1 2

1 4

Sample Output:

2 2 4 3

4

4

1 4 5 3 2

5

4

2

Sample Input:

purchased, in ascending order.

For each test case, print two space-separated integers denong the indices of the two avors

Output Format:

There will always be a unique soluon.

1 ≤ cost[i] ≤ 104, ∀ i ∈ [1, n]

2 ≤ n ≤ 104

2 ≤ m ≤ 104

1 ≤ t ≤ 50

Constraints:

The index within the cost array represents the avor of the ice cream purchased.

3. n space-separated integers denong the cost of each avor: cost[cost[1], cost[2], ..., cost[n]].Note:

2. The integer n, the number of avors oered at the me.

1. The integer m, the amount of money they have pooled.

The next t sets of lines each describe a visit. Each trip is described as follows:

The rst line contains an integer, t, denong the number of trips to the ice cream parlor.

Input Format:

cost: an integer array denong the cost of each avor of ice cream

m: an integer denong the amount of money they have to spend

It has the following:

of the two avors they buy, sorted ascending.

Complete the code in the editor below. It should return an array containing the indices of the prices

The two avors cosng 1 and 5 meet the criteria. Using 1-based indexing, they are at indices 1 and 4

they have.For example, they have m = 6 to spend and there are avors cosng cost = [1, 2, 3, 4, 5, 6].

money.Given a list of prices for the avors of ice cream, select the two that will cost all of the money

same avor that Sunny does. The only other rule they have is that they spend all of their

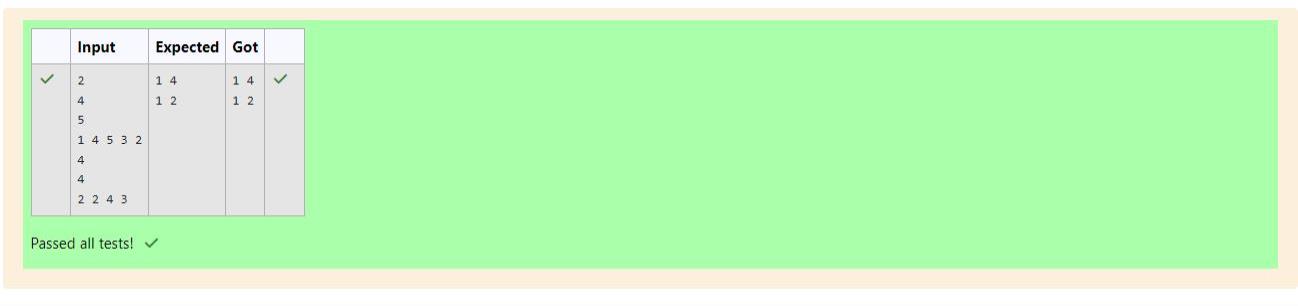
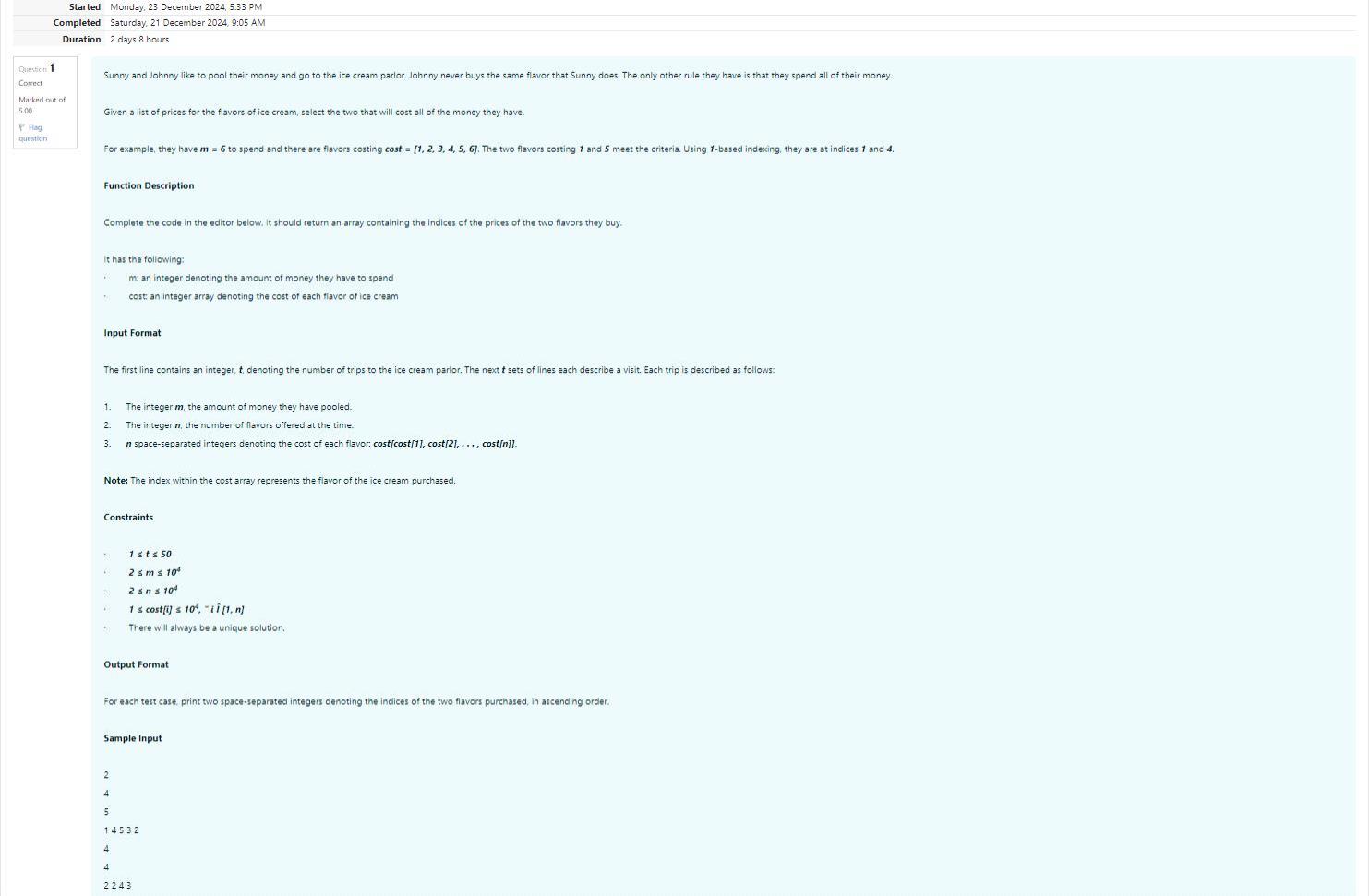
Sunny and Johnny like to pool their money and go to the ice cream parlor. Johnny never buys the

