Ex. No. : 6.9 **Date: 04.05.24**

Register No.: 231901035 **Name: Nitheesh K K**

Merge List

Write a Python program to Zip two given lists of lists.

Input:
m : row size
n: column size
list1 and list 2 : Two lists
Output
Zipped List : List which combined both list1 and list2
Sample test case

Sample input
2
2
1
3
5
7
2
4
6
8

Sample Output

[[1, 3, 2, 4], [5, 7, 6, 8]]

Program:

```
m=int(input())
n=int(input())
l1=[]
l2=[]
c=1
for i in range(0,m*n*2,2):
    a=int(input())
    b=int(input())
    if c%2!=0:
        l1.append(a)
        l1.append(b)
    else:
        l2.append(a)
        l2.append(b)
    c=c+1
l3=[]
l3.append(l1)
l3.append(l2)
print(l3)
```


	Input	Expected	Got	
✔	2 2 1 2 3 4 5 6 7 8	[[1, 2, 5, 6], [3, 4, 7, 8]]	[[1, 2, 5, 6], [3, 4, 7, 8]]	✔

Ex. No. : 6.10 **Date: 04.05.24**

Register No.: 231901035 **Name Nitheesh K K**

Check pair with difference k

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[i] - A[j] = k, i != j.

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn't.

Input
1
3
1
3
5
4
Output:
1
Input
1
3
1
3
5
99
Output
0

For example:

Input	Result
1 3 1 3 5	1

Input	Result
4	
1 3 1 3 5 99	0

Program:

```
t=int(input())
for i in range(0,t):
    n=int(input())
    l=[]
    for j in range(0,n):
        a=int(input())
        l.append(a)
    p=int(input())
    for k in range(0,n):
        c=0
        for m in range(i+1,n):
            if l[m]-l[k]==p:
                c=1
                print('1')
                break
        if c==1:
```

```
        break
if c==0:
    print('0')
```

	Input	Expected	Got	
✔	1 3 1 3 5 4	1	1	✔
✔	1 3 1 3 5 99	0	0	✔

07 – Tuple/Set

Ex. No. : 7.1 **Date: 18.05.24**

Register No.: 231901035 **Name: Nitheesh K K**

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:

```
a = input()
try:
    c = int(a)
    print("Yes")
except:
    print("No")
```


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

Ex. No. : 7.2 **Date: 18.05.24**

Register No.: 231901035 **Name: Nitheesh K K**

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:

```
def findRepeatedSequences(s):
    sequences = {}
    result = []
    for i in range(len(s) - 9):
        seq = s[i:i+10]
        sequences[seq] = sequences.get(seq, 0) + 1
        if sequences[seq] == 2:
            result.append(seq)
    return result
s1 = input()
for i in findRepeatedSequences(s1):
    print(i)
```

	Input	Expected	Got	
✔	AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA	AAAAACCCCC CCCCCAAAAA	✔
✔	AAAAAAAAAAAAA	AAAAAAAAAA	AAAAAAAAAA	✔

Ex. No. : 7.3 **Date: 18.05.24**

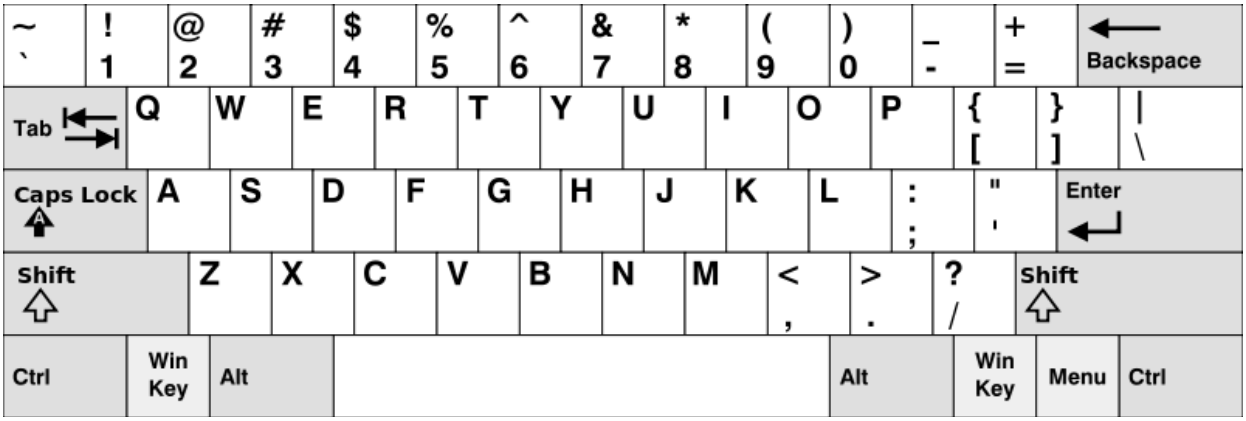
Register No.:231901035 **Name: Nitheesh K K**

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

Input	Result
4 Hello Alaska Dad Peace	Alaska Dad

Program:

```
def findWords(words):
    row1 = set('qwertyuiop')
    row2 = set('asdfghjkl')
    row3 = set('zxcvbnm')

    result = []
    for word in words:
        w = set(word.lower())
        if w.issubset(row1) or w.issubset(row2) or w.issubset(row3):
            result.append(word)
    if len(result) == 0:
        print("No words")
    else:
        for i in result:
            print(i)

a = int(input())
arr = [input() for i in range(a)]
```

findWords(arr)

	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	✔