05 -Listin Python

Ex. No. : 5.1 **Date: 28-03-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

BalancedArray

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal.The array may not be reordered.

Example
arr=[1,2,3,4,6]

· thesumofthe firstthree elements,1+2+3=6.Thevalueofthelastelementis6.
· Usingzerobasedindexing,arr[3]=4isthepivotbetweenthetwosubarrays.
· Theindexofthepivotis3.

Constraints

· $3 \leq n \leq 10^5$
· $1 \leq arr[i] \leq 2 \times 10^4$,where$0 \leq i < n$
· Itisguaranteedthatasolutionalwaysexists.

Thefirstlinecontainsanintegern,thesizeofthearrayarr.
Eachofthenextnlinescontainsan integer,arr[i],where$0 \leq i < n$. Sample Case
0
SampleInput0
4
1
2
3
3
SampleOutput0
2
Explanation0
Thesumofthefirsttwoelements,1+2=3.The value ofthelastelementis3 Using zerobased indexing,arr[2]=3 isthepivot betweenthetwosubarrays The index of the pivot is 2
SampleCase1
SampleInput1
3
1
2
1
SampleOutput1

1
Explanation1
Thefirstandlastelementsareequalto1
Usingzerobasedindexing,arr[1]=2isthepivotbetweenthetwosubarrays The index of the pivot is 1.

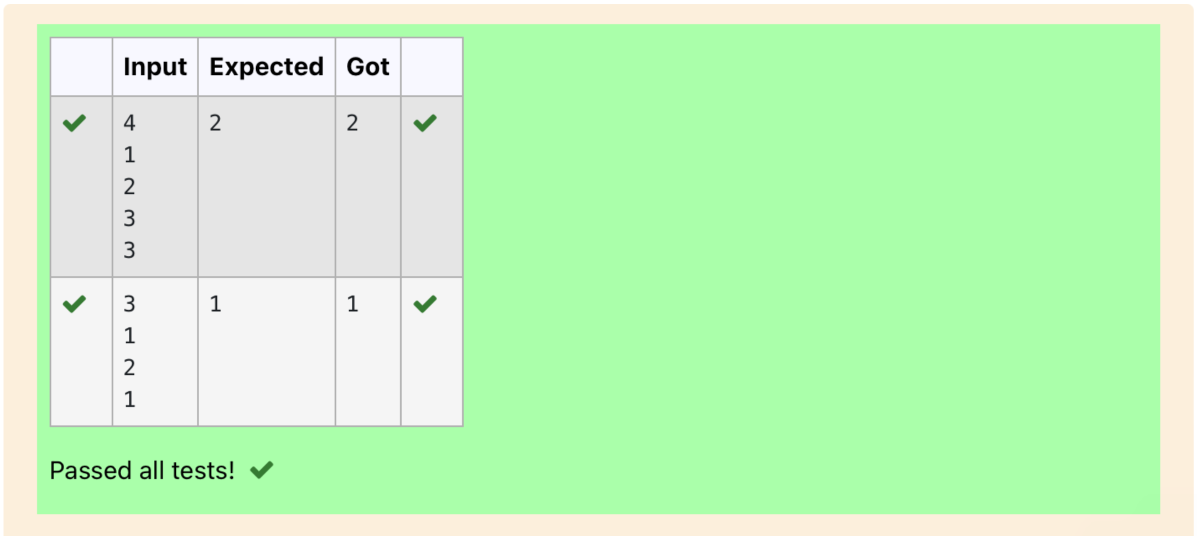
For example:

Input	Result
4 1 2 3 3	2
3 1 2 1	1

Program:

```
a=int(input()) l=[]

foriinrange(a):

    c=int(input())

    l.append(c)

foriinrange(1,a):

    d=sum(l[0:i])

    r=sum(l[i+1:])

    if(d==r):

        print(i)
```

Output:



	Input	Expected	Got	
✓	4 1 2 3 3	2	2	✓
✓	3 1 2 1	1	1	✓

Passed all tests! ✓

Ex. No. : 5.2 **Date: 28-03-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

Checkpairwithdifference k

GivenanarrayAofsortedintegersandanothernonnegativeintegerk,findifthere exists 2 indices i and j such that A[i] - A[j] = k, i != j.

