Week 07

## Q1)ProblemStatement:

SunnyandJohnnyliketopool theirmoneyandgoto theicecreamparlor.Johnnyneverbuysthe same flavor that Sunny does. The only other rule they have is that they spend all of their

money.Givenalistofpricesfortheflavorsof icecream,selectthetwo that will costall ofthemoney theyhave.Forexample,theyhavem=6tospend andthereareflavorscostingcost =[1,2,3,4,5,6]. The twoflavors costing 1and 5meetthe criteria. Using 1-basedindexing,they are atindices 1and 4 Complete the code in the editor below. It should return an array containing the indices of the prices of the two flavors they buy, sorted ascending.

Ithasthefollowing:

m: an integer denoting the amount of money they have to spend cost:anintegerarraydenotingthecostofeachflavoroficecream **Input Format:**

Thefirstlinecontainsaninteger,t,denotingthenumberoftripsto theice creamparlor. The next t sets of lines each describe a visit. Each trip is described as follows:

1. Theintegerm,theamountofmoneytheyhavepooled.
2. Theintegern,thenumberofflavorsofferedatthetime.
3. nspace-separatedintegersdenotingthecostofeachflavor:cost[cost[1],cost[2],...,cost[n]].Note: The index within the cost array represents the flavor of the ice cream purchased.

Constraints:

1≤t≤50

2≤m≤104

2≤n≤104

1≤cost[i]≤104,∀i ∈ [1, n]

Therewillalwaysbeauniquesolution.

## OutputFormat:

Foreachtestcase,printtwo space-separatedintegers denotingtheindicesofthetwo flavors purchased, in ascending order.