Q1) Problem Statement:

Week 07

Q2) problem statement:

Output:



204 205 206

Sample Output:

203 204 204 205 206 207 205 208 203 206 205 206 204

13

203 204 205 206 207 208 203 204 205 206

10

Sample Input:

Output the missing numbers in ascending order.

Output Format:

1 ≤ n, m ≤ 2 × 105, 1 ≤ arr[i], brr[i] ≤ 2 × 104, Xmax − Xmin < 101

Constraints:

4. The next line contains m space-separated integers brr[i]

3. m - the size of the second list, brr

2. The next line contains n space-separated integers arr[i]

1. n - the size of the rst list, arr

There will be four lines of input:

Input Format:

brr: the original array of numbers

arr: the array with missing numbers

It has the following:

Complete the code in the editor below. It should return a sorted array of missing numbers.

second list is less than or equal to 100.

it is missing mulple mesThe dierence between the maximum and minimum number in the

You have to print all the missing numbers in ascending order.Print each missing number once, even if

both lists is the same. If that is not the case, then it is also a missing number.

If a number occurs mulple mes in the lists, you must ensure that the frequency of that number in

Notes:

numbers brr = [7, 2, 5, 4, 6, 3, 5, 3]. The numbers missing are [4, 6].

As an example, the array with some numbers missing, arr = [7, 2, 6, 5, 3, 5, 3]. The original array of

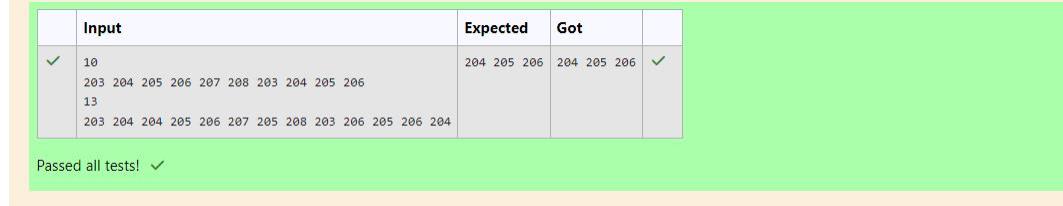
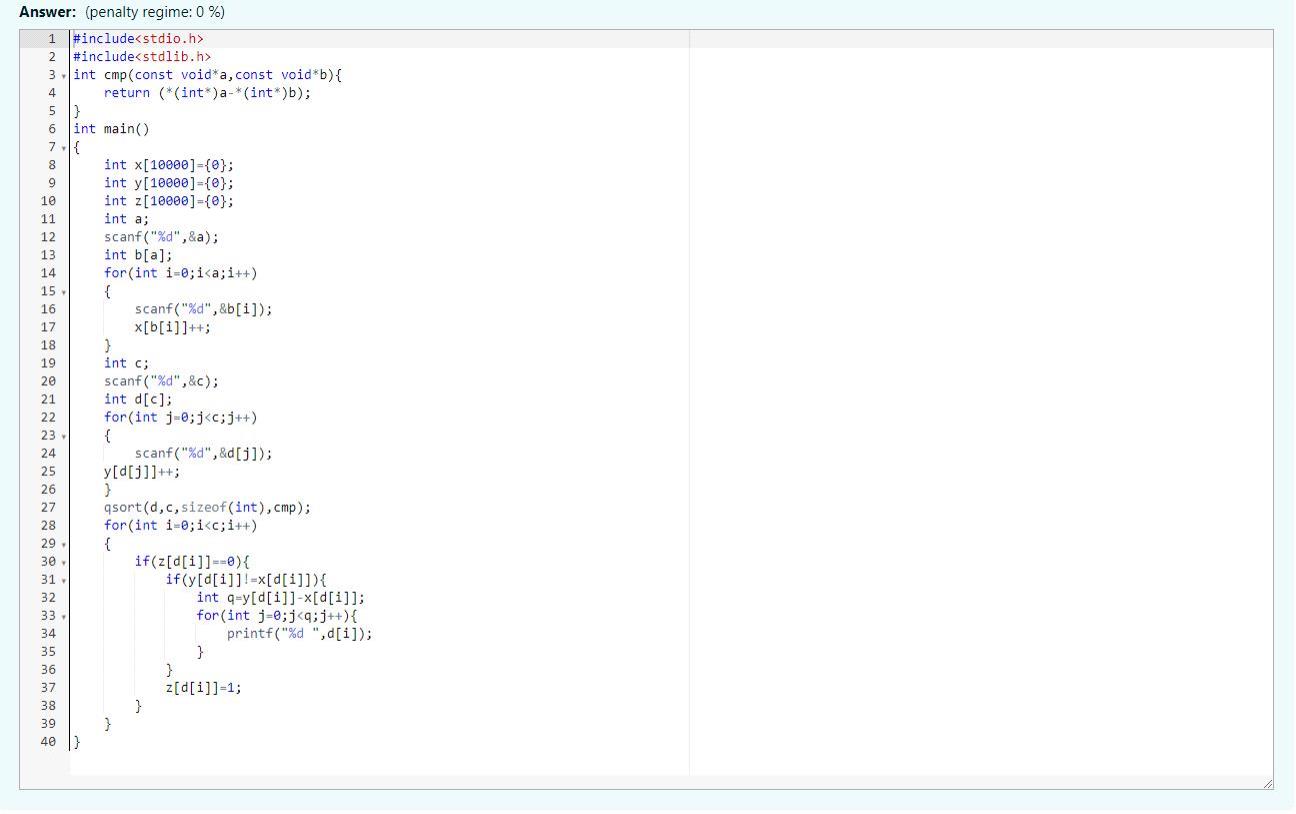
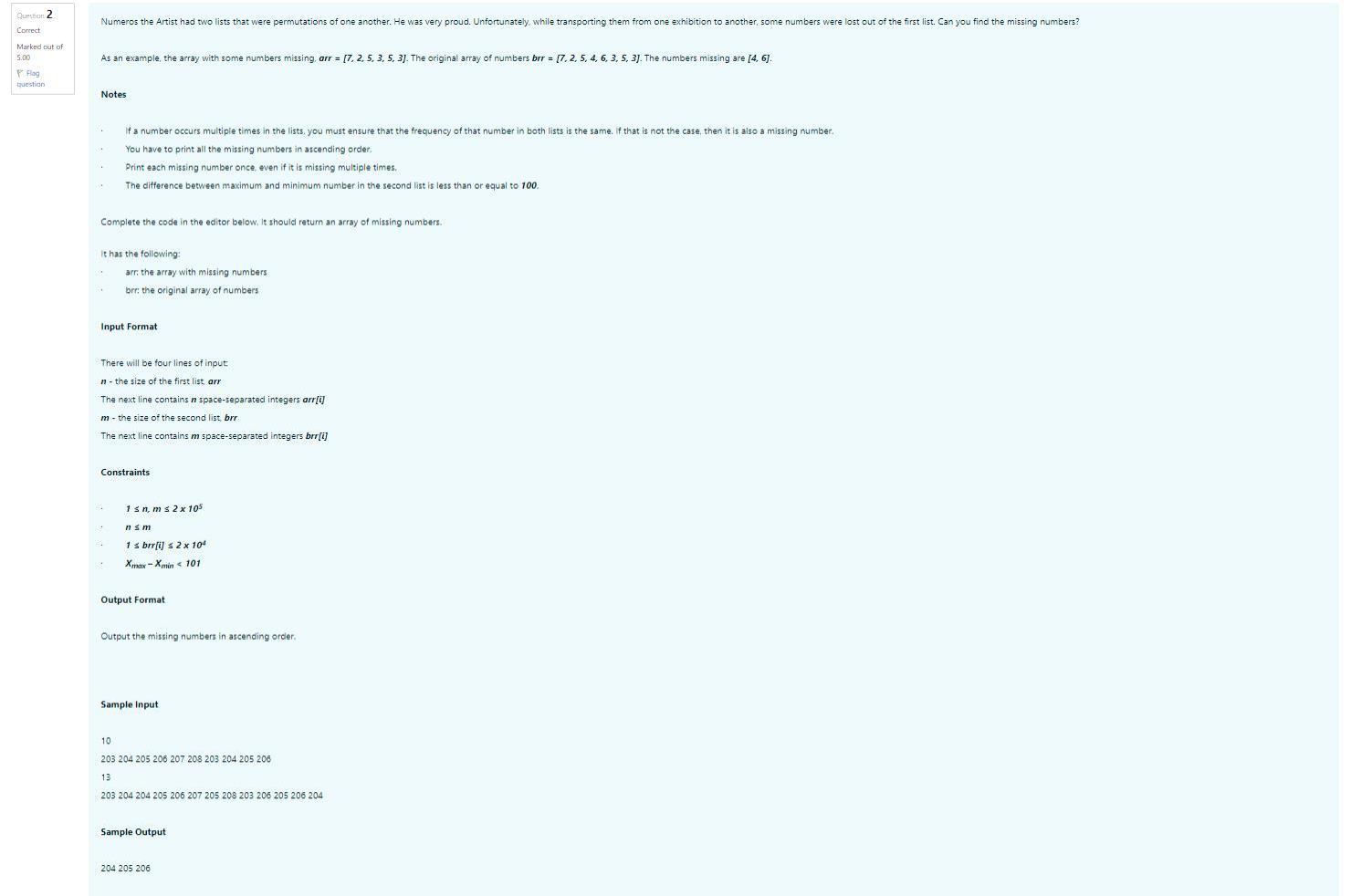
of the rst list. Can you nd the missing numbers?