Input Format

1. Firstlineis numberoftestcasesT.FollowingTlines contain:
2. N,followedbyNintegersofthearray
3. Thenon-negativeintegerk

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:

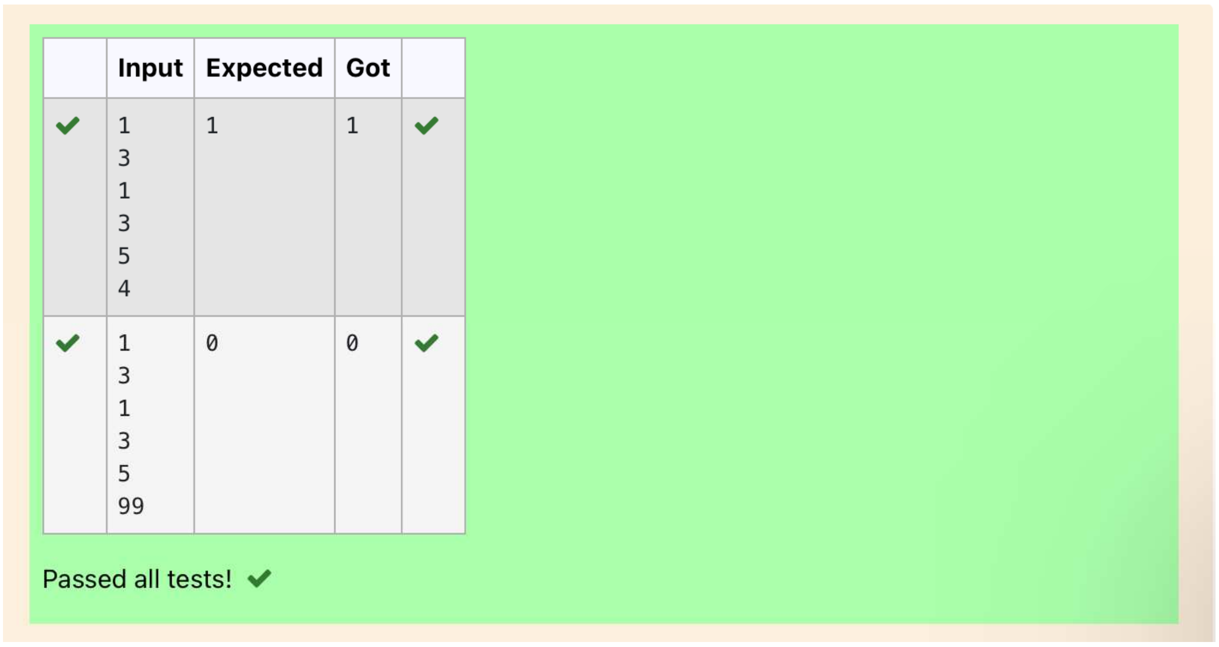
Input	Result
1 3 1 3 5 4	1
Input	**Result**

1 3 1 3 5 99	0

Program:

```
a=int(input())
while(a!=0):
    b=int(input())
    l=[]
    f=0
    foriinrange(b):
        c=int(input())
        l.append(c)
    k=int(input())
    a-=1
    for i in range(b):
        forjinrange(b):
            if(l[i]-l[j]==kandi!=j): f=1
                break
    if(f==1):
        print(1) else:
        print(0)
```

Output:



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

Passed all tests! ✓

Ex. No. : 5.3 **Date: 28-03-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

CountElements

Complete the program to count frequency of each element of an array. Frequency of a particular element will be printed once.

