

[Dashboard](#) / [My courses](#) / [PSP/PUP](#) / [Experiments based on Tuples, Sets and its operations](#) / [Week7 Coding](#)

Started on	Friday, 7 June 2024, 6:25 PM
State	Finished
Completed on	Friday, 7 June 2024, 6:56 PM
Time taken	31 mins 25 secs
Marks	4.00/5.00
Grade	80.00 out of 100.00

Question 1

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.

Question 2

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 x=input()
2 y=x.split()
3 z=list(y)
4 a=[]
5 b=[]
6 for element in z:
7     if element in a:
8         b.append(element)
9     else:
10        a.append(element)
11 c=' '.join(map(str,b))
12 print(c)
```

	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

Incorrect

Mark 0.00 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 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))
```

	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	✓

Your code failed one or more hidden tests.

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

Incorrect

Marks for this submission: 0.00/1.00.

Question 4

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+=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.

Question 5

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 = "AAAAACCCCCAAAAACCCCCAAAAGGGTTT"`

Output: `["AAAAACCCC", "CCCCAAAAA"]`

Example 2:

Input: `s = "AAAAAAAAAAAA"`

Output: `["AAAAAAAAA"]`

For example:

Input	Result
AAAAACCCCCAAAAACCCCCAAAAGGGTTT	AAAAACCCC CCCCAAAAA

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)
9

```

	Input	Expected	Got	
✓	AAAAACCCCCAAAAACCCCCAAAAGGGTTT	AAAAACCCC CCCCAAAAA	AAAAACCCC CCCCAAAAA	✓
✓	AAAAAAAAAAAAA	AAAAAAAAA	AAAAAAAAA	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

[◀ Week7_MCQ](#)

Jump to...

[Dictionary ▶](#)