## Sample Input:

2

4

5

14532

4

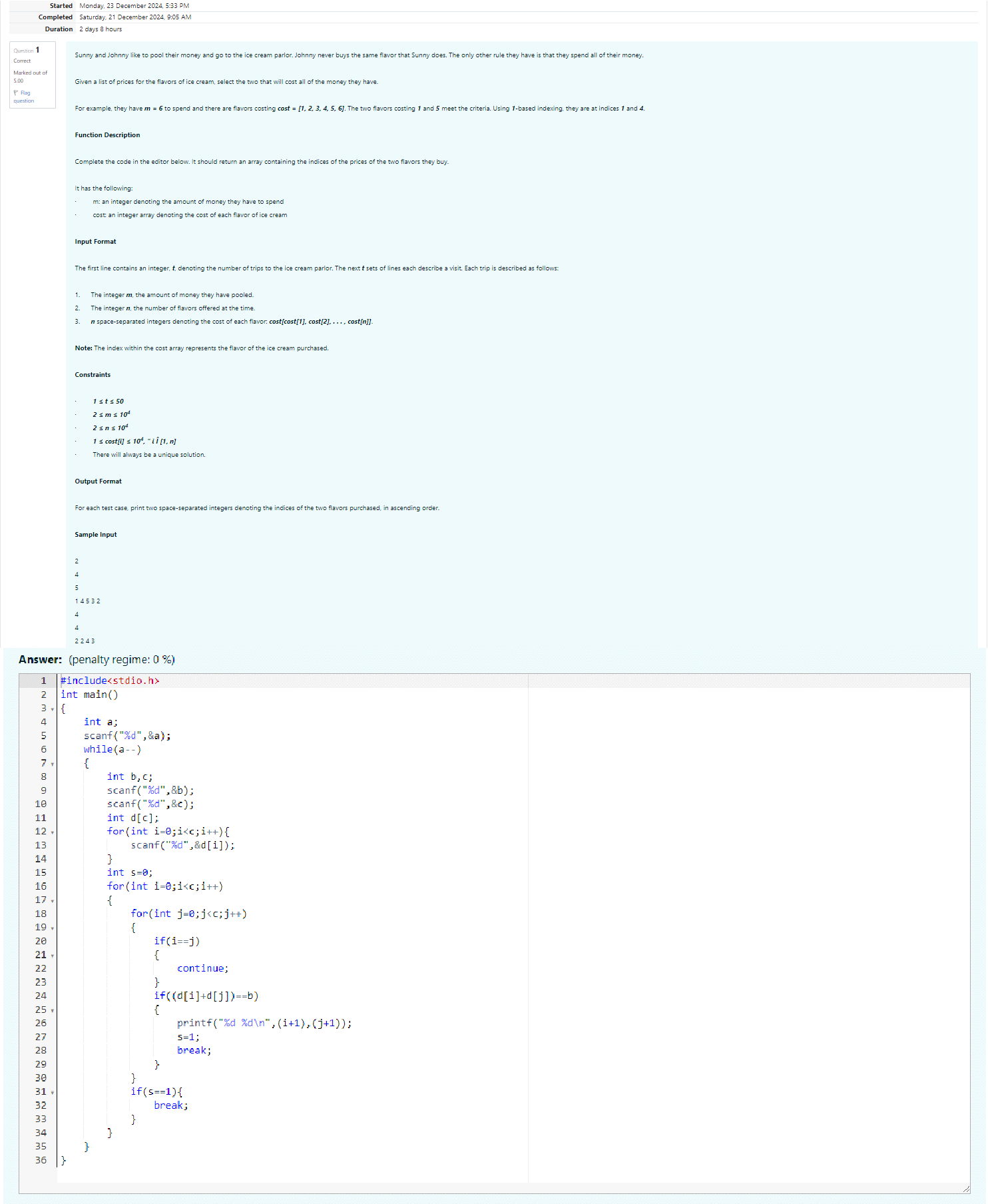
4

2243

## Sample Output:

14

12



# Output:

## Q2)problemstatement:

NumerostheArtisthadtwoliststhatwerepermutationsofoneanother.Hewasveryproud.

Unfortunately,whiletransportingthemfromoneexhibitiontoanother,somenumberswerelostout of the first list. Can you find the missing numbers?

Asanexample,thearraywithsomenumbersmissing,arr=[7,2,6, 5,3,5,3]. Theoriginalarrayof

numbersbrr=[7,2, 5,4,6,3, 5,3].Thenumbersmissingare[4,6].

## Notes:

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

Youhavetoprintallthemissingnumbersinascendingorder.Printeachmissingnumberonce,evenif it is missing multiple timesThe difference between the maximum and minimum number in the

secondlistislessthanorequalto 100.

Completethecodeintheeditorbelow.Itshouldreturnasortedarrayofmissingnumbers. It has the following:

arr:thearraywithmissingnumbers brr: the original array of numbers **Input Format:**

Therewillbefourlinesofinput:

1. n-thesizeofthefirstlist,arr
2. Thenextlinecontainsnspace-separatedintegersarr[i]
3. m-thesizeofthesecond list,brr
4. Thenextlinecontainsmspace-separatedintegersbrr[i] Constraints:

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

## OutputFormat:

Outputthemissingnumbersinascendingorder.