SampleTestCases Test

Case 1

Input 7

23

45

23

56

45

23

40

Output

23occurs3times

45occurs2times

56occurs1times

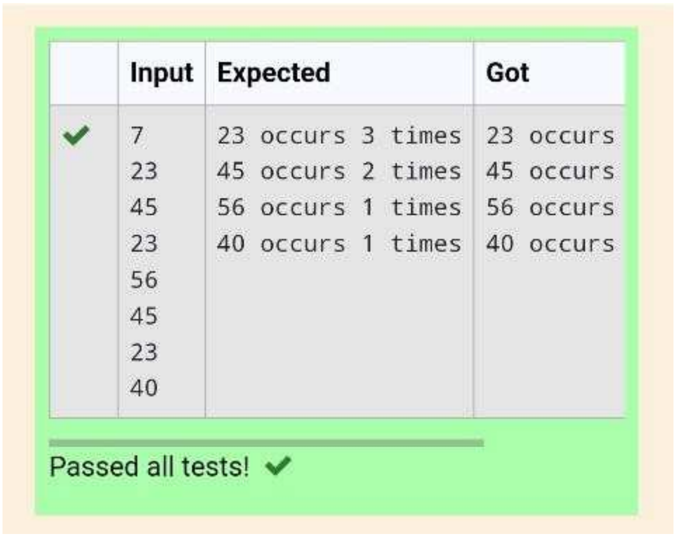
40occurs1times

Program:

```
importcollections
defCountFrequency(arr):
	returncollections.Counter(arr)
```

```
ifname=="main":  #  Input  size
          of array n = int(input())

          #Inputelementsinarray arr = []
          for_in range(n):
          ele=int(input())
          arr.append(ele)

          #Calculatefrequencyofeachelement freq =
          CountFrequency(arr)


          for key, value in freq.items():
          print(f"{key}occurs{value}times")
```

Output:



	Input	Expected	Got
✓	7 23 45 23 56 45 23 40	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 40 occurs 1 times	23 occurs 45 occurs 56 occurs 40 occurs

Passed all tests! ✓

Ex. No. : 5.4 Date: 28-03-2024

RegisterNo.: 2116231501072 Name: Kanishka P

DistinctElementsinanArray

Programtoprintallthedistinctelementsinanarray.Distinctelementsarenothingbut the unique (non-duplicate) elements present in the given array.

InputFormat:

FirstlinetakeanIntegerinputfromstdinwhichisarraylengthn. Second line take n Integers which is inputs of array.

OutputFormat:

PrinttheDistinctElementsinArrayinsinglelinewhichisspaceSeparated

ExampleInput: 5
1
2
2
3
4
Output:
1234
ExampleInput: 6
1
1
2
2
3
3
Output:
123

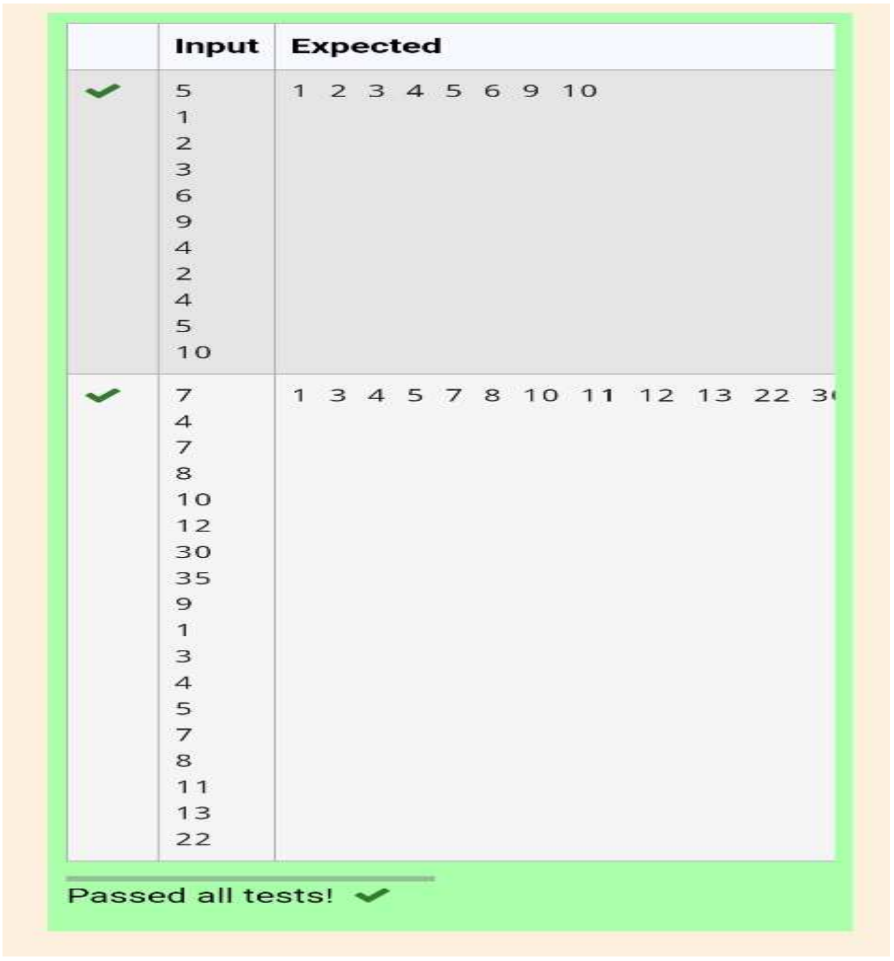
Forexample:
Input Result 5
1
2
2
3
4
1234
6
1

```
1
2
2
3
3
123
```

