07 – Tuple/Set

Ex. No. : 7.1 **Date: 31/05/2024**

Register No.: 231401007 **Name: Ananya Sriram**

Binary String

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

```
s=input()
unique_chars = set(s)
if unique_chars == {'0', '1'} or unique_chars == {'0'} or unique_chars == {'1'}:
    print("Yes")
else:
    print("No")
```

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

Passed all tests! ✔

Ex. No. : 7.2 **Date: 31/05/2024**

Register No.: 231401007 **Name: Ananya Sriram**

Check Pair

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

```
def count_distinct_pairs(t,K):
    seen = set()
    distinct_pairs = set()
    for num in t:
        complement = K - num
        if complement in seen:
            pair = (min(num, complement), max(num, complement))
```

```
            distinct_pairs.add(pair)
        seen.add(num)
    return len(distinct_pairs)
t = tuple(map(int, input().split(',')))
K = int(input())
result = count_distinct_pairs(t, K)
print(result)
```

	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! ✔

Ex. No. : 7.3 **Date: 31/05/2024**

Register No.: 231401007 **Name: Ananya Sriram**

DNA Sequence

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 = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"
Output: ["AAAAACCCCC","CCCCCAAAAA"]

Example 2:

Input: s = "AAAAAAAAAAAAA"
Output: ["AAAAAAAAAA"]

For example:

Input	Result
AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA

Program :

```
a=input()
b=[]
for i in range(0,len(a),10):
b.append(a[i:i+10])
print(b[0])
for i in range(len(b)-1):
if(b[i]==b[i+1]):
print(b[i+1][::-1])
```

	Input	Expected	Got	
	AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCCCCCCCAAAAA	AAAAACCCCCCCCCCAAAAA	
	AAAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	

Ex. No. : 7.4 **Date: 31/05/2024**

Register No.: 231401007 **Name: Ananya Sriram**

Print repeated no

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

```
def findDuplicate(nums):
    seen = set()
    for num in nums:
        if num in seen:
            return num
        seen.add(num)
input_str = input()
nums = list(map(int, input_str.split()))
```

result = findDuplicate(nums)

print(result)

	Input	Expected	Got	
✔	1 3 4 4 2	4	4	✔
✔	1 2 2 3 4 5 6 7	2	2	✔

Passed all tests! ✔

Ex. No. : 7.5 **Date: 31/05/2024**

Register No.: 231401007 **Name: Ananya Sriram**

Remove repeated

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

```
def non_rep(arr1, arr2):
    set1 = set(arr1)
    set2 = set(arr2)
    res_set = set1.symmetric_difference(set2)
    if len(res_set) == 0:
        return "NO SUCH ELEMENTS"
    return sorted(res_set), len(res_set)

size1, size2 = map(int, input().split())
arr1 = list(map(int, input().split()))
arr2 = list(map(int, input().split()))

result = non_rep(arr1, arr2)
if result == "NO SUCH ELEMENTS":
    print(result)
else:
    non_rep_elements, count = result
    print(*non_rep_elements)
    print(count)
```

	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! ✔