[Dashboard](http://118.185.187.137/moodle/my/) / [My courses](http://118.185.187.137/moodle/my/courses.php) / [CS23221-PPL-2023](http://118.185.187.137/moodle/course/view.php?id=148) / [Experiments based on Dictionary and its operations.](http://118.185.187.137/moodle/course/view.php?id=148&section-8) / [Week8\_Coding](http://118.185.187.137/moodle/mod/quiz/view.php?id=1077)

**Started on** Wednesday, 29 May 2024, 6:51 PM

**State** Finished

**Completed on** Wednesday, 29 May 2024, 8:14 PM

**Time taken** 1 hour 22 mins

**Marks** 5.00/5.00

**Grade 100.00** out of 100.00

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.

Question **1**

Correct

Mark 1.00 out of 1.00

Given two sentences s1 and s2, return a list of all the uncommon words. You may return the answer in any order. Example 1:

Input: s1 = "this apple is sweet", s2 = "this apple is sour"

Output: ["sweet","sour"] Example 2:

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

Constraints:

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

s1 and s2 consist of lowercase English letters and spaces. s1 and s2 do not have leading or trailing spaces.

All the words in s1 and s2 are separated by a single space.

Note:

Use dictionary to solve the problem

# For example:

|  |  |
| --- | --- |
| **Input** | **Result** |
| this apple is sweet this apple is sour | sweet sour |

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1 ▼ | def uncommon\_words(s1, s2): |
| 2 | words = s1.split() + s2.split() |
| 3 | word\_count = {} |
| 4 ▼ | for word in words: |
| 5 | word\_count[word] = word\_count.get(word, 0) + 1 |
| 6 | return [word for word in word\_count if word\_count[word] == 1] |
| 7 |  |
| 8 | s1 = input() |
| 9 | s2 = input() |
| 10 | x=uncommon\_words(s1, s2) |
| 11 |  |
| 12 ▼ | for i in x: |
| 13 | print(i,end=" ") |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **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.

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

Question **2**

Correct

Mark 1.00 out of 1.00

**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 |

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1 | n=int(input()) |
| 2 | dict1={} |
| 3 | l1=[] |
| 4 ▼ | for i in range(n): |
| 5 | str1=input() |
| 6 | list1=str1.split() |
| 7 | key1=list1[0] |
| 8 | l=len(list1) |
| 9 | list2=[] |
| 10 ▼ | for j in range(1,l): |
| 11 | list2.append(list1[j]) |
| 12 | list2=map(int,list2) |
| 13 | dict1[key1]=sum(list2) |
| 14 | list3=sorted(dict1.items(), key=lambda x: (x[1], x[0])) |
| 15 | l3=len(list3) |
| 16 ▼ | for i in range(l3): |
| 17 |  |
| 18 | tup=list3[i] |
| 19 ▼ | for j in range(2): |
| 20 | print(tup[j],end=" ") |
| 21 | print(" ") |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **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.

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.

Question **3**

Correct

Mark 1.00 out of 1.00

# 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

# Sample Input:

10

John John Johny Jamie Jamie Johny Jack Johny Johny Jackie

# Sample Output:

Johny

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1 | a=int(input()) |

2 d=[]

3 e={}

1. ▼ for i in range(a):
2. x=input()
3. d.append(x)
4. ▼ for i in d:
5. ▼ if i in e:

9 e[i]+=1

10 ▼ else:

11 e[i]=1

1. d=max(e,key=e.get)
2. print(d)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **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.

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:

Question **4**

Correct

Mark 1.00 out of 1.00

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.

[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

REC is worth 5 points.

# For example:

|  |  |
| --- | --- |
| **Input** | **Result** |
| REC | REC is worth 5 points. |

**Answer:** (penalty regime: 0 %)

|  |  |
| --- | --- |
| 1 | a=input() |
| 2 | c=0 |
| 3 ▼ | for i in a: |
| 4 ▼ | if i in ['A','E','I','L','N','O','R','S','T','U']: |
| 5 | c+=1 |
| 6 ▼ | elif i in ['D','G']: |
| 7 | c+=2 |
| 8 ▼ | elif i in ['B','C','M','P']: |
| 9 | c+=3 |
| 10 ▼ | elif i in ['F','H','V','W','Y']: |
| 11 | c+=4 |
| 12 ▼ | elif i in ['K']: |
| 13 | c+=5 |
| 14 ▼ | elif i in ['J','X']: |
| 15 | c+=8 |
| 16 ▼ | else: |
| 17 | c+=10 |
| 18 | print(a,"is worth",c,"points.") |
| 19 |  |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **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.

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.

Question **5**

Correct

Mark 1.00 out of 1.00

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

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

# For example:

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

**Answer:** (penalty regime: 0 %)

1. n=int(input())
2. listkey=[]
3. listsum=[]
4. listasm=[]
5. listlab=[]
6. ▼ for i in range(n):
7. str1=input()
8. list1=str1.split()
9. key1=list1[0]
10. listkey.append(key1)
11. l=len(list1)
12. list2=[]
13. ▼ for j in range(1,l):
14. list2.append(list1[j])
15. list3=[]
16. ▼ for j in range(len(list2)):
17. list3.append(int(list2[j]))
18. listsum.append(sum(list3))
19. listasm.append(list3[1])
20. listlab.append(list3[2])
21. maxasm=max(listasm)
22. maxsum=max(listsum)
23. minlab=min(listlab)
24. minsum=min(listsum)
25. listmaxasm=[]
26. listmaxsum=[]
27. listminlab=[]
28. listminsum=[]
29. ▼ for i in range(len(listkey)):
30. ▼ if (listsum[i]==maxsum):
31. listmaxsum.append(listkey[i])
32. listmaxsum.sort()
33. ▼ for i in range(len(listkey)):
34. ▼ if (listasm[i]==maxasm):
35. listmaxasm.append(listkey[i])
36. listmaxasm.sort()
37. ▼ for i in range(len(listkey)):
38. ▼ if (listlab[i]==minlab):
39. listminlab.append(listkey[i])
40. listminlab.sort()
41. ▼ for i in range(len(listkey)):
42. ▼ if (listsum[i]==minsum):
43. listminsum.append(listkey[i])
44. listminsum.sort()
45. ▼ for i in range(len(listmaxsum)):
46. print(listmaxsum[i],end=" ")
47. print(" ")
48. ▼ for i in range(len(listmaxasm)):
49. print(listmaxasm[i],end=" ")
50. print(" ")
51. ▼ for i in range(len(listminlab)):
52. print(listminlab[i],end=" ")

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

Passed all tests! 

**Correct**

Marks for this submission: 1.00/1.00.

[◄ Week8\_MCQ](http://118.185.187.137/moodle/mod/quiz/view.php?id=1076&forceview=1)

Jump to...

[Week9\_MCQ ►](http://118.185.187.137/moodle/mod/quiz/view.php?id=1079&forceview=1)