Started on	Wednesday, 5 June 2024, 1:04 PM
State	Finished
Completed on	Wednesday, 5 June 2024, 2:46 PM
Time taken	1 hour 42 mins
Marks	3.00/5.00
Grade	**60.00** out of 100.00

Question **1**

Incorrect

Mark 0.00 out of 1.00

Given a tuple and a positive integer k, the task is to find the count of distinct pairs in the tuple whose sum is equal to **K**.

Examples:

Input: t = (5, 6, 5, 7, 7, 8), K = 13
Output: 2
Explanation:
Pairs with sum K(= 13) are {(5, 8), (6, 7), (6, 7)}.
Therefore, distinct pairs with sum K(= 13) are { (5, 8), (6, 7) }.
Therefore, the required output is 2.

For example:

Input	Result
1,2,1,2,5 3	1
1,2 0	0

Answer: (penalty regime: 0 %)

```python
1  x=input()
2  y=int(input())
3  a=x.split(',')
4  t=tuple(int(num) for num in a )
5  ans=set()
6  for i in range (len (t)):
7      for j in range (i+1,len(t)):
8          if t[i]+t[j]==y:
9              pair=(min(t[i],t[j]))
10             if pair not in ans:
11   ans.add((t[i],t[j]))
12 print(len(ans))
```

Syntax Error(s)

Sorry: IndentationError: unindent does not match any outer indentation level (__tester__.python3, line 10)

Incorrect

Marks for this submission: 0.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

The **DNA sequence** is composed of a series of nucleotides abbreviated as `'A'`, `'C'`, `'G'`, and `'T'`.

- For example, `"ACGAATTCCG"` is a **DNA sequence**.

When studying **DNA**, it is useful to identify repeated sequences within the DNA.

Given a string `s` that represents a **DNA sequence**, return all the `10`-**letter-long** sequences (substrings) that occur more than once in a DNA molecule. You may return the answer in **any order**.

Example 1:

```
Input: s = "AAAAACCCCCAAAAACCCCCAAAAAGGGTTT"
Output: ["AAAAACCCCC","CCCCCAAAAA"]
```

Example 2:

```
Input: s = "AAAAAAAAAAAA"
Output: ["AAAAAAAAAA"]
```

For example:

Input	Result
AAAAACCCCCAAAAACCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA

Answer: (penalty regime: 0 %)

```python
1   s=input()
2   d={}
3   a=[]
4   for i in range (len(s)-9):
5       x=s[i:i+10]
6       if x in d:
7           if d[x]==1:
8               a.append(x)
9           d[x]+=1
10      else: d[x]=1
11  for i in a:
12      print(i)
```

	Input	Expected	Got	
✓	AAAAACCCCCAAAAACCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA	AAAAACCCCC CCCCCAAAAA	✓
✓	AAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Write a program to eliminate the common elements in the given 2 arrays and print only the non-repeating

elements and the total number of such non-repeating elements.

Input Format:

The first line contains space-separated values, denoting the size of the two arrays in integer format respectively.

The next two lines contain the space-separated integer arrays to be compared.

Sample Input:

5 4

1 2 8 6 5

2 6 8 10

Sample Output:

1 5 10

3

Sample Input:

5 5

1 2 3 4 5

1 2 3 4 5

Sample Output:

NO SUCH ELEMENTS

For example:

Input	Result
5 4 1 2 8 6 5 2 6 8 10	1 5 10 3
5 5 1 2 3 4 5 1 2 3 4 5	NO SUCH ELEMENTS

Answer: (penalty regime: 0 %)

```python
1   a=input()
2   s1=input()
3   s2=input()
4   t1=s1.split()
5   t2=s2.split()
6   x=set(t1)
7   y=set(t2)
8   common=x. intersection(y)
9   z=x.union(y)
10  p=z-common
11  q=sorted(int (x) for x in p)
12  result=' '.join(map(str,q))
13  if len(q)==0:
14      print("NO SUCH ELEMENTS")
15  else:
16      print(result,)
17      print(len(q))
18
```

	Input	Expected	Got	
✓	5 4 1 2 8 6 5 2 6 8 10	1 5 10 3	1 5 10 3	✓
✓	3 3 10 10 10 10 11 12	11 12 2	11 12 2	✓
✓	5 5 1 2 3 4 5 1 2 3 4 5	NO SUCH ELEMENTS	NO SUCH ELEMENTS	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Incorrect

Mark 0.00 out of 1.00

Given an array of integers `nums` containing `n + 1` integers where each integer is in the range `[1, n]` inclusive.There is only **one repeated number** in `nums`, return *this repeated number*. Solve the problem using set.

Example 1:

Input: nums = [1,3,4,2,2]

Output: 2

Example 2:

Input: nums = [3,1,3,4,2]

Output: 3

For example:

Input	Result
1 3 4 4 2	4

Answer: (penalty regime: 0 %)

```
1  a=int(input())
2  word=[]
3  for i in range (a):
4      q=input()
5      word.append(q)
6  a=set("qwertyuiop")
7  b=set("asdfghjkl")
8  c=set("zxcvbnm")
9  result=[]
10 for j in work:
11     i=j.lower()
12     if set (i)<=a or set (i)<=b or set(i)<=c:
13         result.append(j)
14     if len (result)>0:
15         for i in result:
16             print(i)
17     else:
18         print("No words")
```

	Input	Expected	Got	

✗	1 3 4 4 2	4	***Run error***
			Traceback (most recent call last):
			  File "__tester__.python3", line 1, in <module>
			    a=int(input())
			    ^^^^^^^^^^^^
			ValueError: invalid literal for int() with base 10: '1 3 4 4 2'
				✗

Testing was aborted due to error.

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.

Question **5**

Correct

Mark 1.00 out of 1.00

There is a malfunctioning keyboard where some letter keys do not work. All other keys on the keyboard work properly.

Given a string text of words separated by a single space (no leading or trailing spaces) and a string brokenLetters of all distinct letter keys that are broken, return the number of words in text you can fully type using this keyboard.

Example 1:

Input: text = "hello world", brokenLetters = "ad"

Output:

1

Explanation: We cannot type "world" because the 'd' key is broken.

For example:

Input	Result
hello world ad	1
Faculty Upskilling in Python Programming ak	2

Answer: (penalty regime: 0 %)

```
1  a=input()
2  b=input()
3  x=set()
4  y=set()
5  for letter in a:
6      x.add(letter)
7  for letter in b:
8      y.add(letter)
9  z=0
10  for element in x:
11      if element in y:
12          z=z+1
13  print(z)
```

	Input	Expected	Got	
✓	hello world ad	1	1	✓
✓	Welcome to REC e	1	1	✓
✓	Faculty Upskilling in Python Programming ak	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◄ Week7_MCQ

Jump to...

Dictionary ►

◄ Week7_MCQ

Jump to...

Dictionary ►