

Started on	Saturday, 1 June 2024, 2:54 PM
State	Finished
Completed on	Saturday, 1 June 2024, 2:57 PM
Time taken	2 mins 47 secs
Marks	5.00/5.00
Grade	100.00 out of 100.00

Question 1

Correct

Mark 1.00 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 def find_duplicate(nums):
2     num_set = set()
3     for num in nums:
4         if num in num_set:
5             return num
6         num_set.add(num)
7     nums = input().split()
8     nums=[int(num)for num in nums]
9
10    print(find_duplicate(nums))
11
12

```

	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 **2**

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 %)

```
1 def non_repeating_elements(arr1, arr2):
2     set1 = set(arr1)
3     set2 = set(arr2)
4     non_repeating = set1.symmetric_difference(set2)
5     if not non_repeating:
6         print("NO SUCH ELEMENTS")
7     else:
8         print(*non_repeating)
9         print(len(non_repeating))
10
11 # Read input sizes and arrays
12 size1, size2 = map(int, input().split())
13 arr1 = list(map(int, input().split()))
14 arr2 = list(map(int, input().split()))
15
16 # Call the function
17 non_repeating_elements(arr1, arr2)
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 3

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 %)

```

1 def count_distinct_pairs(t, K):
2     seen = set()
3     pairs = set()
4
5     for num in t:
6         complement = K - num
7         if complement in seen:
8
9             pair = tuple(sorted((num, complement)))
10            pairs.add(pair)
11            seen.add(num)
12
13    return len(pairs)
14
15
16 try:
17     t_input = input()
18     K = int(input())
19
20
21     t = tuple(map(int, t_input.split(',')))
22
23
24     print(count_distinct_pairs(t, K))
25 except ValueError:
26     print("Invalid input. Please enter integers separated by commas for the tuple and a single integer fo
27 except Exception as e:
28     print(f"An error occurred: {e}")
29
30
31

```

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

	Input	Expected	Got	
✓	1, 2 0	0	0	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 1.00 out of 1.00

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".

~	1	@	#	\$	%	^	&	*	()	-	+	Backspace
Tab	Q	W	E	R	T	Y	U	I	O	P	{	}	
Caps Lock	A	S	D	F	G	H	J	K	L	:	"	Enter	
Shift	Z	X	C	V	B	N	M	<	>	?	Shift		
Ctrl	Win Key	Alt								Alt	Win Key	Menu	Ctrl

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
6     def canBeTyped(word, row):
7         return all(char in row for char in word.lower())
8
9     result = []
10    for word in words:
11        if canBeTyped(word, row1) or canBeTyped(word, row2) or canBeTyped(word, row3):
12            result.append(word)
13

```

```

13
14     return result
15
16 num_words = int(input(""))
17
18 words = []
19 for _ in range(num_words):
20     word = input()
21     words.append(word)
22
23
24 result = findWords(words)
25
26 if result:
27     for word in result:
28         print(word)
29 else:
30     print("No words")
31
32
33
34

```

	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.

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 s=input('')
2 binary_char={'0','1'}
3 if set(s)<=binary_char:
4     print("Yes")
5 else:
6     print("No")
7
8
```

	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 ▶