Program:

```
defmerge_arrays_without_duplicates(arr1,arr2):
    #Combinethearraysandconverttoasettoremoveduplicates result_set = set(arr1
    + arr2)
    # Convert the set back to a sorted list
    merged_sorted_array=sorted(result_set) return
    merged_sorted_array
#Inputreadandprocessing def
process_input():
    #Readingnumberofelementsandtheelementsforthefirstarray n1 = int(input())
    array1=[]
    for _ in range(n1):element =
        int(input())
        array1.append(element)
    #Readingnumberofelementsandtheelementsforthesecondarray n2 = int(input())
    array2=[]
    for _ in range(n2):
        element=int(input())
        array2.append(element)
    #Mergethearrayswithoutduplicates
    result=merge_arrays_without_duplicates(array1,array2) # Print the
    result
    print("".join(map(str,result)))
```

Output:



	Input	Expected
✓	5 1 2 3 6 9 4 2 4 5 10	1 2 3 4 5 6 9 10
✓	7 4 7 8 10 12 30 35 9 1 3 4 5 7 8 11 13 22	1 3 4 5 7 8 10 11 12 13 22 3

Passed all tests! ✓

Ex. No. : 5.5 **Date: 28-03-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

ElementInsertion

Consideraprogramtoinsertanelement/iteminthesortedarray.Completethelogicby filling up required code in editable section. Consider an array of size 10. The eleventh item is the data is to be inserted.

SampleTestCases Test Case 1
Input

```
1
3
4
5
6
7
8
9
10
11
2
```

Output

```
ITEM to be inserted:2
Afterinsertionarrayis: 1
2
3
4
5
6
7
8
9
10
11
```

TestCase2 Input

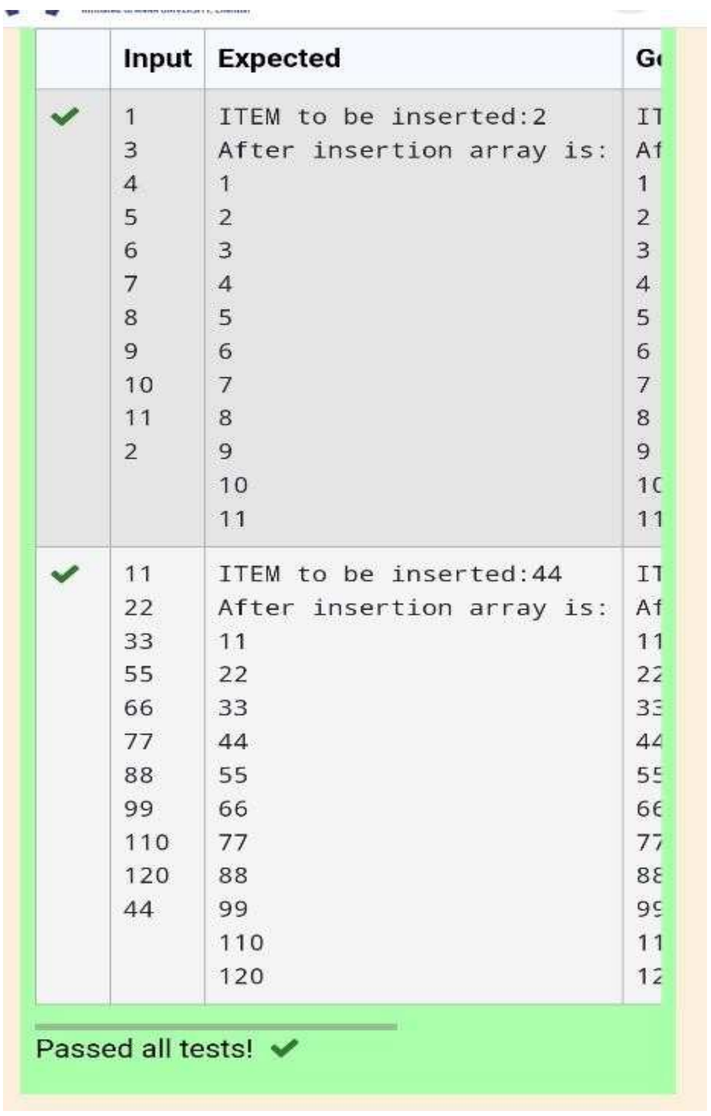
```
11
22
33
55
66
77
88
99
110
120
44
```

