

Book_28_Apr_2024[1].pdfInbox (1,417) - 220301107UNIT IV- Correlation and Unit III- Random ProcessWeek8_Coding: Attempt sushmitha011/sushmitha

←→↻🔒 Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=16611&cmid=110

REC-PS

SUSHMITHA SREE S 2022-BIOMED-B S2

```
4 word_count = {}
5 for word in words1:
6     word_count[word] = word_count.get(word, 0) + 1
7 for word in words2:
8     word_count[word] = word_count.get(word, 0) + 1
9     uncommon_words = [word for word in word_count if word_count[word] == 1]
10 return uncommon_words
11 s1 = input().strip()
12 s2 = input().strip()
13 result = uncommon_from_sentences(s1, s2)
14 print(" ".join(result))
```

	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.

Book_28_Apr_2024[1].pdfInbox (1,417) - 220301107UNIT IV- Correlation and Unit III- Random ProcessWeek8_Coding: Attempt sushmitha011/sushmitha

←→↻🔒 Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=16611&cmid=110

REC-PS

SUSHMITHA SREE S 2022-BIOMED-B S2

Question 4

Correct

Mark 1.00 out of 1.00

🚩 Flag question

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.

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

Book_28_Apr_202411.pdf | Inbox (1,417) - 220301107 | UNIT IV: Correlation and ... | Unit III: Random Process | Week8_Coding_Attempt | sushmitha011/sushmitha | +

← → ↺ 🏠 🔒 Not secure rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=16611&cmid=110 ☆ ⓘ

REC-PS

SUSHMITHA SREE S 2022-BIOMED-B S2

Answer: (penalty regime: 0 %)

```
1 from collections import Counter
2 def winner(votes):
3     vote_count = Counter(votes)
4     max_votes = max(vote_count.values())
5     max_candidates = [candidate for candidate, count in vote_count.items() if count == max_votes]
6     return min(max_candidates)
7 n = int(input().strip())
8 votes = [input().strip() for _ in range(n)]
9 result = winner(votes)
10 print(result)
```

	Input	Expected	Got	
✓	10 John John Johnny	Johnny	Johnny	✓

Book_28_Apr_202411.pdf | Inbox (1,417) - 220301107 | UNIT IV- Correlation and | Unit III- Random Process | Week8_Coding-Attempt | sushmitha011/sushmitha | +

← → ↻ 🏠 **Not secure** rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=16611&cmid=110 ☆ ⓘ

REC-PS

SUSHMITHA SREE S 2022-BIOMED-B S2

Question 3

Correct

Mark 1.00 out of

🚩 Flag question

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

Sample Input:

10
John
John
Johnny
Jamie
Jamie
Johnny
Jack
Johnny

Windows taskbar showing the Start button, Search bar, and various application icons (including File Explorer, Edge, and WhatsApp). The system tray on the right displays the date and time as 21:05 on 19-06-2024.

REC-PS

SUSHMITHA SREE S 2022-BIOMED-B S2

Answer: `userinput requires 0 try`

```
1 def parse_input():
2     n = int(input())
3     test_dict = {}
4     for _ in range(n):
5         entry = input().split()
6         key = entry[0]
7         values = list(map(int, entry[1:]))
8         test_dict[key] = values
9     return test_dict
10
11 def sort_and_sum_dict(test_dict):
12     summed_dict = {k: sum(v) for k, v in test_dict.items()}
13     sorted_dict = dict(sorted(summed_dict.items(), key=lambda item: item[1]))
14     return sorted_dict
15
16 test_dict = parse_input()
17 sorted_summed_dict = sort_and_sum_dict(test_dict)
18 for key, value in sorted_summed_dict.items():
19     print(f"{key} {value}")
```

	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	✓

REC-PS

SUSHMITHA SREE S 2022-BIOMED-B S2

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

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

