

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

<b>Started on</b>	Saturday, 25 May 2024, 12:56 PM
<b>State</b>	Finished
<b>Completed on</b>	Saturday, 25 May 2024, 1:15 PM
<b>Time taken</b>	19 mins 17 secs
<b>Marks</b>	5.00/5.00
<b>Grade</b>	<b>100.00</b> out of 100.00

## Question 1

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: `["AAAAACCCCC", "CCCCAAAAA"]`

## Example 2:

Input: `s = "AAAAAAAAAAAA"`

Output: `["AAAAAAAAA"]`

## For example:

Input	Result
AAAAACCCCCAAAAACCCCCAAAAGGGTTT	AAAAACCCCC CCCCAAAAA

Answer: (penalty regime: 0 %)

```

1 s=input("")
2 seqs=set()
3 reseq=set()
4 for i in range(len(s)-9):
5     se=s[i:i+10]
6     if se in seqs:
7         reseq.add(se)
8     else:
9         seqs.add(se)
10 for se in sorted(reseq):
11     print(se)
```

	Input	Expected	Got	
✓	AAAAACCCCCAAAAACCCCCAAAAGGGTTT	AAAAACCCCC CCCCAAAAA	AAAAACCCCC CCCCAAAAA	✓
✓	AAAAAAAAAAAA	AAAAAAAAA	AAAAAAAAA	✓

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 %)

```

1 |
2 | t = tuple(map(int, input().split(',')))
3 | K = int(input())
4 |
5 |
6 | unique_pairs = set()
7 |
8 |
9 | for i in range(len(t)):
10 |
11 |     for j in range(i + 1, len(t)):
12 |         if t[i] + t[j] == K:
13 |
14 |             unique_pairs.add((min(t[i], t[j]), max(t[i], t[j])))
15 |
16 |
17 | print(len(unique_pairs))
18 |
19 |

```

	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

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

**Answer:** (penalty regime: 0 %)

```
1
2 size1, size2 = map(int, input().split())
3 arr1 = list(map(int, input().split()))
4 arr2 = list(map(int, input().split()))
5
6
7 non_repeating_elements = set(arr1) ^ set(arr2)
8
9 if non_repeating_elements:
10     print(*non_repeating_elements)
11     print(len(non_repeating_elements))
12 else:
13     print("NO SUCH ELEMENTS")
14
```

	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	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

## Question 4

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 n=str(input())
2 l=[]
3 for i in n:
4     if i=="0" or i=="1":
5
6         l.append(i)
7
8 if len(l)==len(n):
9     print("Yes")
10 else:
11     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 5

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     seen = set()
3     for num in nums:
4         if num in seen:
5             return num
6         seen.add(num)
7 if __name__ == '__main__':
8     nums = list(map(int, input().split()))
9     duplicate = find_duplicate(nums)
10    print(f"{duplicate}")

```

	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.

[◀ Week7\\_MCQ](#)

Jump to...



