WEEK 6:

1.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

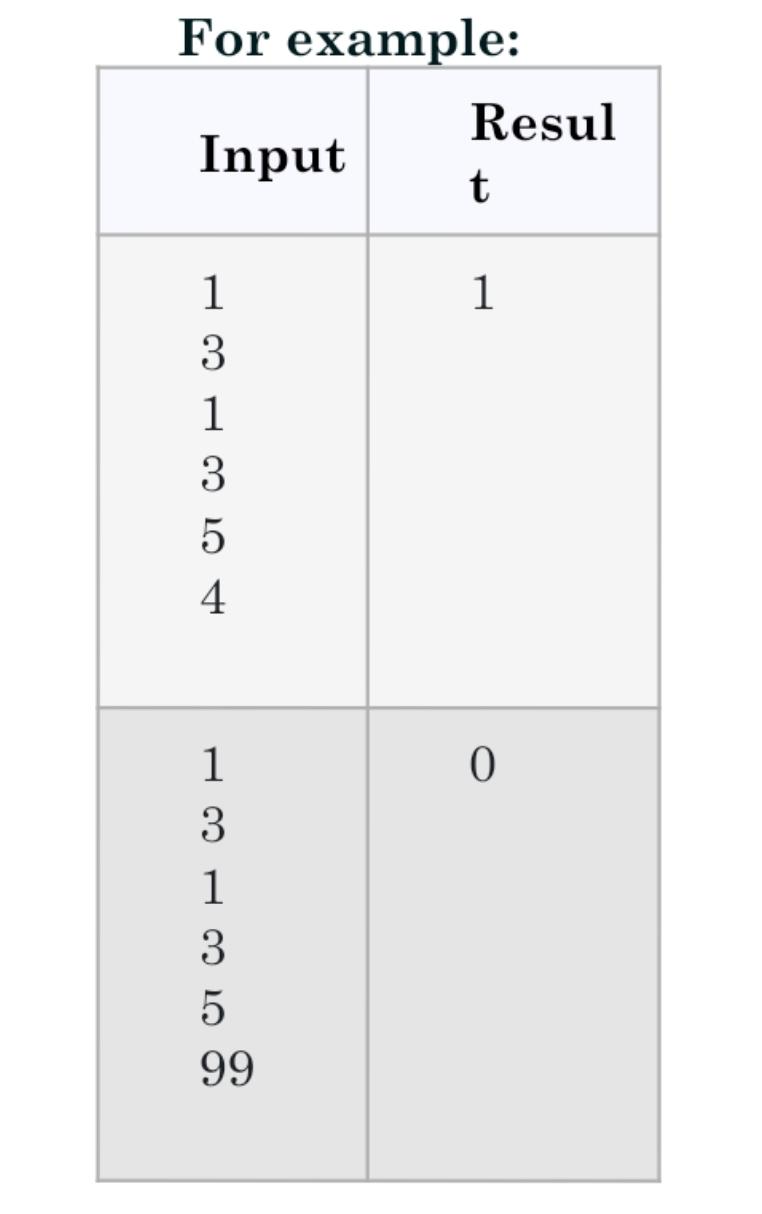
1. First line is number of test cases T. Following T lines contain:

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.



**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

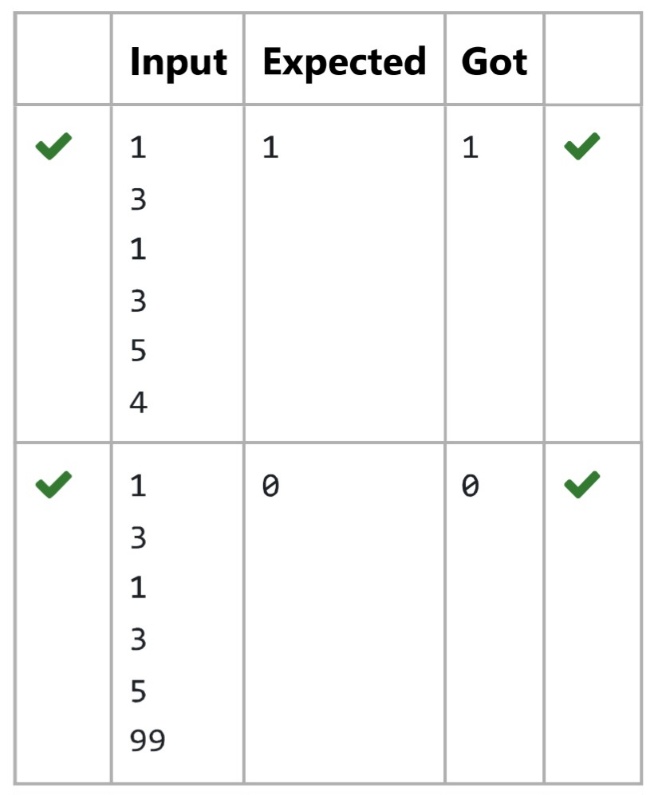
if c == 1:

break

if c == 0:

Pass

**Output:**

****

2. 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

44

55

66

77

88

99

110

120

**Program:**

my\_list = []

for i in range(0, 10):

e = int(input())

my\_list.append(e)

a = int(input())

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

my\_list.append(a)

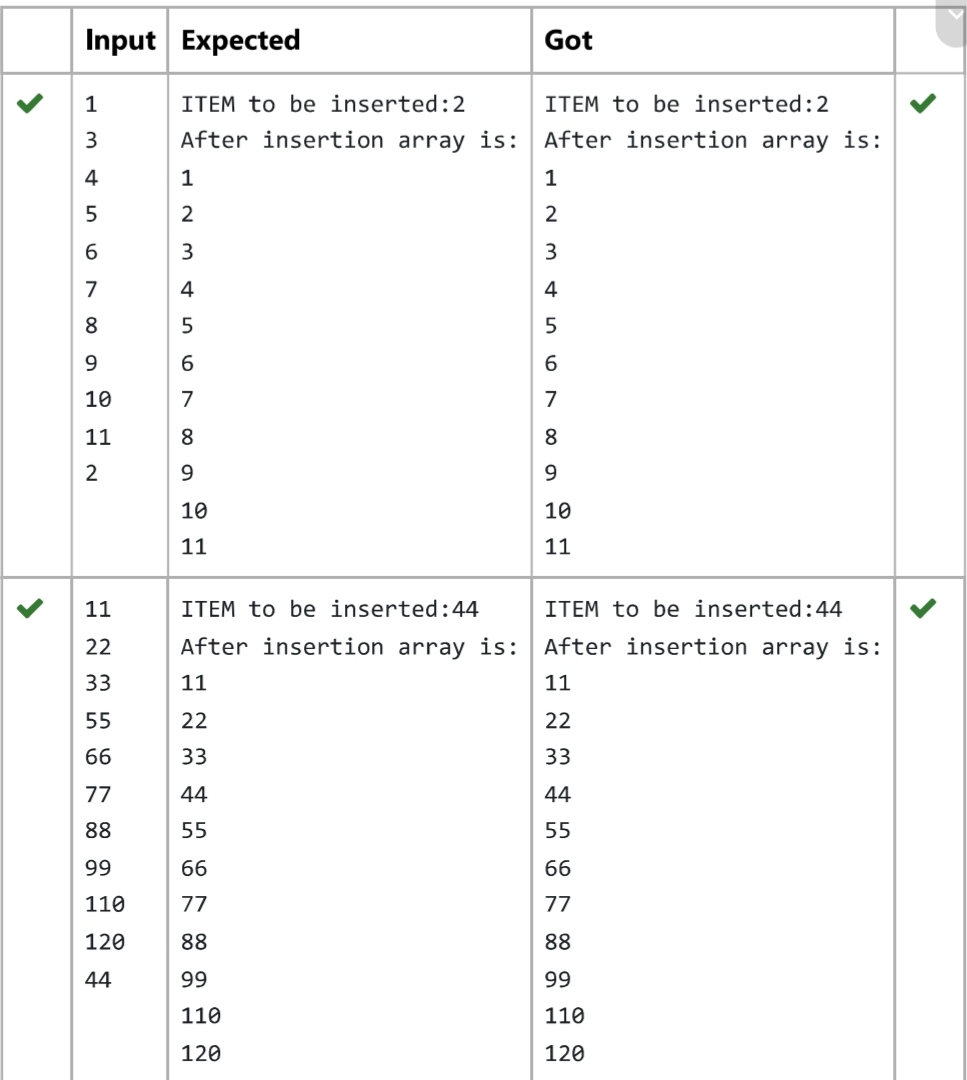
my\_list.sort()

print("After insertion, the array is:")

for j in range(0, 11):

print(my\_list[j])

**Output:**

****

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

Constraints

1 ≤ n ≤ 1015

1 ≤ p ≤ 109

The first line contains an integer n, the number to factor.