## Sample Input:

10

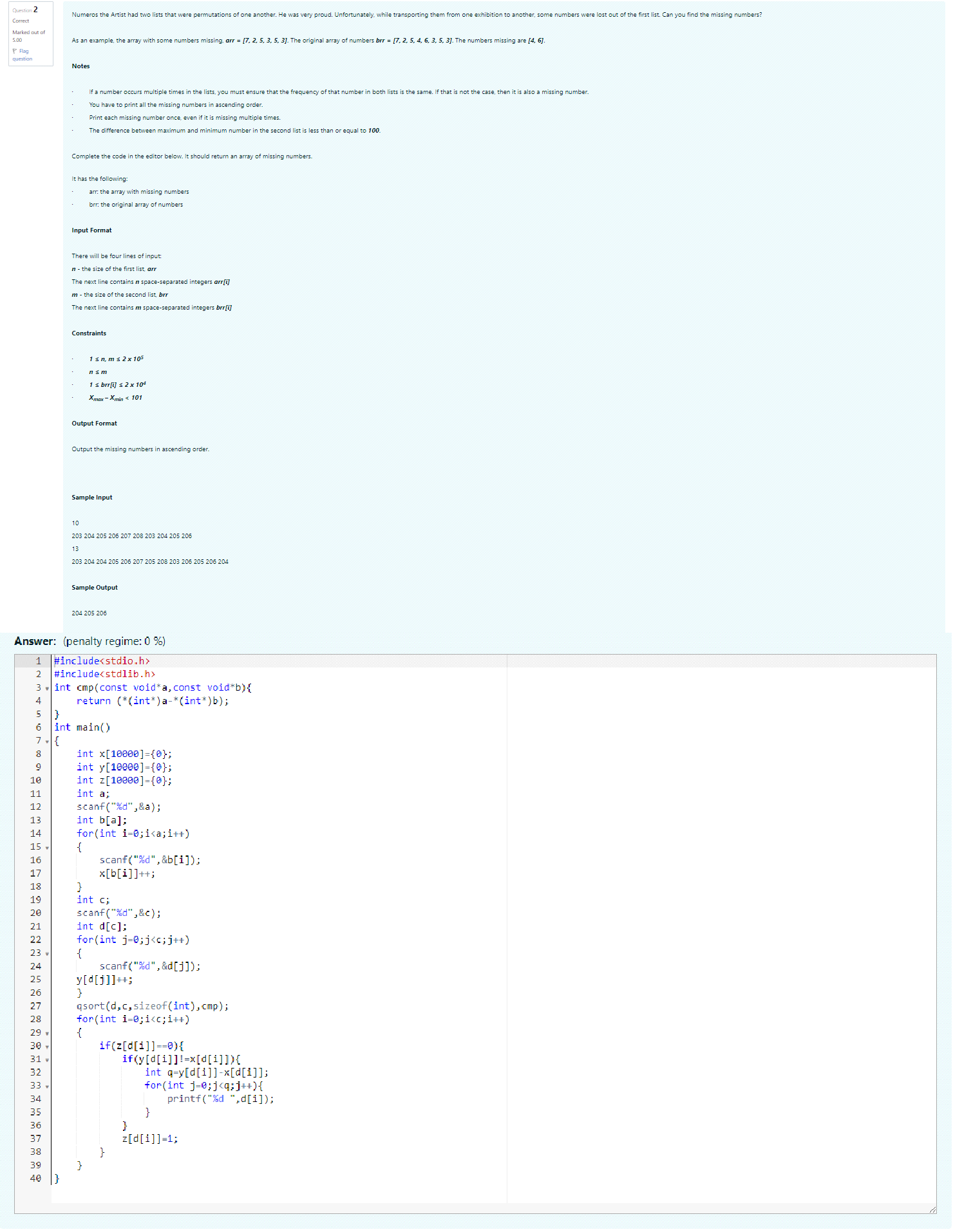
203204 205206207208203204205206

13

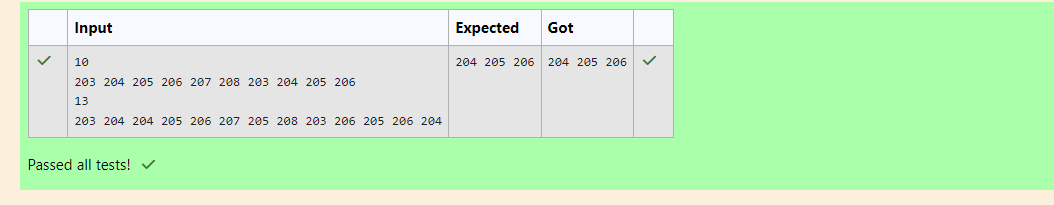
203204 204205206207205208203206205206204

## Sample Output:

204205206



# Output:



## Q3)ProblemStatement:

Watsongives Sherlock anarrayofintegers.Hischallenge istofindanelementofthearray suchthat thesumofallelementstotheleftisequaltothesumofallelementstotheright.Forinstance,given the array arr = [5, 6, 8, 11], 8 is between two subarrays that sum to 11. If your starting array is [1], that element satisfies the rule as left and right sums to 0. You will be given arrays of integers and must determine whether there is an element that meets the criterion.

Completethecodeintheeditorbelow.Itshouldreturnastring,eitherYESifthereisanelement meeting the criterion or NO otherwise. It has the following: arr: an array of integers.

## Input Format:

The first line contains T, the number of test cases. ThenextTpairsoflineseachrepresentatestcase.

* Thefirstlinecontainsn,thenumberofelementsinthearray arr.
* Thesecondlinecontainsnspace-separatedintegersarr[i]where0≤i<n. Constraints:

1≤T≤10,1 ≤n ≤105,1≤arr[i] ≤2x104,0≤i≤n

## OutputFormat:

Foreachtest caseprint YESifthereexistsanelement inthearray,suchthat thesumofthe elements on its left is equal to the sum of the elements on its right; otherwise print NO. **Sample Input 0:**

