Dictionary**

**Ex. No. : 9.1 Date: 22/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## Uncommonwords

A sentence isa string of single-space separated wordswhere eachword consistsonly 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.

Example1:

Input:s1="thisappleissweet",s2="thisappleissour" Output: ["sweet","sour"]

Example2:

Input:s1="appleapple",s2="banana" Output: ["banana"]

Constraints:

1<=s1.length,s2.length<=200

s1ands2consistoflowercaseEnglishlettersandspaces. s1 and s2 do not have leading or trailing spaces.

Allthewordsins1ands2areseparatedbyasinglespace. Note:

Usedictionarytosolvetheproblem

**For example:**

|  |  |
| --- | --- |
| **Input** | **Result** |
| thisappleissweet this apple is sour | sweetsour |

# Program:

a=input().split() b=input().split() c1,c2,l={},{},[]

for i in a: c1[i]=c1.get(i,0)+1

for j in b: c2[j]=c2.get(j,0)+1

for w,c in c1.items(): if(c==1andwnotinb):

l.append(w)

for w,c in c2.items(): if(c==1andwnotina):

l.append(w) print(\*l)

# Output:



**Ex. No. : 9.2 Date: 22/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## SortDictionarybyValuesSummation

Giveadictionarywithvaluelists,sortthekeysbysummationofvaluesinvalue list.

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

**Output**:{‘Gfg’:17,‘best’:18}

**Explanation**:Sortedbysum,andreplaced. **Input** :test\_dict = {‘Gfg’ : [8,8], ‘best’ : [5,5]} **Output** : {‘best’: 10, ‘Gfg’: 16}

**Explanation**:Sortedbysum,andreplaced. Sample Input:

2

Gfg 674

Best 7 6 5 SampleOutput Gfg 17

Best18

**For example:**

|  |  |
| --- | --- |
| **Input** | **Result** |
| 2  Gfg674  Best765 | Gfg17  Best18 |

# Program:

**a=int(input()) d={}**

**for i in range(a): b=input() b=b.partition(" ") d[b[0]]=b[-1].split("")**

**n=list(d.values()) k=list(d.keys())**

**foriinrange(len(n)): s=0**

**forjinrange(len(n[i])):**

**s+=int(n[i][j])**

**d.update({k[i]:s}) l=list(d.items()) if(l[0][1]<l[1][1]):**

**fork,vind.items(): print(k,v)**

**else:**

**j=1**

**fork,vind.items(): if(j==1):**

**k1,v1=k,vj+=1**

**else:**

**print(k,v) print(k1,v1)**

# Output:



**Ex. No. : 9.3 Date: 22/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## WinnerofElection

Given an array of names of candidates in an election. Acandidate 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 candidatesJohnandJohnygetmaximumvotes.SinceJohnisalphabeticallysmaller, we print it. Use dictionary to solve the above problem

**SampleInput:**

10

John John Johny Jamie JamieJohny Jack Johny JohnyJackie

**SampleOutput:**

Johny

**For example:**

|  |  |
| --- | --- |
| **Input** | **Result** |
| 10  John JohnJohny Jamie JamieJohny Jack | Johny |

|  |  |
| --- | --- |
| **Input** | **Result** |
| Johny Johny Jackie |  |

# Program:

n=int(input())

votes= {}

for \_ in range(n): candidate=input()

votes[candidate]=votes.get(candidate,0)+1

max\_votes=max(votes.values())

max\_candidates=[candidateforcandidate,countin votes.items() if count == max\_votes]

print(min(max\_candidates))

# Output:



**Ex. No. : 9.4 Date: 22/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## StudentRecord

Createastudentdictionaryfornstudentswiththestudentnameaskeyand their test mark assignment mark and lab mark as values. Do the following computations and display the result.

1. Identifythestudentwiththehighest[average](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5717)score
2. IdentifythestudentwhoasthehighestAssignmentmarks
3. IdentifythestudentwiththeLowestlabmarks
4. Identifythestudentwiththelowest[average](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5717)score Note:

Ifmorethanonestudenthasthesamescoredisplayallthestudentnames

Sampleinput:

4

James67 8956

Lalith894545

Ram898989

Sita 70 70 70 SampleOutput:

Ram

JamesRam Lalith Lalith

# Program:

n = int(input()) max\_average=float('-inf') min\_average = float('inf')

max\_assignment = float('-inf') min\_lab = float('inf') max\_average\_students = [] max\_assignment\_students=[] min\_lab\_students = [] min\_average\_students = []

for\_in range(n):

name,test,assignment,lab=input().split() test = int(test)

assignment=int(assignment) lab = int(lab)

average=(test+assignment+lab)/3 if average >max\_average:

max\_average = average max\_average\_students=[name]

elif average == max\_average: max\_average\_students.append(name)

if average <min\_average: min\_average = average min\_average\_students=[name]

elif average == min\_average: min\_average\_students.append(name)

if assignment >max\_assignment: max\_assignment = assignment max\_assignment\_students=[name]

elif assignment == max\_assignment: max\_assignment\_students.append(name)

iflab<min\_lab: min\_lab=lab

min\_lab\_students=[name] elif lab == min\_lab:

min\_lab\_students.append(name) print(\*sorted(max\_average\_students)) print(\*sorted(max\_assignment\_students)) print(\*sorted(min\_lab\_students)) print(\*sorted(min\_average\_students))

# Output:



|  |  |  |
| --- | --- | --- |
| **Ex. No. :** | **9.5** | **Date: 22/5/24** |
| **Register No.: 231801171** |  | **Name: Stergio Eugin**  [**ScrambleScore**](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5780) |

In the game of Scrabble™, each letter has points associated with it. The total score of awordisthesumofthescoresofitsletters.Morecommonlettersareworth fewer points while less common letters are worth more points.

Write a program that computes and displays the Scrabble™ score for a word. Createadictionarythatmapsfromletterstopointvalues.Thenusethedictionary 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.

Thepointsassociatedwitheachletterareshownbelow: Points Letters

1. A, E, I,L,N,O,R,S,Tand U
2. DandG
3. B,C,M andP
4. F, H,V, WandY
5. K

8 J and X 10QandZ

[Sample](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5127)Input REC

[Sample](https://www.rajalakshmicolleges.net/moodle/mod/quiz/view.php?id=5127)Output

RECisworth 5points.

# Program:

letter\_scores={

'A':1,'E':1,'I':1,'L':1,'N':1,'O':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

}

word = input().upper()

score=sum(letter\_scores.get(letter,0)forletterinword) print(word,"isworth",score,"points.")

Output:

