GE19211 / GE23233 / GE23231 - PSPP/PUP

Quiz navigation
1 ✓ 2 ✓ 3 ✓ 4 ✓ 5 ✓
Show one page at a time
Finish review

Started on	Tuesday, 21 May 2024, 1:36 PM
State	Finished
Completed on	Wednesday, 22 May 2024, 11:55 PM
Time taken	1 day 10 hours
Marks	5.00/5.00
Grade	**100.00** out of 100.00

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

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 %)

```python
def check_binary(str):
    # Create a set of characters in the string
    char_set = set(str)

    # Check if the set contains only '0' and '1'
    for char in char_set:
        if char not in {'0', '1'}:
            return "No"
    return "Yes"

str = input()

# Test the function
print(check_binary(str))  # Output: Yes

```

	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.

Question **2**
Correct
Mark 1.00 out of 1.00
Flag question

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 %)

```python
def find_duplicates(nums):
     num_set = set()
     for i in nums:
         if i in num_set:
            return i
         num_set.add(i)
nums = input().split()
print(find_duplicates(nums))


```

	Input	Expected	Got	
✓	`1 3 4 4 2`	`4`	`4`	✓
✓	`1 2 2 3 4 5 6 7`	`2`	`2`	✓

Passed all tests! ✓

Correct
Marks for this submission: 1.00/1.00.

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

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
def countWords(text, brokenLetters):
    brokenSet = set(brokenLetters)
    words = text.split(' ')
    count = 0
    for word in words:
        if not set(word) & brokenSet:
             count += 1
           #if any(letter in word for letter in brokenLetters):
              #continue
           #else:
            #count += 1
    return count
text = input().lower()
brokenLetters = input()
print(countWords(text, brokenLetters))
```

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

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
def count_distinct_pairs(t, K):
    seen = set()
    pairs = set()

    for num in t:
        complement = K - num
        if complement in seen:
            # Create a pair tuple with sorted order to avoid duplicate pairs
            pair = tuple(sorted((num, complement)))
            pairs.add(pair)
        seen.add(num)

    return len(pairs)

# Input handling
try:
    t_input = input()
    K = int(input())

    # Convert the input string to a tuple of integers
    t = tuple(map(int, t_input.split(',')))

    # Call the function and print the result
    print(count_distinct_pairs(t, K))
except ValueError:
    print("Invalid input. Please enter integers separated by commas for the tuple and a single integer for K.")
except Exception as e:
    print(f"An error occurred: {e}")

```

	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! ✓

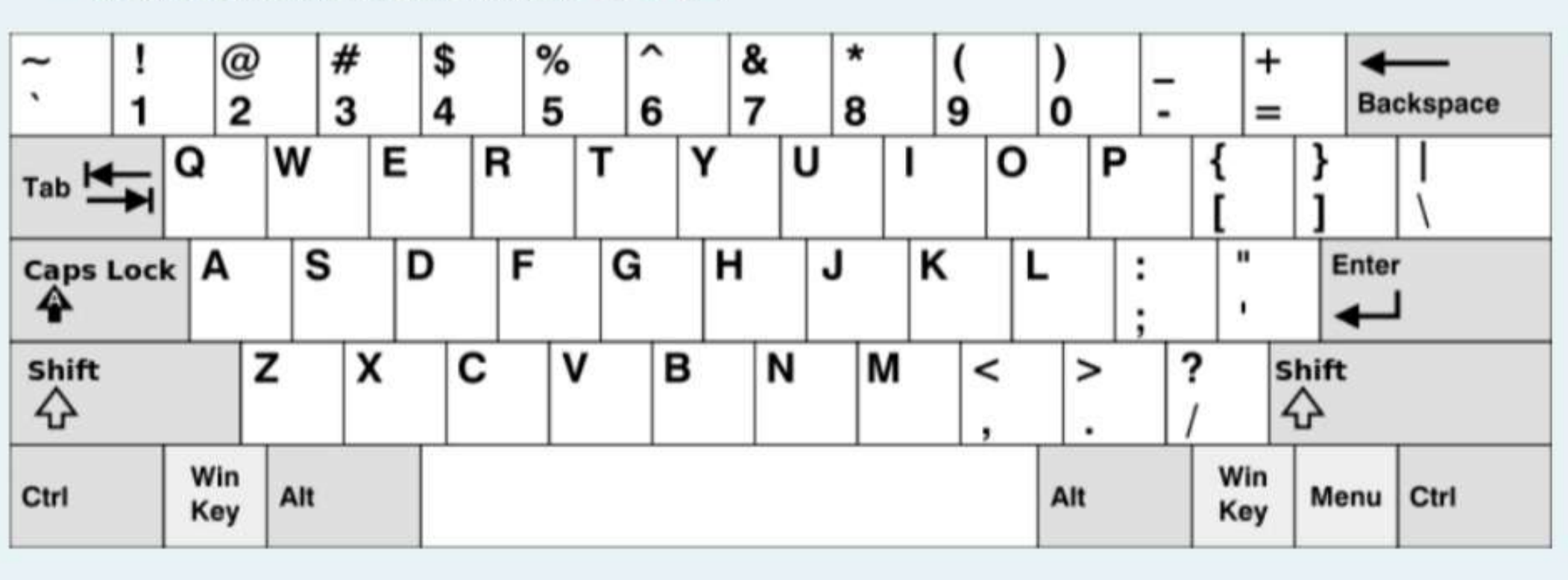
Correct
Marks for this submission: 1.00/1.00.

Question **5**
Correct
Mark 1.00 out of 1.00
Flag question

Given an array of strings `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 %)

```python
def findWords(words):
    row1 = set("qwertyuiop")
    row2 = set("asdfghjkl")
    row3 = set("zxcvbnm")

    def canBeTyped(word, row):
        return all(char in row for char in word.lower())

    result = []
    for word in words:
        if canBeTyped(word, row1) or canBeTyped(word, row2) or canBeTyped(word, row3):
            result.append(word)

    return result

# Get the number of words
num_words = int(input(""))

# Get the words from the user
words = []
for _ in range(num_words):
    word = input()
    words.append(word)

# Find and print the words that can be typed using one row of the keyboard
result = findWords(words)

if result:
    for word in result:
        print(word)
else:
    print("No words")


```

	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.

Finish review

◄ Week7_MCQ
Jump to...
Dictionary ►