2

3

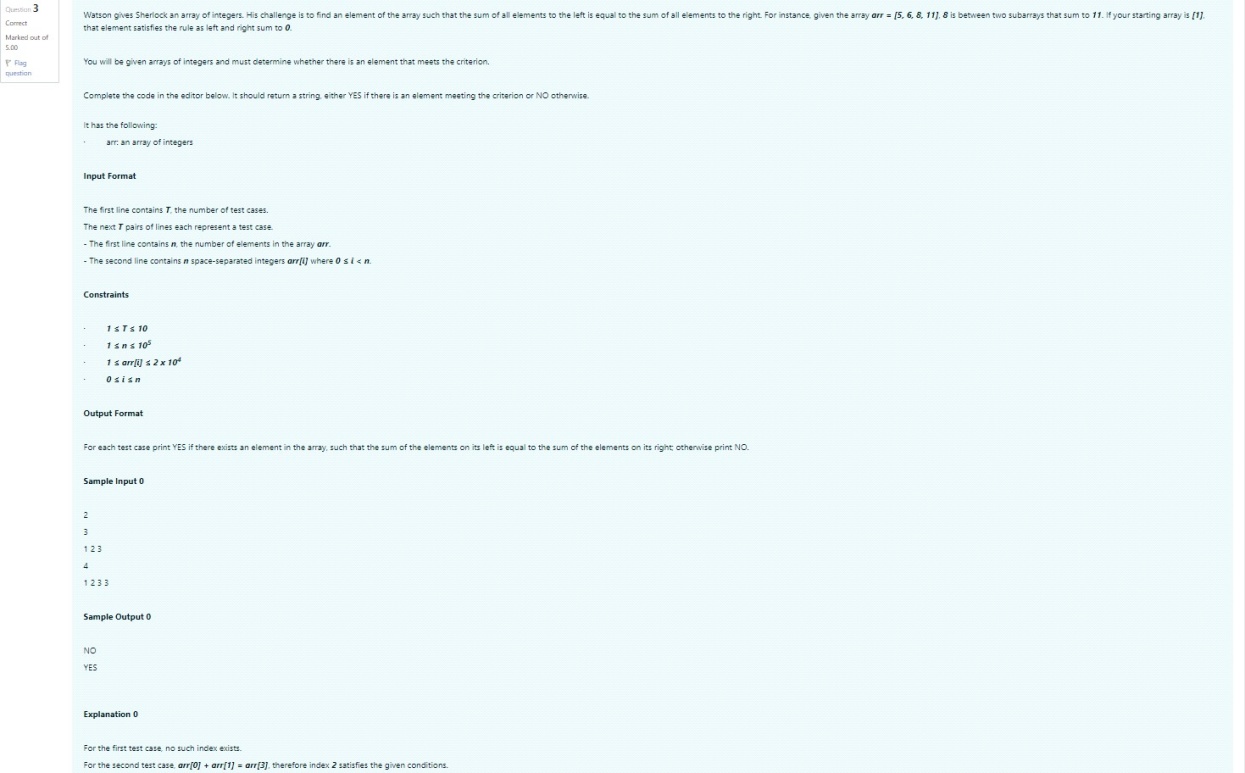
123

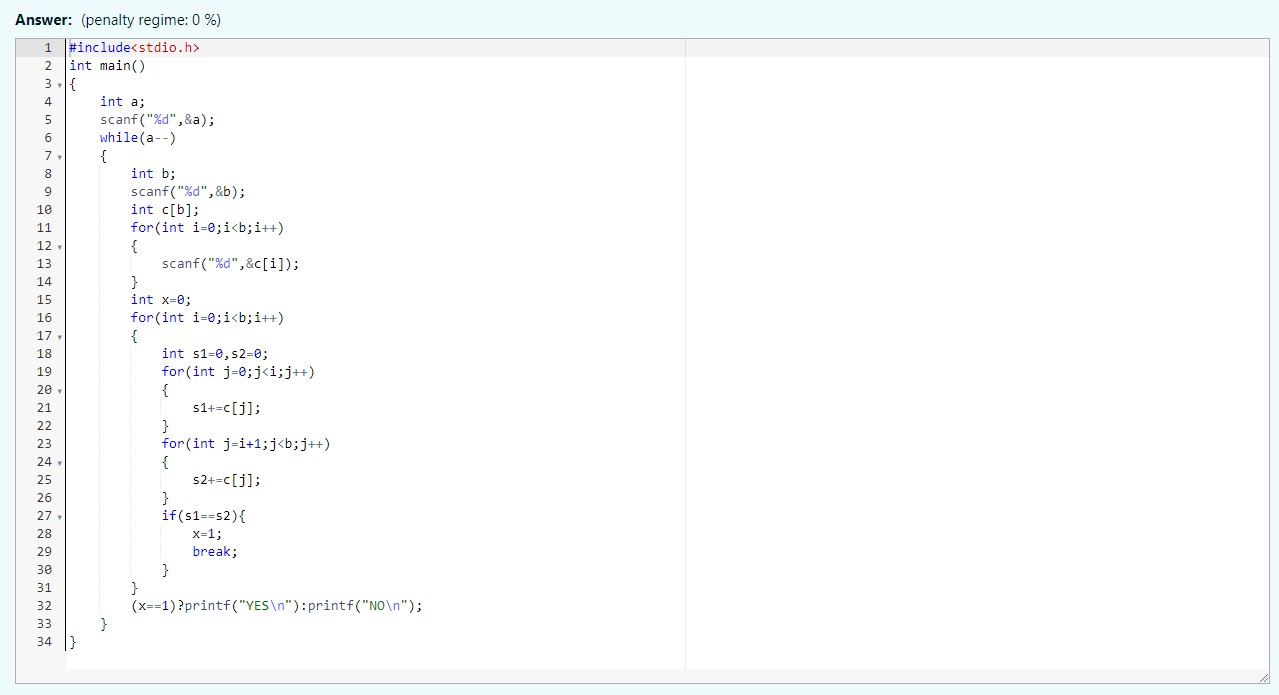
4

1233

## SampleOutput0:

NO YES





Output:

