5ispresentatlocation3.
5ispresent2timesinthearray.

TestCase2

Input

5
67
80
45
97
100
50

Output

50isnotpresentinthearray.

Program:

```
n = int(input())
arr = [int(input()) for _ in range(n)]
element_to_search = int(input())
locations = []
occurrences = 0
for i in range(len(arr)):
    if arr[i] == element_to_search:
        locations.append(i + 1)
        occurrences +=1
if occurrences == 0:
    print(f"{element_to_search} is not present in the array.")
else:
    for loc in locations:
        print(f"{element_to_search} is present at location {loc}.")
```

```
print(f"{element_to_search} is present {occurrences} times in the array.")
```

	Input	Expected	Got	
✔	4 5 6 5 7 5	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	✔
✔	5 67 80 45 97 100 50	50 is not present in the array.	50 is not present in the array.	✔

Ex. No. : 6.8 **Date: 04.05.24**

Register No.: 231901018 **Name: Kavin Sainath S**

Strictly increasing

WriteaPythonprogramtocheckifagivenlistisstrictlyincreasingornot.Moreover,Ifremoving onlyoneelementfromthelistresultsinastrictlyincreasinglist,westillconsiderthelisttrue
Input:
n:Numberofelements
List1:Listofvalues
Output
Print"True"iflistisstrictlyincreasingordecreasingelseprint"False"
SampleTestCase

Input

7

1

2

3

0

4

5

6

Output

True

Program:

```
def check_increasing_or_decreasing(lst):
    increasing = True
    decreasing = True
    for i in range(1, len(lst)):
        if lst[i] > lst[i - 1]:
            decreasing = False
        elif lst[i] < lst[i - 1]:
            increasing = False
    return increasing or decreasing


def check_strictly_increasing_with_removal(lst):
    for i in range(len(lst)):
        temp_lst = lst[:i] + lst[i+1:]
        if check_increasing_or_decreasing(temp_lst):
            return True
    return False
n = int(input())
lst = []
for _ in range(n):
    lst.append(int(input())
if check_increasing_or_decreasing(lst) or check_strictly_increasing_with_removal(lst):
```

```
    print("True")
else:
    print("False")
```

	Input	Expected	Got	
✔	7 1 2 3 0 4 5 6	True	True	✔
✔	4 2 1 0 -1	True	True	✔

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

Register No.: 231901018 **Name: Kavin Sainath S**

Merge List

WriteaPythonprogramtoZiptwogivenlistsoflists.

Input:
m:rowsize
n:columnsize
list1andlist2:Twolists
Output
ZippedList:Listwhichcombinedbothlist1andlist2
Sampletestcase

Sampleinput
2
2
1
3
5
7
2
4
6
8

SampleOutput

[[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.: 231901018 **Name Kavin Sainath S**

Check pair with difference k

GivenanarrayAofsortedintegersandanothernonnegativeintegerk,findifthereexists2indicesi andjsuchthatA[i]-A[j]=k,i!=j.

InputFormat

1. FirstlineisnumberoftestcasesT.FollowingTlinescontain:
2. N,followedbyNintegersofthearray
3. Thenon-negativeintegerk

Outputformat

Print1ifsuchapairexistsand0ifitdoesn't.

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

For example:

Input	Result
1	1

Input	Result
3 1 3 5 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.: 231901018 **Name: Kavin Sainath S**

Binary String

Codershereisasimpletaskforyou,Givenstringstr.Yourtaskistocheckwhetheritisabinarystring ornotbyusingpython set.

Examples:

Input:str="01010101010"

Output:Yes

Input:str="REC101"

Output:No

For example:

Input	Result
01010101010	Yes
01010110101	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.: 231901018 **Name: Kavin Sainath S**

DNA Sequence

The **DNA sequence** iscomposedofaseriesofnucleotidesabbreviatedas 'A', 'C', 'G',and 'T'.

Forexample, "ACGAATTCCG" isa **DNA sequence**.

Whenstudying **DNA**,itisusefultoidentifyrepeatedsequenceswithintheDNA.

Givenastring s thatrepresentsa **DNA sequence**,returnallthe **10-letter-long** sequences (substrings)thatoccurmorethanonceinaDNAmolecule.Youmayreturntheanswerin **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**

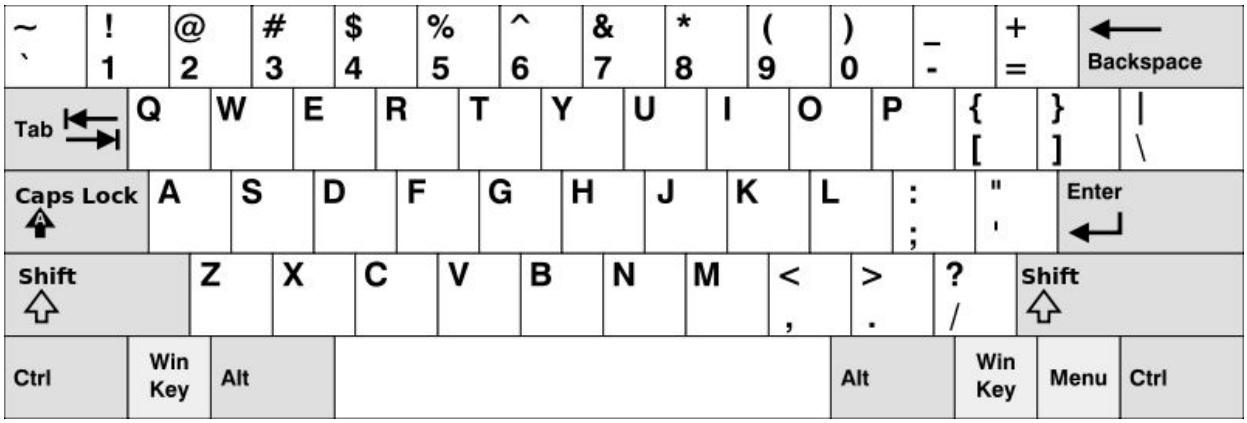
Register No.:231901018 **Name: Kavin Sainath S**

American keyboard

Givenanarrayofstrings words,return *the words that can be typed using letters of the alphabet on only one row of American keyboard like the image below.*

Inthe **American keyboard**:

- thefirstrowconsistsofthecharacters "qwertyuiop",
- thesecondrowconsistsofthecharacters "asdfghjkl",and
- thethirdrowconsistsofthecharacters "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:

```
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	✔