Unfortunately, while transporng them from one exhibion to another, some numbers were lost out

Numeros the Arst had two lists that were permutaons of one another. He was very proud.

Q3) Problem Statement:

Output:



YES

NO

Sample Output 0:

1 2 3 3

4

1 2 3

3

2

Sample Input 0:

elements on its left is equal to the sum of the elements on its right; otherwise print NO.

For each test case print YES if there exists an element in the array, such that the sum of the

Output Format:

1 ≤ T ≤ 10, 1 ≤ n ≤ 105, 1 ≤ arr[i] ≤ 2 x 104, 0 ≤ i ≤ n

Constraints:

- The second line contains n space-separated integers arr[i] where 0 ≤ i < n.

- The rst line contains n, the number of elements in the array arr.

The next T pairs of lines each represent a test case.

The rst line contains T, the number of test cases.

Input Format:

meeng the criterion or NO otherwise. It has the following: arr: an array of integers.

Complete the code in the editor below. It should return a string, either YES if there is an element

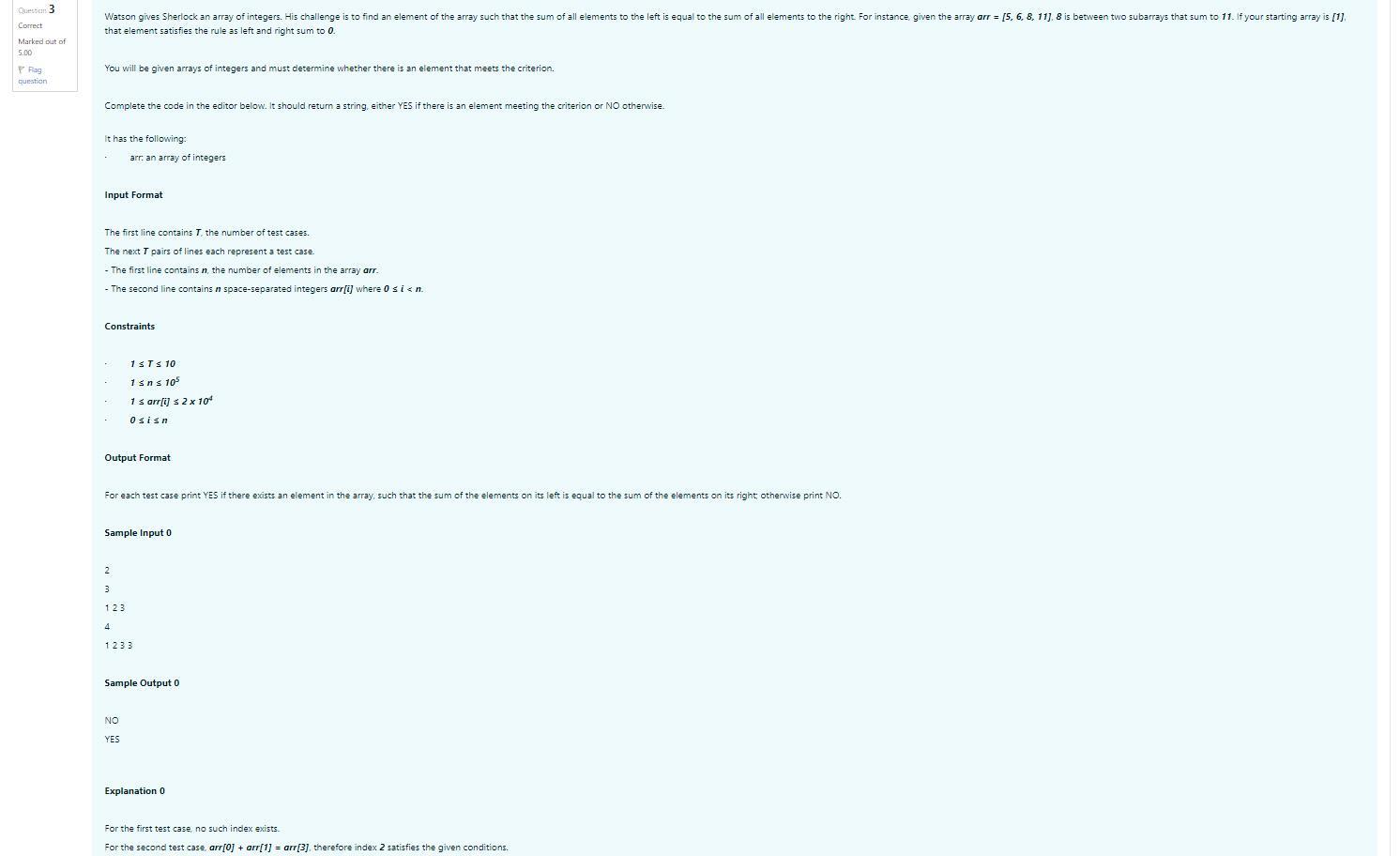
must determine whether there is an element that meets the criterion.

that element sases the rule as le and right sums to 0. You will be given arrays of integers and

the array arr = [5, 6, 8, 11], 8 is between two subarrays that sum to 11. If your starng array is [1],

the sum of all elements to the le is equal to the sum of all elements to the right. For instance, given

Watson gives Sherlock an array of integers. His challenge is to nd an element of the array such that



Output:

