CS23336-Introduction to Python Programming

Started on	Saturday, 9 November 2024, 9:08 PM
State	Finished
Completed on	Saturday, 9 November 2024, 10:19 PM
Time taken	1 hour 10 mins
Marks	10.00/10.00
Grade	**100.00** out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

Flag question

Question text

A company wants to send its quotation secretly to its client. The company decided to encrypt the amount they are sending to their client with some special symbols so that the equation amount will not be revealed to any external person. They used the special symbols !,@,#,$,%,^,&,*,>,< for 0,1,2,3,4,5,6,7,8,9 respectively. Write a python code to help the company to convert the amount to special symbols.

(Value rounded off to 2 decimal points)

Input

n: Float data type which reads amount to send

Output

s: : String data type which displays symbols

Sample Testcase 1

Input

10000

Output

@!!!!.!!

Sample Testcase2

1234.56

Output

@#$%.^&

For example:

Input	Result
1345.23	@$%^.#$
15000.59	@^!!!.^<
156789	@^&*><.!!

Answer:(penalty regime: 0 %)

```python
def fun(a):
    s={'0':'!','1':'@','2':'#','3':'$','4':'%','5':'^','6':'&','7':'*','8':'>','9':'<'}
    a=round(a,2)
    astr=f"{a:.2f}"
    str1=''
    for char in astr:
        if char in s:
            str1+=s[char]
```

```
 9      else:
10          str1+=char
11      return str1
12  n=float(input())
13  print(fun(n))
```

Feedback

Input	Expected	Got
1345.23	@$%^.#$	@$%^.#$
15000.59	@^!!!.^<	@^!!!.^<
1234	@#$%.!!	@#$%.!!
156789	@^&*><.!!	@^&*><.!!

Passed all tests!

Correct
Marks for this submission: 1.00/1.00.

Question 2

Correct
Mark 1.00 out of 1.00

Question text

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 %)

```
1  def words(s1,s2):
2      w1=s1.split()
3      w2=s2.split()
4      uncommon=[]
5      for w in w1:
6          if w1.count(w)==1 and w not in w2:
7              uncommon.append(w)
8      for w in w2:
9          if w2.count(w)==1 and w not in w1:
10             uncommon.append(w)
11     return uncommon
12
13  s1=input()
14  s2=input()
15  print(*words(s1,s2))
```

Feedback

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.

Question 3

Correct
Mark 1.00 out of 1.00
⬛Flag question

Question text

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

Explanation: Sort the words in s to their original positions "This1 is2 a3 sentence4", then remove the numbers.

Example 2:

Input:

Myself2 Me1 I4 and3

Output:

Me Myself and I

Explanation: Sort the words in s to their original positions "Me1 Myself2 and3 I4", then remove the numbers.

Constraints:

2 <= s.length <= 200

s consists of lowercase and uppercase English letters, spaces, and digits from 1 to 9.

The number of words in s is between 1 and 9.

The words in s are separated by a single space.

s contains no leading or trailing spaces.

Answer:(penalty regime: 0 %)

```python
def recon():
    s=input()
    w=s.split()
    sw=sorted(w,key=lambda w1:int(''.join(filter(str.isdigit,w1))))
    o=' '.join(''.join(filter(str.isalpha,w1))for w1 in sw)
    return o
print(recon())
```

Feedback

Input	Expected	Got
is2 sentence4 This1 a3	This is a sentence	This is a sentence

Passed all tests!

Correct
Marks for this submission: 1.00/1.00.

Question 4

Correct
Mark 1.00 out of 1.00
Flag question

Question text

Given a number, convert it into corresponding alphabet.

Input	Output
1	A
26	Z
27	AA
676	YZ

Input Format

Input is an integer

Output Format

Print the alphabets

Constraints

1 <= num <= 4294967295
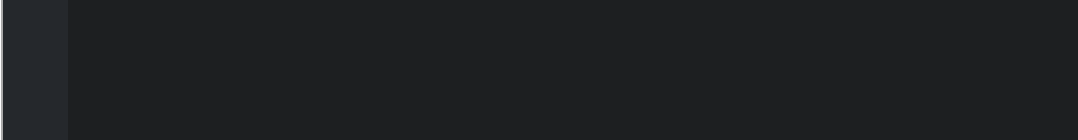
Sample Input 1

26

Sample Output 1

Z

For example:

Test	Result
print(excelNumber(26))	Z

Answer:(penalty regime: 0 %)

Reset answer

```python
def excelNumber(n):
    res=[]
    while n>0:
        n-=1
        rem=n%26
        res.append(chr(rem+ord('A')))
        n//=26
    return ''.join(res[::-1])
```

Feedback

Test	Expected	Got
print(excelNumber(26))	Z	Z
print(excelNumber(27))	AA	AA

Passed all tests!

Correct
Marks for this submission: 1.00/1.00.

Question 5

Correct
Mark 1.00 out of 1.00
🏳 Flag question

Question text

You are given a string `word`. A letter is called **special** if it appears both in lowercase and uppercase in `word`.

Your task is to return the number of **special** letters in `word`.

Constraints

- The input string `word` will contain only alphabetic characters (both lowercase and uppercase).
- The solution must utilize a dictionary to determine the number of special letters.
- The function should handle various edge cases, such as strings without any special letters, strings with only lowercase or uppercase letters, and mixed strings.

Examples

Example 1:

Input: word = "aaAbcBC"
Output: 3

Explanation:
The special characters in `word` are 'a', 'b', and 'c'.

Example 2:

Input: word = "abc"
Output: 0

Explanation:
No character in `word` appears in uppercase.

For example:

Test	Result
print(count_special_letters("AaBbCcDdEe"))	5

Answer:(penalty regime: 0 %)

Reset answer

```
1  def count_special_letters(word: str) -> int:
2      # Your implementation here
3      count={}
4      for char in word:
5          l=char.lower()
6          if l in count:
7              count[l].add(char)
8          else:
```

```
 10    count1=0 count[l]={char}
 11    for _set in count.values():
 12       if len(_set)>1:
 13           count1+=1
 14    return count1
```

Feedback

Test	Expected	Got
print(count_special_letters("AaBbCcDdEe"))	5	5
print(count_special_letters("ABCDE"))	0	0
print(count_special_letters("abcde"))	0	0

Passed all tests!

Correct
Marks for this submission: 1.00/1.00.

Question 6

Correct
Mark 1.00 out of 1.00

Question text

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

John

Jack

Johny

Johny

Jackie

Sample Output:

Johny

Answer:(penalty regime: 0 %)

```
1  def find(v):
2      count={}
3      for v1 in v:
4          if v1 in count:
5              count[v1]+=1
6          else:
7              count[v1]=1
8      mv=max(count.values())
9      c=[c1 for c1, co in count.items() if co==mv]
10     return min(c)
11 n=int(input())
12 v=[]
13 for _ in range(n):
14     v.append(input().capitalize())
15 print(find(v))
```

Feedback

Input	Expected	Got
10 John John Johny Jamie Jamie Johny Jack Johny Johny Jackie	Johny	Johny
6 Ida Ida Ida Kiruba Kiruba	Ida	Ida

Kiruba

Passed all tests!

Correct
Marks for this submission: 1.00/1.00.

Question 7

Correct
Mark 1.00 out of 1.00
⚑Flag question

Question text

Objective:

Develop a Python program that takes an input string from the user and counts the number of occurrences of each vowel (a, e, i, o, u) in the string. The program should be case-insensitive, meaning it should treat uppercase and lowercase vowels as the same.

Description:

Vowels play a significant role in the English language and other alphabet-based languages. Counting vowels in a given string is a fundamental task that can be applied in various text processing applications, including speech recognition, linguistic research, and text analysis. The objective of this problem is to create a Python script that accurately counts and displays the number of times each vowel appears in a user-provided string.

Program Requirements:

Input:

First line reading String as input, The string can contain any characters, including letters, numbers, and special characters.

Output:

Display the number of occurrences of each vowel in the string.

