**Week6:**

1. 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.

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.

**PROGRAM:**

n=int(input())

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

element\_to\_search=int(input())

locations=[]

occurrence=0

for i in range(len(arr)):

if arr[i]==element\_to\_search:

locations.append(i+1)

occurrence+=1

if occurrence==0:

print(f"{element to search} is not present in the array "

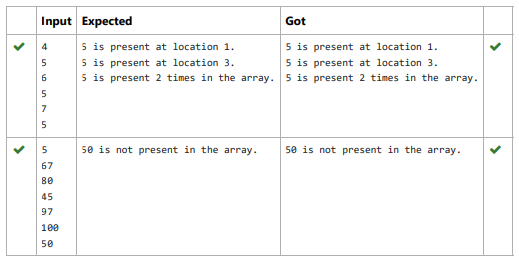
else:

for loc in locations:

print(f"{element\_to\_search} is present at location {l

print(f"{element\_to\_search} is present {occurrence} times

**OUTPUT:**

****

2. Output is a merged array without duplicates.

**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())

l1=[]

for i in range(0,n1):

a=int(input())

l1.append(a)

n2=int(input())

l2=[]

for i in range(0,n2):

a=int(input())

l2.append(a)

l3=[]

l3.extend(l1)

l3.extend(l2)

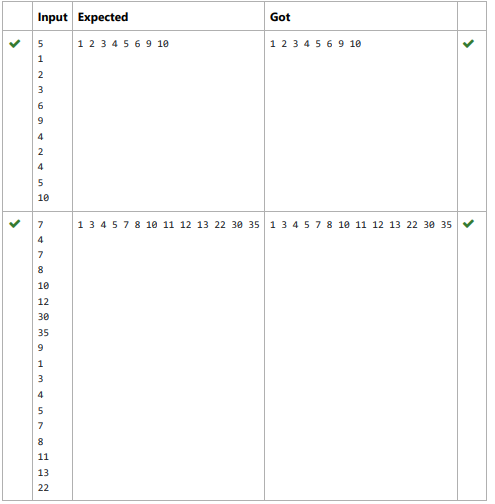
a=list(set(l3))

a.sort()

for i in a:

print(i,end=' ')

**OUTPUT:**

****

3. 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())

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

frequency={}

for num in arr:

if num in frequency:

frequency[num]+=1

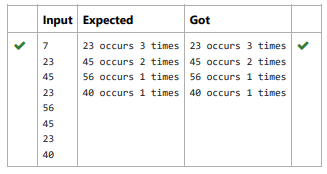
else:

frequency[num]=1

for num,freq in frequency.items():

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

**OUTPUT:**



4. Find the intersection of two sorted arrays.

**Input:**

1

3 10 17 57

6 2 7 10 15 57 246

**Output:**

10 57

**PROGRAM:**

t=int(input())

l1=list()

while(t!=0):

n1=int(input())

l1=[]

l2=[]

for i in range(0,n1):

a=int(input())

l1.append(a)

n2=int(input())

for i in range(0,n2):

a=int(input())

l2.append(a)

t=t-1

c=set(l1)

d=set(l2)

e=list(c.intersection(d))

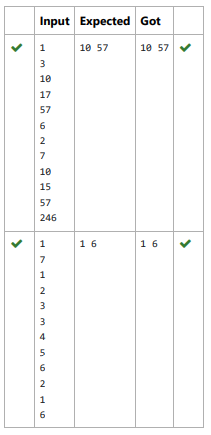
e.sort()

for i in e:

print(i,end=' ')

print('\n')

**OUTPUT:**

****

5. Given two lists A and B, and B is an anagram of A. B is an anagram of A means B is made by randomizing the order of the elements in A.

We want to find an index mapping P, from A to B. A mapping P[i] = j means the ith element in A appears in B at index j.

These lists A and B may contain duplicates. If there are multiple answers, output any of them.

**Sample Input:**

5

12 28 46 32 50

50 12 32 46 28

**Sample Output:**

1 4 3 2 0  
  
**PROGRAM:**

def index\_mapping(A, B):

index\_map = {num: i for i, num in enumerate(B)}

return ' '.join(str(index\_map[num]) for num in A)

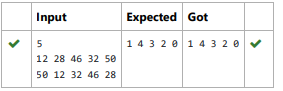
n = int(input())

A = list(map(int, input().split()))

B = list(map(int, input().split()))

print(index\_mapping(A, B))

**OUTPUT:**

****

6. 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

**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:")

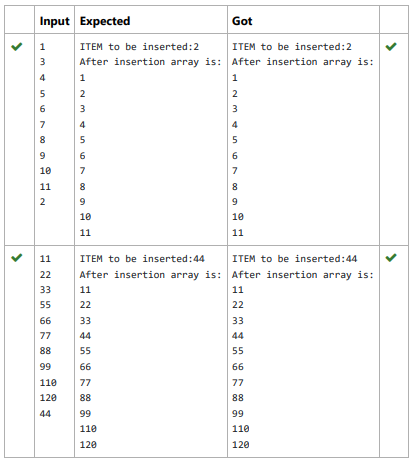
l.append(a)

l.sort()

for j in range(0,11):

print(l[j])

**OUTPUT:**

****

7. 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 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

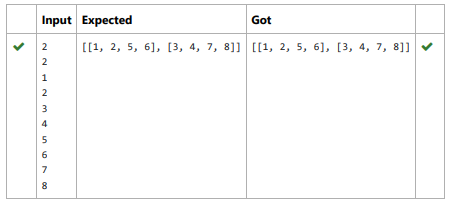
l3=[]

l3.append(l1)

l3.append(l2)

print(l3)

**OUTPUT:**

****

8. 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.

**Sample Input 0**

4

1

2

3

3

**Sample Output 0**

2

**Sample Case 1**

**Sample Input 1**

3

1

2

1

**Sample Output 1**

1

**PROGRAM:**

a=int(input())

b=[]

for i in range(a):

element=int(input())

b.append(element)

total=sum(b)

left=0

right=total-b[0]

if left==right:

print(0)

exit()

for i in range(1,a):

left+=b[i-1]

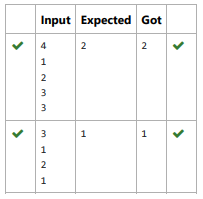
right-=b[i]

if left==right:

print(i)

break

**OUTPUT:**

****

9. 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()))

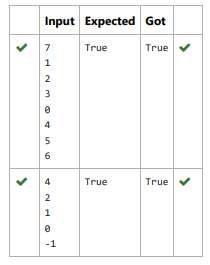
if check\_increasing\_or\_decreasing(lst) or check\_strictly\_in

print("True")

else:

print("False")

**OUTPUT:**

****

10. Program to print all the distinct elements in an array. Distinct elements are nothing but the unique (non-duplicate) elements present in the given array.

**Example Input:**

5

1

2

2

3

4

**Output:**

1 2 3 4

**PROGRAM:**

n=int(input())

arr=[]

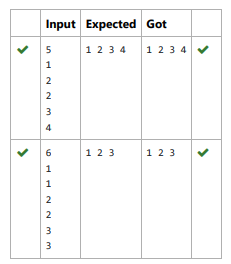
for \_ in range(n):

arr.append(int(input()))

distinct\_element=set(arr)

print(\*distinct element)

**OUTPUT:**

****