08 – Tuple/Set

Ex. No. : 8.1 **Date:** 27/4/24

Register No.: 231501024 **Name: ASHWIN T**

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

PROGRAM

```
str=input()
str=[x for x in str]
str=set(str)
c=0
if '0' in str and '1' in str:
    c+=2
if(c==len(str)):
    print("Yes")
else:
    print("No")
```

Output:

	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.

Ex. No. : 8.2 **Date:** 27/4/24

Register No.: 231501024 **Name: ASHWIN T**

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

PROGRAM

```
s=input()
l1=s.split(',')
n=int(input())
l2=[]
for i in range(len(l1)):
    j=i+1
    for j in range(len(l1)): if(int(l1[i])
        +int(l1[j])==n):
            l2.append(list[l1[i],l1[j]])
s=set(l2)

print(len(s)//2)
```

Output:

	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

Ex. No. : 8.3 **Date:** 27/4/24

Register No.: 231501024 **Name: ASHWIN T**

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

```
s = input()
sequence_length = 10
seen = {}
result = []
for i in range(len(s) - sequence_length + 1):
    sequence = s[i:i + sequence_length]
    if sequence in seen:
        seen[sequence] += 1
    else:
        seen[sequence] = 1

for sequence, count in seen.items():
    if count > 1:
        result.append(sequence)
for i in result:
    print(i)
```

Output:

	Input	Expected	Got	
✔	AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCCCCCCCAAAAA	AAAAACCCCCCCCCCAAAAA	✔
✔	AAAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No. : 8.4 **Date:** 27/4/24

Register No.: 231501024 **Name: ASHWIN T**

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

PROGRAM

```
st=input()
nums=st.split()
for i in nums:
    if nums.count(i)==2:
        b=nums.index(i)
print(nums[b])
```

Output:

	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.

Ex. No. : 8.5 **Date:** 27/4/24

Register No.: 231501024 **Name: ASHWIN T**

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

For example:

Input	Result
5 4 1 2 8 6 5 2 6 8 10	1 5 10 3

PROGRAM

```
sizes = list(map(int, input().split()))
arr1 = list(map(int, input().split()))
arr2 = list(map(int, input().split()))
set1 = set(arr1)
set2 = set(arr2)
unique1 = set1 - set2
unique2 = set2 - set1

unique_elements = list(unique1) + list(unique2)

if unique_elements:
    print(" ".join(map(str, unique_elements)))
    print(len(unique_elements))
else:
    print("NO SUCH ELEMENTS")
```

Output:

	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.

Ex. No. : 8.6 **Date:** 27/4/24

Register No.: 231501024 **Name: ASHWIN T**

Malfunctioning Keyboard

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

PROGRAM

```
s1=str(input())
s2=str(input())
l1=(s1.lower()).split(' ')
l2=list(s2)
for i in l2:
    for word in l1:
        if i in list(word):
            l1.remove(word)
print(len(l1))
```

Output:

	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.

Ex. No. : 8.7 **Date:** 27/4/24

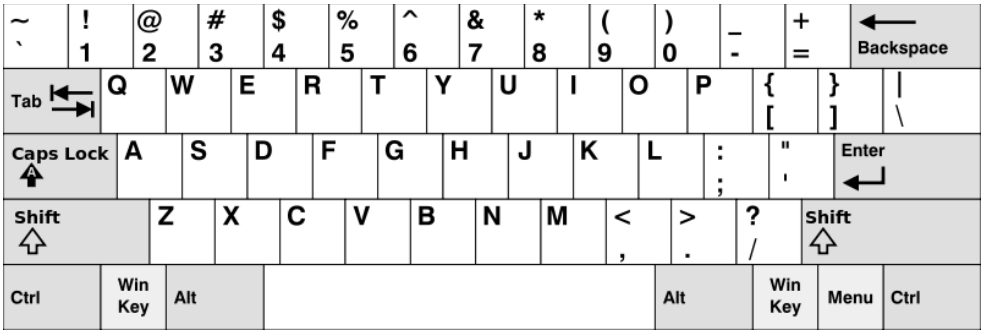
Register No.: 231501024 **Name: ASHWIN T**

American keyboard

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

PROGRAM

```
n=int(input())
l=[]
for j in range(n):
    l.append(str(input()))
row1 = set("qwertyuiop")
row2 = set("asdfghjkl")
row3 = set("zxcvbnm")
result = []
for word in l:
    lower_word = set(word.lower())
    if lower_word <= row1 or lower_word <= row2 or lower_word <= row3:
        result.append(word)
if(len(result)!=0):
    for i in range(len(result)):
        print(result[i])
else:
    print("No  words")
```

Output:

	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.