**Ex. No. : 6.7 Date: 04.05.24**

**Register No.:231801123 Name: Pavithra S**

**Location**

Write a program to print all the locations at which a particular element (taken as input) is found in a list and also print the total number of times it occurs in the list. The location starts from 1.

For example, if there are 4 elements in the array:

5

6

5

7

If the element to search is 5 then the output will be:

5 is present at location 1

5 is present at location 3

5 is present 2 times in the array.

Sample Test Cases

Test Case 1

Input

4

5

6

5

7

5

Output

5 is present at location 1.

5 is present at location 3.

5 is present 2 times in the array.

Test Case 2

Input

5

67

80

45

97

100

50

Output

50 is not present in the array.

**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.")

****

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

**Register No.: 231801123 Name: Pavithra S**

**Strictly increasing**

Write a Python program to check if a given list is strictly increasing or not. Moreover, If removing only one element from the list results in a strictly increasing list, we still consider the list true

Input:

n : Number of elements

List1: List of values

Output

Print "True" if list is strictly increasing or decreasing else print "False"

Sample Test Case

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")

****

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

**Register No.: 231801123 Name: Pavithra S**

**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)

****

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

**Register No.: 231801123 Name: Pavithra S**

**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  4 | 1 |
| 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')

