GE19211 / GE23233 / GE23231 - PSPP/PUP

Quiz navigation

1 ✓ 2 ✓ 3 ✓ 4 ✓ 5 ✓

Show one page at a time

Finish review

Started on	Thursday, 30 May 2024, 2:12 PM
State	Finished
Completed on	Thursday, 30 May 2024, 2:16 PM
Time taken	3 mins 43 secs
Marks	5.00/5.00
Grade	**100.00** out of 100.00

Question **1**
Correct
Mark 1.00 out of 1.00
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

Johny

Jamie

Jamie

Johny

Jack

Johny

Johny

Jackie

Sample Output:

Johny

Answer: (penalty regime: 0 %)

```python
A = [input() for _ in range(int(input()))]
B = {name: A.count(name) for name in set(A)}
print(min(name for name, count in B.items() if count == max(B.values())))


```

	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.

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

Answer: (penalty regime: 0 %)

```python
n = int(input())
test_dict = {key: sum(map(int, values)) for key, *values in (input().split() for _ in range(n))}
sorted_dict = {key: value for key, value in sorted(test_dict.items(), key=lambda x: x[1])}
for key, value in sorted_dict.items():
    print(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	✓

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

Question **3**
Correct
Mark 1.00 out of 1.00
Flag question

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

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 James 67 89 56 Lalith 89 45 45 Ram 89 89 89 Sita 70 70 70	Ram James Ram Lalith Lalith

Answer: (penalty regime: 0 %)

```python
n = int(input())
students = {}
for _ in range(n):
    name, test, assignment, lab = input().split()
    students[name] = {'test': int(test), 'assignment': int(assignment), 'lab': int(lab)}
averages = {name: sum(info.values()) / 3 for name, info in students.items()}

a = max(averages.values())
A = sorted([name for name, avg in averages.items() if avg == a])

b = max((info['assignment'] for info in students.values()))
B = sorted([name for name, info in students.items() if info['assignment'] == b])

c = min((info['lab'] for info in students.values()))
C = sorted([name for name, info in students.items() if info['lab'] == c])

d = min(averages.values())
D = sorted([name for name, avg in averages.items() if avg == d])

print('\n'.join([" ".join(A), " ".join(B), " ".join(C), " ".join(D)]))
```

	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.

Question **4**
Correct
Mark 1.00 out of 1.00
Flag question

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.

Sample Input

REC

Sample Output

REC is worth 5 points.

For example:

Input	Result
REC	REC is worth 5 points.

Answer: (penalty regime: 0 %)

```python
A = {'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()
B = sum(A.get(letter, 0) for letter in word)
print(f"{word} is worth {B} points.")
```

	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.

Question **5**
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

Answer: (penalty regime: 0 %)

```python
s1, s2 = input().split(), input().split()
c1, c2 = {}, {}
for w in s1: c1[w] = c1.get(w, 0) + 1
for w in s2: c2[w] = c2.get(w, 0) + 1
A = [w for w, c in c1.items() if c == 1 and w not in c2]
A += [w for w, c in c2.items() if c == 1 and w not in c1]
print(*A, 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.

Finish review

◄ Week8_MCQ Jump to... Functions ►