WEEK_10-Dictionary

Started on	Friday, 26 April 2024, 7:16 PM
State	Finished
Completed on	Sunday, 5 May 2024, 11:07 AM
Time taken	8 days 15 hours
Marks	7.00/7.00
Grade	**50.00** out of 50.00 (**100**%)
Name	SAKTHI MAHESWARI C 2022-CSD-A

WEEK_10-Dictionary

Question **1**

Correct

Mark 1.00 out of 1.00

Flag question

Multiply All the Items in a Dictionary

Input: Any input in Dictionary format (Ex: d={'A':10,'B':10,'C':239})

Output: multiplication of dictionary values (23900)

Answer: (penalty regime: 0 %)

```
d={'A':10,'B':10,'C':239}
prod=1
for i in d:
    prod=prod*d[i]
print(prod)
```

	Input	Expected	Got	
✔	d={'A':10,'B':10,'C':239}	23900	23900	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

WEEK_10-Dictionary

Question **2**

Correct

Mark 1.00 out of 1.00

Flag question

A teacher wants to evaluate her class results for the subject she handles. She want to do the following analysis:

1. Display Class average
2. Display Maximum mark Roll no
3. Display Minimum mark Roll no

Kindly help her out. Use dictionary for storing the student details.

Input Format:

In line 1 no of students will be given

Followed by n lines containing student rollno and marks

Output Format:

Line 1 Class average

Line 2 Maximum mark Roll no

Line 3 Minimum mark Roll no

Answer: (penalty regime: 0 %)

```
n=int(input())
d=dict()
for i in range(0,n):
    s1=input()
    l1=s1.split(" ")
    #print(l1)
    d[l1[0]]=l1[1]
roll=[]
mark=[]
sum=0
for i in d:
    roll.append(i)
    mark.append(d[i])
    sum=sum+int(d[i])
print(int(sum/n))
maxi=max(mark)
mini=min(mark)
for i in d:
    if d[i]==maxi:
        print(i)
for i in d:
    if d[i]==mini:
        print(i)
```

	Input	Expected	Got	
✔	4 01 87 02 99 03 45 04 77	77 02 03	77 02 03	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

WEEK_10-Dictionary

Question **3**

Correct

Mark 1.00 out of 1.00

Flag question

Create a program that determines and displays the number of unique characters in a string entered by the user. For example, Hello, World! has 10 unique characters while zzz has only one unique character. Use a dictionary or set to solve this problem.

For example:

Input	Result
Hello, World!	10

Answer: (penalty regime: 0 %)

```
s1=input()
set1=set()
for i in s1:
    set1.add(i)
print(len(set1))

```

	Input	Expected	Got	
✔	Hello, World!	10	10	✔
✔	zzz	1	1	✔
✔	RECCSE	4	4	✔
✔	AAABBBCCC	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

WEEK_10-Dictionary

Question **4**

Correct

Mark 1.00 out of 1.00

Flag question

To Check if a Given Key Exists in a Dictionary or Not

Input: Any dictionary format input (Ex: d={'A':1,'B':2,'C':3})

Enter Key to check: A

Output:

Key is present and value of the key is: (location)

Present # True Statement

Not Present # False Statement

Answer: (penalty regime: 0 %)

```
d={"A":1,"B":2,"C":3}
k=input()
c=0
for i in d.keys():
    if i==k:
        c=1
if(c==1):
    print("Present")
else:
    print("Not Present")
```

	Input	Expected	Got	
✔	A	Present	Present	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

WEEK_10-Dictionary

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

Two words are anagrams if they contain all of the same letters, but in a different order. For example, “evil” and “live” are anagrams because each contains one “e”, one “i”, one “l”, and one “v”. Create a program that reads two strings from the user, determines whether or not they are anagrams, and reports the result.

Sample Input 1

evil

live

Sample Output 1

Those strings are anagrams.

Answer: (penalty regime: 0 %)

```
s1=input()
s2=input()
c=0
for i in s1:
    for i in s2:
        if(s1.count(i)==s2.count(i)) and (len(s1)==len(s2)):
            c=1
        else:
            c=0
if(c==1):
    print("Those strings are anagrams.")
else:
    print("Those strings are not anagrams.")
```

	Input	Expected	Got	
✔	evil live	Those strings are anagrams.	Those strings are anagrams.	✔
✔	meet met	Those strings are not anagrams.	Those strings are not anagrams.	✔
✔	rec cer	Those strings are anagrams.	Those strings are anagrams.	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

WEEK_10-Dictionary

Question **6**

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

Answer: (penalty regime: 0 %)

```
d={1:["A","E","I","L","O","N","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"]}
s1=input()
score=0
for i in s1:
    if i in d[1]:
        score=score+1
    elif i in d[2]:
        score=score+2
    elif i in d[3]:
        score=score+3
    elif i in d[4]:
        score=score+4
    elif i in d[5]:
        score=score+5
    elif i in d[8]:
        score=score+8
    else:
        score=score+10
print("{} is worth {} points.".format(s1,score))
```

	Input	Expected	Got	
✔	REC	REC is worth 5 points.	REC is worth 5 points.	✔
✔	RAJALAKSHMI	RAJALAKSHMI is worth 27 points.	RAJALAKSHMI is worth 27 points.	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

WEEK_10-Dictionary

Question **7**
Correct
Mark 1.00 out of 1.00
Flag question

A sentence is a list of words that are separated by a single space with no leading or trailing spaces. Each word consists of lowercase and uppercase English letters.

A sentence can be shuffled by appending the 1-indexed word position to each word then rearranging the words in the sentence.

For example, the sentence "This is a sentence" can be shuffled as "sentence4 a3 is2 This1" or "is2 sentence4 This1 a3".

Given a shuffled sentence s containing no more than 9 words, reconstruct and return the original sentence.

Example 1:

Input:

is2 sentence4 This1 a3

Output:

This is a sentence

Answer: (penalty regime: 0 %)

```
s=input()
l1=s.split(" ")
d=dict()
for i in l1:
    key=i[-1]
    val=i[0:-1]
    d[key]=val
for i in sorted(d):
    print(d[i],end=" ")
```

	Input	Expected	Got	
✔	is2 sentence4 This1 a3	This is a sentence	This is a sentence	✔
✔	Myself2 Me1 Vijay4 and3	Me Myself and Vijay	Me Myself and Vijay	✔

Passed all tests! ✔