Output

```
ITEM to be inserted:44
Afterinsertionarrayis: 11
22
33
44
55
66
77
88
99
110
120
```

Program: Output:

```
definsert_sorted(list,n):

    list.append(n)

    sorted_list=sorted(list)

    print("Afterinsertionarrayis:") for i in

    range(11):

        print(sorted_list[i])


sorted_list=[int(input())foriin
range(10)]

new_element=int(input())

print("ITEMtobeinserted:",
new_element, sep='')

insert_sorted(sorted_list,
new_element)
```



	Input	Expected	Gc
✓	1 3 4 5 6 7 8 9 10 11 2	ITEM to be inserted:2 After insertion array is: 1 2 3 4 5 6 7 8 9 10 11	IT Af 1 2 3 4 5 6 7 8 9 1(11
✓	11 22 33 55 66 77 88 99 110 120 44	ITEM to be inserted:44 After insertion array is: 11 22 33 44 55 66 77 88 99 110 120	IT Af 11 22 33 44 55 66 77 88 99 11 12

Passed all tests! ✓

Ex. No. : 5.6 **Date: 28-03-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

FindtheFactor

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the pthelement of thelist, sorted ascending. If there is no pthelement, return 0.

Constraints

$1 \le n \le 10^{15}$

$1 \le p \le 10^{9}$

Thefirstlinecontainsanintegern,thenumbertofactor.

Thesecondlinecontainsanintegerp,the1-basedindexofthefactortoreturn.

SampleCase0
SampleInput0

10

3

SampleOutput0

5

Explanation0

Factoringn=10results in{1,2, 5,10}.Returnthep =3rdfactor,5,asthe answer.

SampleCase1
SampleInput1

10

5

SampleOutput1

0

Explanation1

Factoringn=10results in{1,2, 5,10}.There areonly4factorsandp=5, therefore 0 is returned as the answer.

SampleCase2
SampleInput2

1

1

SampleOutput2

1

Explanation2

Factoringn=1resultsin {1}.Thep=1stfactorof1is returnedasthe answer.

For example:

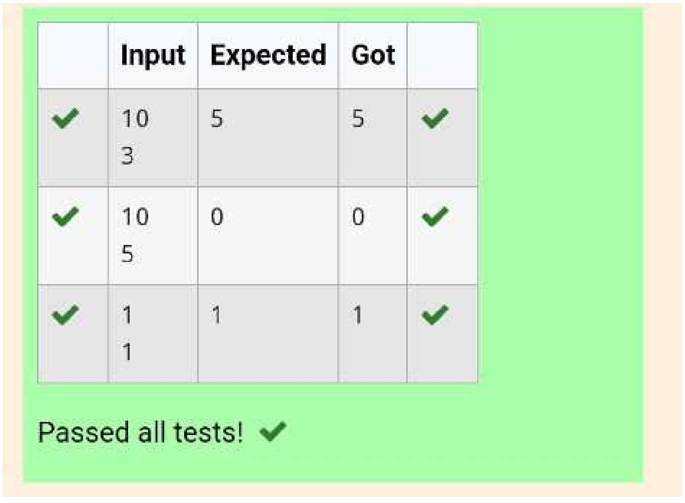
Input	Result
10 3	5
10 5	0
1 1	1

Program:

```
import sys

importmath


deffind_factors(n):

    factors=[]

    foriinrange(1,int(math.sqrt(n))+1): if n % i

    == 0:

    factors.append(i) if

    i != n // i:

        factors.append(n//i) return

    sorted(factors)


def get_pth_factor(n, p):

    factors=find_factors(n) if p

    <= len(factors): return

    factors[p - 1] else:

    return 0
```

```
#Readinginputdirectly fromthestandardinput(typicallyforcompetitive programming)
input=sys.stdin.read data
= input().split()n =
int(data[0])
p=int(data[1])

#Calculateandprintthep-thfactor
print(get_pth_factor(n, p))
```

Output:



	Input	Expected	Got	
✓	10 3	5	5	✓
✓	10 5	0	0	✓
✓	1 1	1	1	✓

Passed all tests! ✓

Ex. No. : 5.7 **Date: 28-03-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

MergeList

WriteaPythonprogramtoZiptwogivenlistsoflists.

