**WEEK-6 CODING**

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

Sample Test Cases

Test Case 1

Input

7

23

45

23

56

45

23

40

Output

23 occurs 3 times

45 occurs 2 times

56 occurs 1 times

40 occurs 1 times

**PROGRAM:**

n = int(input().strip())

arr = []

for \_ in range(n):

arr.append(int(input().strip()))

f = {}

for num in arr:

if num in f:

f[num] += 1

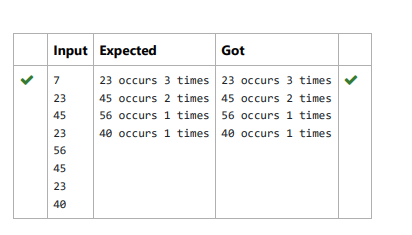
else:

f[num] = 1

for num, freq in f.items():

print(f"{num} occurs {freq} times")

**OUTPUT:**

****

2. 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:**

n=int(input(""))

list1=[int(input()) for \_ in range(n)]

def is\_strictly\_increasing(lst):

count=0

for i in range(1, len(lst)):

if lst[i] < lst[i-1]:

count +=1

if count > 1:

return False

if i==1 or lst[i] > lst[i-2]:

continue

elif i<len(lst)-1 and lst[i+1]>lst[i-1]:

continue

else:

return False

return True

def is\_strictly\_decreasing(lst):

reversed\_lst=lst[::-1]

return is\_strictly\_increasing(reversed\_lst)

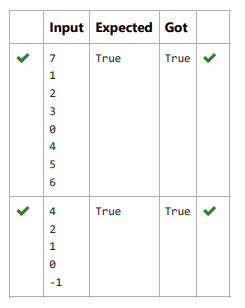
if is\_strictly\_increasing(list1) or is\_strictly\_decreasing(list1):

print("True")

else:

print("False")

**OUTPUT:**



3.Consider a program to insert an element / item in the sorted array. Complete the logic by filling up required code in editable

section. Consider an array of size 10. The eleventh item is the data is to be inserted.

Sample Test Cases

Test Case 1

Input

1

3

4

5

6

7

8

9

10

11

2

Output

ITEM to be inserted:2

After insertion array is:

1

2

3

4

5

6

7

8

9

10

11

Test Case 2

Input

11

22

33

55

66

77

88

99

110

120

44

Output

ITEM to be inserted:44

After insertion array is:

11

22

33

4455

66

77

88

99

110

120

**PROGRAM:**

l=[]

for i in range(0,10):

e=int(input())

l.append(e)

a=int(input())

print("ITEM to be inserted:{:d}".format(a))

print("After insertion array is:")

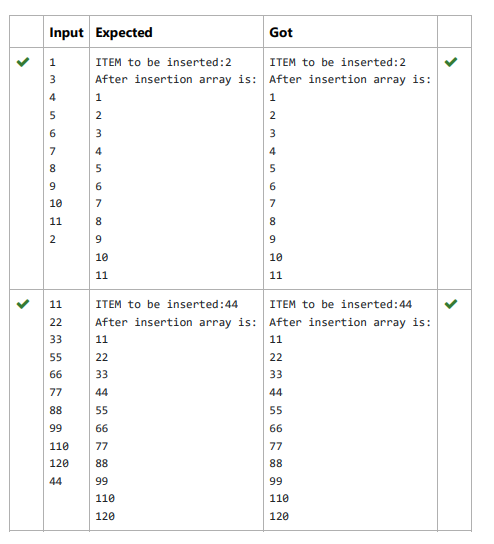
l.append(a)

l.sort()

for j in range(0,11):

print(l[j])

**OUTPUT:**

****

4. 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]

· the sum of the first three elements, 1+2+3=6. The value of the last element is 6.

· Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.

· The index of the pivot is 3.

Constraints

· 3 ≤ n ≤ 10

· 1 ≤ arr[i] ≤ 2 × 10 , where 0 ≤ i < n

· It is guaranteed that a solution always exists.

The first line contains an integer n, the size of the array arr.

Each of the next n lines contains an integer, arr[i], where 0 ≤ i < n.

Sample Case 0

Sample Input 0

4

1

2

3

3

Sample Output 0

2

Explanation 0

· The sum of the first two elements, 1+2=3. The value of the last element is 3.

· Using zero based indexing, arr[2]=3 is the pivot between the two subarrays.

· The index of the pivot is 2.

Sample Case 1

Sample Input 1

3

1

2

1

Sample Output 1

1

Explanation 1

· The first and last elements are equal to 1.

· Using zero based indexing, arr[1]=2 is the pivot between the two subarrays.

· The index of the pivot is 1.

**PROGRAM:**

def find\_pivot\_index(arr):

total\_sum = sum(arr)

left\_sum = 0

for i, num in enumerate(arr):

total\_sum -= num

if left\_sum == total\_sum:

return i

left\_sum += num

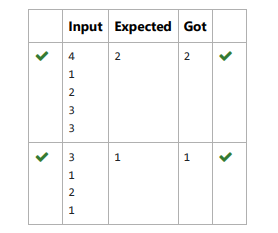
return -1

n = int(input())

arr = [int(input()) for \_ in range(n)]

print(find\_pivot\_index(arr))

**OUTPUT:**



5. Output is a merged array without duplicates.

Input Format

N1 - no of elements in array 1

Array elements for array 1

N2 - no of elements in array 2

Array elements for array2

Output Format

Display the merged array

Sample Input 1

5

1

2

3

6

9

4

2

4

5

10

Sample Output 1

1 2 3 4 5 6 9 10

**PROGRAM:**

N1 = int(input().strip())

arr1 = set()

for \_ in range(N1):

arr1.add(int(input().strip()))

N2 = int(input().strip())

arr2 = set()

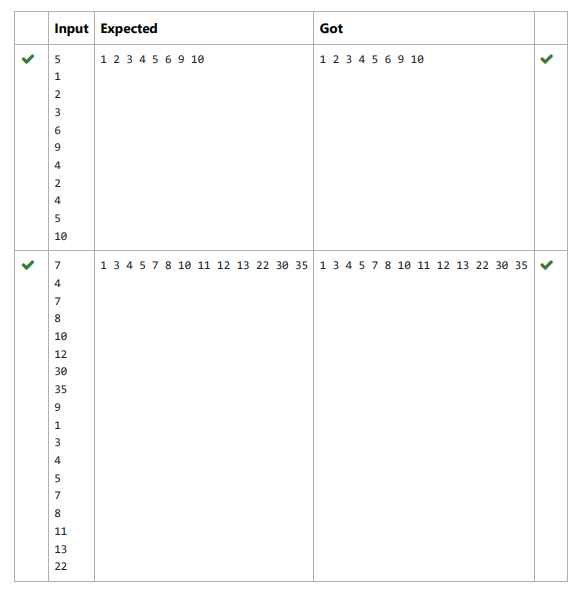
for \_ in range(N2):

arr2.add(int(input().strip()))

merged\_array = sorted(arr1.union(arr2))

print(\*merged\_array)

**OUTPUT:**

****