The second line contains an integer p, the 1-based index of the factor to return.

**Sample Case 0**

**Sample Input 0**

10

3

**Sample Output 0**

5

**Explanation 0**

Factoring n = 10 results in {1, 2, 5, 10}. Return the p = 3rd factor, 5, as the answer.

**Sample Case 1**

**Sample Input 1**

10

5

**Sample Output 1**

0

**Explanation 1**

Factoring n = 10 results in {1, 2, 5, 10}. There are only 4 factors and p = 5, therefore 0 is returned as the answer.

**Sample Case 2**

**Sample Input 2**

1

1

**Sample Output 2**

1

**Explanation 2**

Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.

**Program:**

import math

n = int(input())

p = int(input())

factors = []

for i in range(1, int(math.sqrt(n)) + 1):

if n % i == 0:

factors.append(i)

if n // i != i:

factors.append(n // i)

factors.sort()

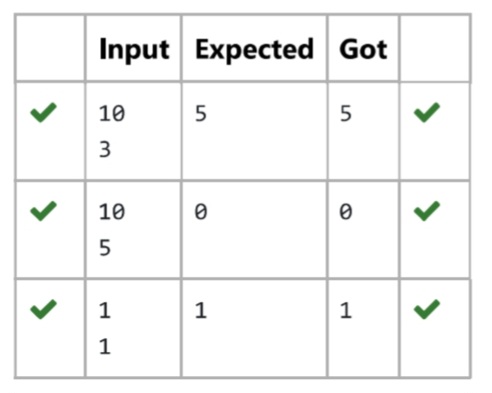
if p < len(factors):

print(factors[p - 1])

else:

print(0)

**Output:**

****

4.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.

Input Format:

First line take an Integer input from stdin which is array length n.

Second line take n Integers which is inputs of array.

Output Format:

Print the Distinct Elements in Array in single line which is space Separated

Example Input:

5

1

2

2

3

4

Output:

1 2 3 4

Example Input:

6

1

1

2

2

3

3

Output:

1 2 3

**Program:**

n = int(input())

array = []

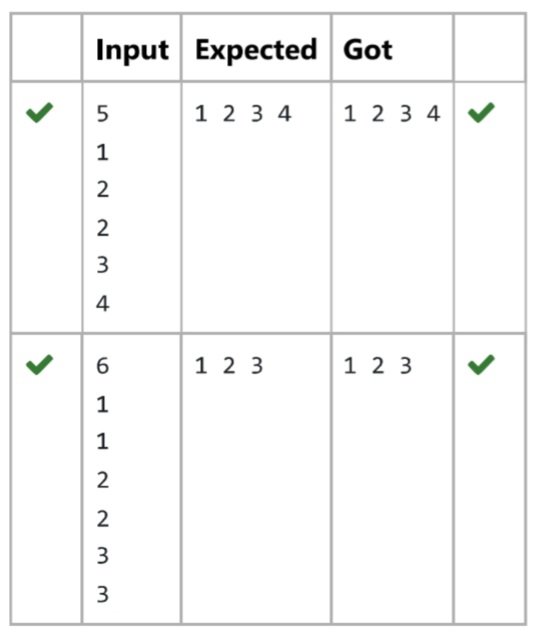
for x in range(n):

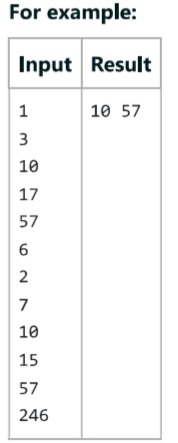
array.append(int(input()))

result = set(array)

print(\*result)

**Output:**

****

5.Find the intersection of two sorted arrays.