Input:
m : row
sizen:columnsiz
e
list1andlist2:Twolists Output
ZippedList:Listwhichcombinedbothlist1andlist2 Sample test case

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

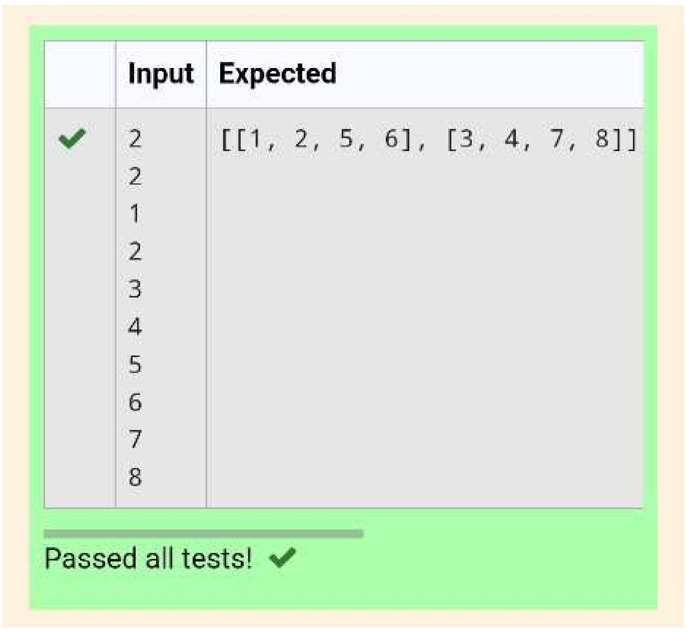
Sample Output

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

Program:

```
defzip_lists(list1,list2):

    return[row1+row2forrow1,row2inzip(list1,list2)]


defmain():

    m=int(input())
    n = int(input())



    list1=[[int(input())for_inrange(n)]for_inrange(m)]
    list2=[[int(input())for_inrange(n)]for_inrange(m)]

    zipped_list=zip_lists(list1,list2) print(zipped_list)

if name______=="main":
    main()
```

Output:



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

Passed all tests! ✓

Ex. No. **:** **5.8** **Date:**

RegisterNo.: **Name:**

MergeTwoSortedArraysWithoutDuplication

Outputisamergedarraywithoutduplicates. Input Format
N1-noofelementsinarray1 Array
elements for array 1 N2-
noofelementsinarray2 Array
elements for array2 Output
Format
Displaythemergedarray Sample

Input 1

```
5
1
2
3
6
9
4
2
4
5
10
```

SampleOutput1

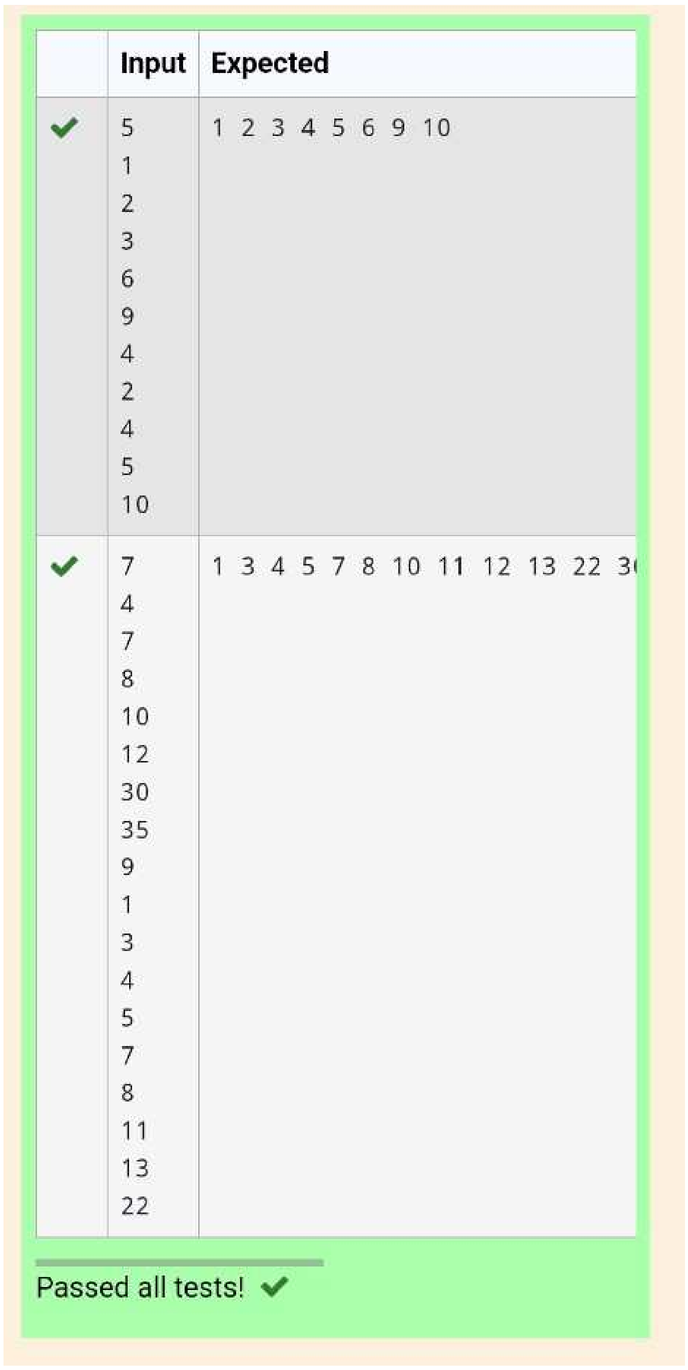
```
123456910
```

Program:

```
defmerge_arrays_without_duplicates(arr1,arr2):
    #Combinethearraysandconverttoasettoremoveduplicates result_set = set(arr1 +
    arr2)
    # Convert the set back to a sorted list
    merged_sorted_array=sorted(result_set) return
    merged_sorted_array


#Inputreadandprocessing
```

```
defprocess_input():
    #Readingnumberofelementsandtheelementsforthefirstarray n1 = int(input())
    array1=[]
    for _ in range(n1):element =
        int(input())
        array1.append(element)

    #Readingnumberofelementsandtheelementsforthesecondarray n2 = int(input())
    array2=[]
    for _ in range(n2):element =
        int(input())
        array2.append(element)

    #Mergethearrayswithoutduplicates
    result=merge_arrays_without_duplicates(array1,array2)

    #Printthe result
    print("".join(map(str,result)))
```

Output:



	Input	Expected
✓	5 1 2 3 6 9 4 2 4 5 10	1 2 3 4 5 6 9 10
✓	7 4 7 8 10 12 30 35 9 1 3 4 5 7 8 11 13 22	1 3 4 5 7 8 10 11 12 13 22 3

Passed all tests! ✓

Ex. No. : 5.9 **Date: 28-03-2024**

RegisterNo.: 2116231501072 **Name: Kanishka P**

PrintElementLocation

Writeaprogramtoprintallthelocationsatwhichaparticularelement(takenas input) is found in a list and also print the total number of times it occurs in the list. The location starts from 1.

Forexample,ifthereare4elementsinthearray: 5
6
5
7

Iftheelementtosearchis5thentheoutputwillbe: 5 is present at location 1
5ispresentatlocation3
5ispresent2timesinthearray. Sample Test Cases

TestCase1
Input

```
4
5
6
5
7
5
```

Output

```
5ispresentatlocation1.
5ispresentatlocation3.
5ispresent2timesinthearray.
```

TestCase2
Input

```
5
67
80
45
97
100
50
```

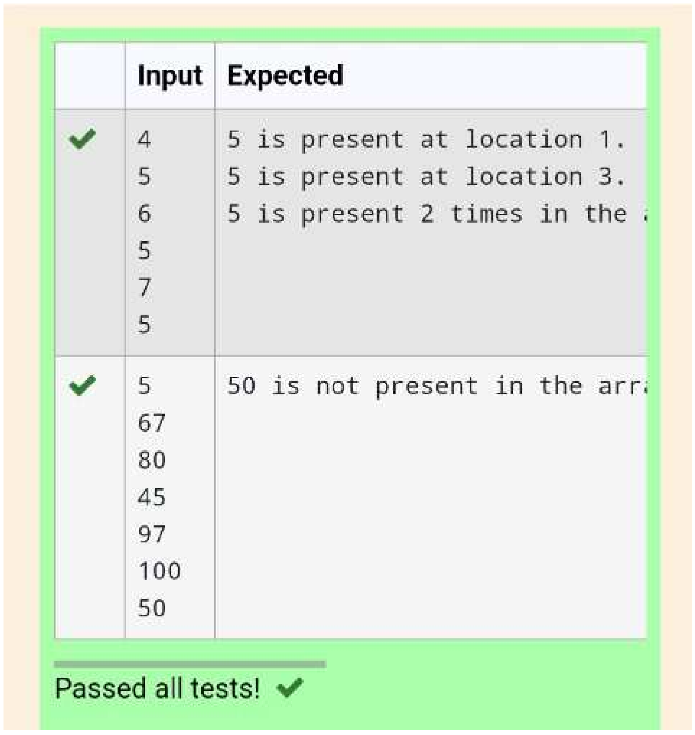
Output

```
50isnotpresentinthearray.
```