The output should list each vowel followed by its count.

Example:

Consider the following example for better understanding:

- **Input:** "Python Programming"
- **Output**

    **a = 1
    e = 0
    i = 1
    o = 2
    u = 0**

For example:

Input	Result
Hello World	a = 0 e = 1 i = 0 o = 2 u = 0
Python	a = 0 e = 0 i = 0 o = 1 u = 0

Answer:(penalty regime: 0 %)

```
1  def vowels(s):
2      vowel={'a':0,'e':0,'i':0,'o':0,'u':0}
3      s=s.lower()
4      for char in s:
5          if char in vowel:
```

```
  6              vowel[char]+=1
  7          return vowel
  8  s=input()
  9  count=vowels(s)
 10 ▾ for _ in 'aeiou':
 11      print(f"{_} = {count[_]}")
```

Feedback

Input	Expected	Got
Hello World	a = 0 e = 1 i = 0 o = 2 u = 0	a = 0 e = 1 i = 0 o = 2 u = 0
AEIOU aeio u	a = 2 e = 2 i = 2 o = 2 u = 2	a = 2 e = 2 i = 2 o = 2 u = 2
Python	a = 0 e = 0 i = 0 o = 1 u = 0	a = 0 e = 0 i = 0 o = 1 u = 0
abcdefghijklmnopqrstuvwxyz	a = 1 e = 1 i = 1 o = 1 u = 1	a = 1 e = 1 i = 1 o = 1 u = 1
12345!@#$%AEIOU	a = 1 e = 1 i = 1 o = 1 u = 1	a = 1 e = 1 i = 1 o = 1 u = 1

Passed all tests!

Correct
Marks for this submission: 1.00/1.00.

Question 8

Correct
Mark 1.00 out of 1.00

Question text

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
def cal(word):
    points={'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,
    'K':5,
    'J':8,'X':8,
    'Q':10,'Z':1}
    word=word.upper()
    score=sum(points.get(l,0) for l in word)
    return score
word=input()
score1=cal(word)
print(f"{word} is worth {score1} points.")
```

Feedback

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 9

Correct
Mark 1.00 out of 1.00
🏳Flag question

Question text

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 %)

```
1   st={}
2   n=int(input())
3   for _ in range(n):
4       data=input().split()
5       name=data[0]
6       test=int(data[1])
7       ass=int(data[2])
8       lab=int(data[3])
9       st[name]=(test,ass,lab)
10
11  havg=float('-inf')
12  lavg=float('inf')
13  hass=float('-inf')
14  llab=float('inf')
15
16  havgst=[]
17  lavgst=[]
18  hassst=[]
19  llabst=[]
```

```
20
21    for name,marks in st.items():
22        test,ass,lab=marks
23
24        avgscore=(test+ass+lab)/3
25
26        if avgscore>havg:
27            havg=avgscore
28            havgst=[name]
29        elif avgscore==havg:
30            havgst.append(name)
31
32        if ass>hass:
33            hass=ass
34            hassst=[name]
35        elif ass==hass:
36            hassst.append(name)
37
38        if lab<llab:
39            llab=lab
40            llabst=[name]
41        elif lab==llab:
42            llabst.append(name)
43
44        if avgscore<lavg:
45            lavg=avgscore
46            lavgst=[name]
47        elif avgscore==lavg:
48            lavgst.append(name)
49
50    havgst.sort()
51    hassst.sort()
52    llabst.sort()
```

Feedback

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 10

Correct
Mark 1.00 out of 1.00
Flag question

Question text

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
1  def dic(n,t):
2      sum1={}
3      for k,v in t.items():
4          sum1[k]=sum(v)
5      sdict=dict(sorted(sum1.items(),key=lambda x:x[1]))
6      for k,v in sdict.items():
7          print(f"{k} {v}")
8  n=int(input())
9  t={}
10 for _ in range(n):
11     e=input().split()
12     k=e[0]
13     v=list(map(int,e[1:]))
14     t[k]=v
15 dic(n,t)
```

Feedback

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.

Finish review

Quiz navigation

Finish review