**Program:**

t = int(input())

while t != 0:

n1 = int(input())

list1 = []

for i in range(n1):

a = int(input())

list1.append(a)

n2 = int(input())

list2 = []

for i in range(n2):

a = int(input())

list2.append(a)

set1 = set(list1)

set2 = set(list2)

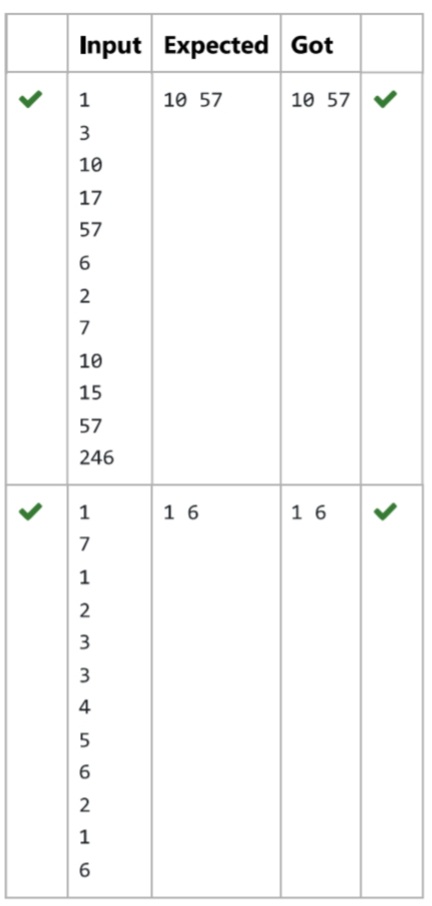
intersection = list(set1.intersection(set2))

intersection.sort()

print(intersection)

t -= 1

**Output:**

****

6.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())

list1 = []

list2 = []

c = 1

for i in range(0, m \* n \* 2, 2):

a = int(input())

b = int(input())

if c % 2 != 0:

list1.append(a)

list1.append(b)

else:

list2.append(a)

list2.append(b)

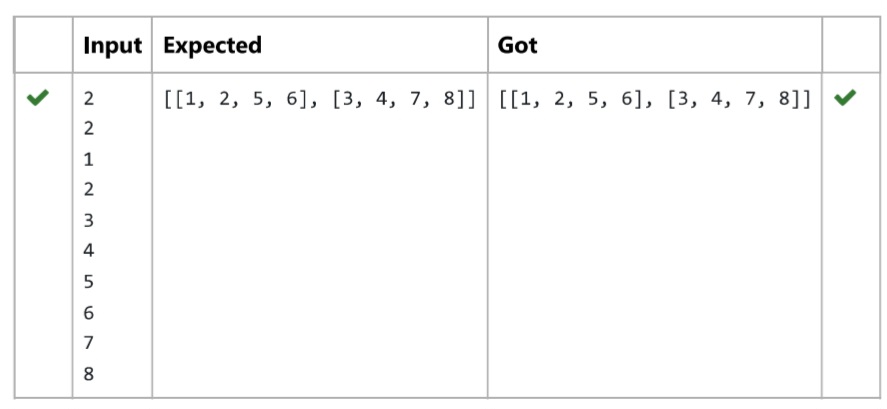
c += 1

list3 = []

list3.append(list1)

list3.append(list2)

print(list3)

**Output:**

7.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

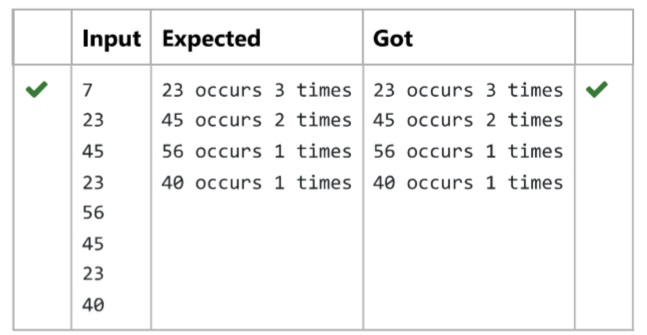
else:

frequency[num] = 1

for num, freq in frequency.items():