Program:

```
deffind_element_locations(lst,target): locations = []
   count=0
   foriinrange(len(lst)): if
      lst[i] == target:
        locations.append(i+1)
        count+=1

   returnlocations,count

defmain():
   n = int(input())
   lst=[int(input())for_inrange(n)] target =
   int(input())

   locations,count=find_element_locations(lst,target)

   ifcount== 0:
      print(f"{target}isnotpresentinthearray.") else:
      forlocinlocations:
        print(f"{target} is present at location {loc}.")
      print(f"{target}ispresent{count}timesinthearray.")

if name______=="main":
   main()
```

Output:



	Input	Expected
✓	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
✓	5 67 80 45 97 100 50	50 is not present in the arr

Passed all tests! ✓

Ex. No.   :   5.10   **Date: 28-03-2024**

RegisterNo.: 2116231501072   **Name: Kanishka P**

<u>Strictlyincreasing</u>

WriteaPythonprogramtocheckifagivenlistisstrictlyincreasingornot.Moreover,
Ifremovingonlyoneelementfromthelistresultsinastrictlyincreasinglist,westill consider the list true
Input:
n:Numberofelements List1:
List of values Output
Print"True"iflistisstrictlyincreasingordecreasingelseprint"False"

SampleTestCase Input

7

1

2

3

0

4

5

6

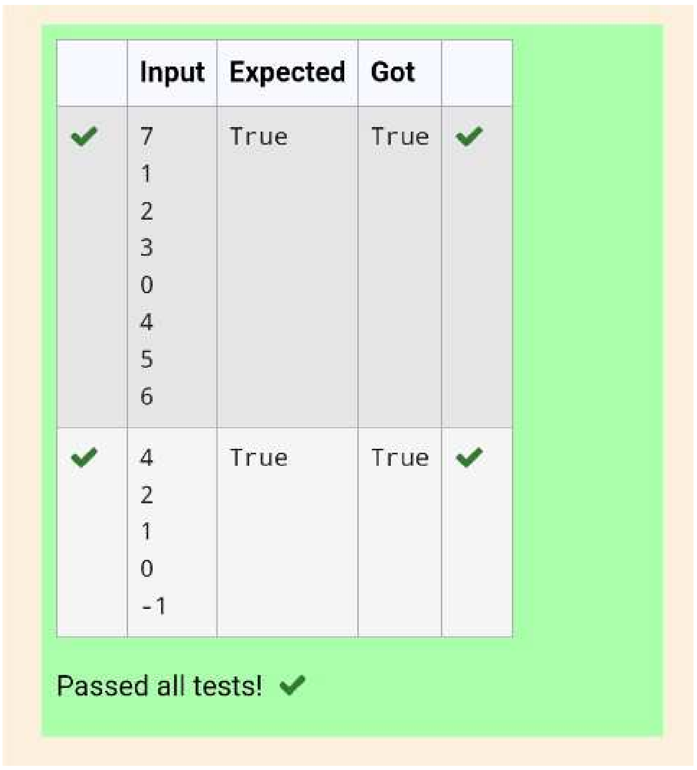
Output True

Program:

```
n=int(input())

arr=[int(input())foriinrange(n)] l =

arr.copy()
```

```
g=0
size = len(arr)
arr_asc=sorted(arr)
arr_des=sorted(arr)[::-1]
ifarr==arr_ascorarr==arr_des: print('True')
   g=1 else:
   for i in arr: l.remove(i)
      arr_asc.remove(i)
      arr_des.remove(i)
      ifl==arr_ascorl==arr_des: print('True')
        g=1
        break
      l=arr.copy()
      arr_asc = sorted(arr)
      arr_des=sorted(arr)[::-1]
if g==0:
   print('False')
```

Output:



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

Passed all tests! ✓