Started on	Wednesday, 5 June 2024, 1:02 PM
State	Finished
Completed on	Wednesday, 5 June 2024, 2:20 PM
Time taken	1 hour 18 mins
Marks	5.00/5.00
Grade	**100.00** out of 100.00

Question **1**

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 %)

```python
1  a=list(input().split())
2  b=list(input())
3  c=0
4  for i in a:
5      d=0
6      for j in b:
7          if j in i.lower():
8              d+=1
9      if d== 0:
10         c+=1
11 print(c)
```

	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.

Question **2**

Correct

Mark 1.00 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]), max(t[i],t[j]))
10             if pair not in ans:
11                 ans.add((t[i],t[j]))
12 print(len(ans))
13
```

	Input	Expected	Got	
✓	5,6,5,7,7,8 13	2	2	✓
✓	1,2,1,2,5 3	1	1	✓
✓	1,2 0	0	0	✓

Passed all tests! ✓

Correct

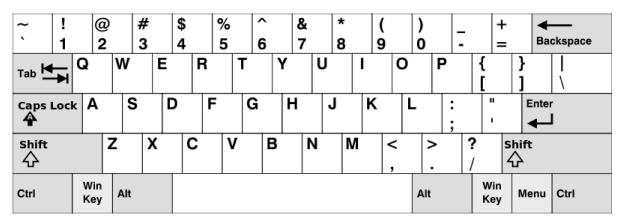Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Given an array of <u>strings</u> `words`, return *the words that can be typed using letters of the alphabet on only one row of American keyboard like the image below.*

In the **American keyboard**:

- the first row consists of the characters `"qwertyuiop"`,
- the second row consists of the characters `"asdfghjkl"`, and
- the third row consists of the characters `"zxcvbnm"`.

Example 1:

```
Input: words = ["Hello","Alaska","Dad","Peace"]
Output: ["Alaska","Dad"]
```

Example 2:

```
Input: words = ["omk"]
Output: []
```

Example 3:

```
Input: words = ["adsdf","sfd"]
Output: ["adsdf","sfd"]
```

For example:

Input	Result
4 Hello Alaska Dad Peace	Alaska Dad
2 adsfd afd	adsfd afd

Answer: (penalty regime: 0 %)

```
1  def findwords(words):
2      row1 = set('qwertyuiop')
3      row2 = set('asdfghjkl')
4      row3 = set('zxcvbnm')
5      result = []
```

6/19/24, 9:23 PMWeek7_Coding: Attempt review | REC-PS

```python
 6      for word in words:
 7          w = set(word.lower())
 8          if w.issubset(row1) or w.issubset(row2) or w.issubset(row3):
 9              result.append(word)
10      if len(result) ==0:
11          print("No words")
12      else:
13          for i in result:
14              print(i)
15  a=int(input())
16  arr = [input() for i in range(a)]
17  findwords(arr)
```

	Input	Expected	Got	
✓	4 Hello Alaska Dad Peace	Alaska Dad	Alaska Dad	✓
✓	1 omk	No words	No words	✓
✓	2 adsfd afd	adsfd afd	adsfd afd	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

www.rajalakshmicolleges.org/moodle/mod/quiz/review.php?attempt=23032&cmid=107 7/11

Question **4**

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 = "AAAAAAAAAAAAA"
Output: ["AAAAAAAAAA"]
```

For example:

Input	Result
AAAAACCCCCAAAAACCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA

Answer: (penalty regime: 0 %)

```
1  s=input()
2  substring_counts={}
3  for i in range(len(s)-9):
4      substring=s[i:i+10]
5      substring_counts[substring]=substring_counts.get(substring,0)+1
6  repeated_substrings=[substring for substring, count in substring_counts.items() if count>1]
7  for substring in repeated_substrings:
8      print(substring)
```

	Input	Expected	Got	
✓	AAAAACCCCCAAAAACCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA	AAAAACCCCC CCCCCAAAAA	✓

	Input	Expected	Got	
✓	AAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	✓

Passed all tests! ✓   9/11

> Correct

Marks for this submission: 1.00/1.00.

	Input	Expected	Got	
✓	AAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	✓

Question **5**

Correct

Mark 1.00 out of 1.00

Coders here is a simple task for you, Given string str. Your task is to check whether it is a binary string or not by using python set.

Examples:

Input: str = "01010101010"

Output: Yes

Input: str = "REC101"

Output: No

For example:

Input	Result
01010101010	Yes
010101 10101	No

Answer: (penalty regime: 0 %)

```
1  a=input()
2  try:
3      int(a)
4      print("Yes")
5  except:
6      print("No")
```

	Input	Expected	Got	
✓	01010101010	Yes	Yes	✓
✓	REC123	No	No	✓
✓	010101 10101	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

◄ Week7_MCQ

Jump to...

Dictionary ►

◄ Week7_MCQ