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

**Output:**

****

8.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())

list1 = []

for i in range(n1):

a = int(input())

list1.append(a)

n2 = int(input())

list2 = []

for i in range(n2):

a = int(input())

list2.append(a)

list3 = []

list3.extend(list1)

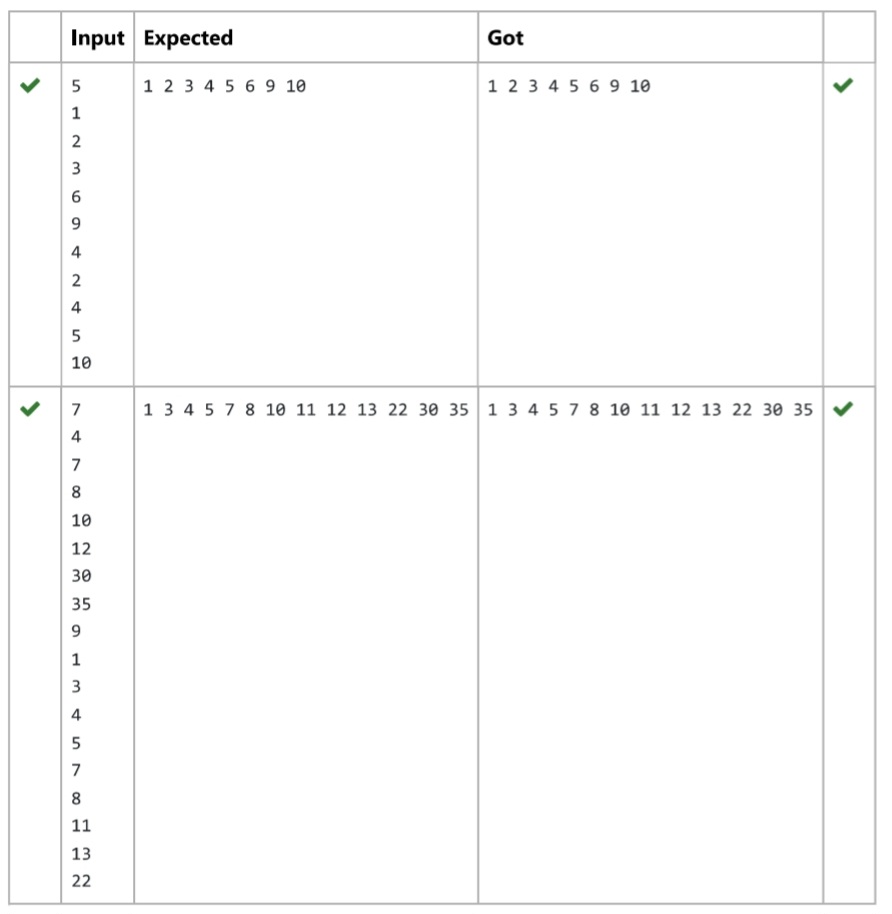
list3.extend(list2)

unique\_sorted\_list = list(set(list3))

unique\_sorted\_list.sort()

for i in unique\_sorted\_list:

print(i, end=' ')



9. 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 ≤ 105

· 1 ≤ arr[i] ≤ 2 × 104, 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:**

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

else:

print(-1)

**Output:**

****

10. Given two lists A and B, and 8 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 indexj. These lists A and B may contain duplicates. If there are multiple answers, output any of them,

For example, given

Input

5

12 28 46 32 50

50 12 32 46 28

Output

1 4 3 2 0

Explanation

A[12, 28, 46, 32, 501

B [50, 12, 32, 46, 28]

We should return

[1.4. 3. 2, 0]

as PID) = 1 because the Oth element of A appears at B[1], and P[1] 4 because the 1st element of A appears at B[4], and so on.

Note:

1. A, B have equal lengths in range [1, 100].

2. A[i], B[i] are integers in range [0, 10^5].

**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)

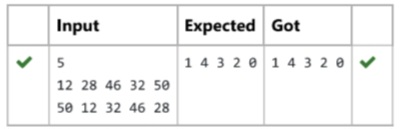
n = int(input())

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

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

print(index\_mapping(A, B))

**Output:**

****