Week 07

**Q1) Problem Statement:**   
Sunny and Johnny like to pool their money and go to the ice cream parlor. Johnny never buys the same flavor that Sunny does. The only other rule they have is that they spend all of their   
money.Given a list of prices for the flavors of ice cream, select the two that will cost all of the money they have.For example, they have m = 6 to spend and there are flavors costing cost = [1, 2, 3, 4, 5, 6]. The two flavors costing 1 and 5 meet the criteria. Using 1-based indexing, they are at indices 1 and 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.

It has the following:   
m: an integer denoting the amount of money they have to spend   
cost: an integer array denoting the cost of each flavor of ice cream   
**Input Format:**   
The first line contains an integer, t, denoting the number of trips to the ice cream parlor.

The next t sets of lines each describe a visit. Each trip is described as follows: 1. The integer m, the amount of money they have pooled.

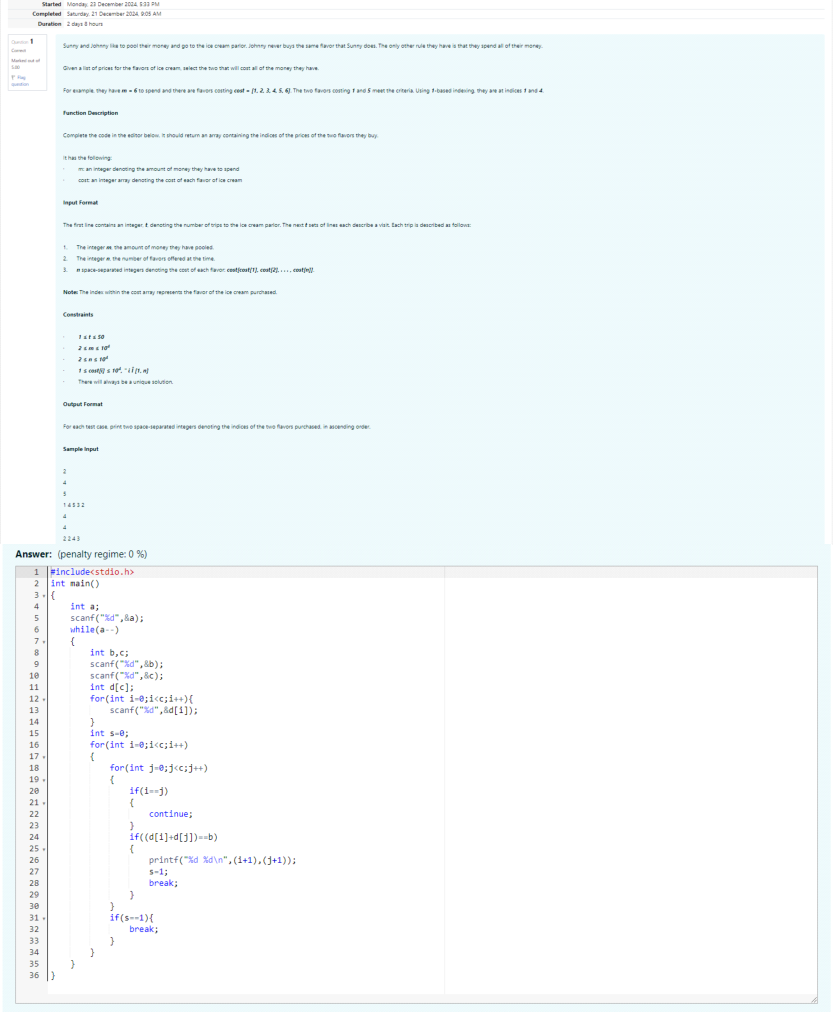
2. The integer n, the number of flavors offered at the time.

3. n space-separated integers denoting the cost of each flavor: cost[cost[1], cost[2], ..., cost[n]].Note: The index within the cost array represents the flavor of the ice cream purchased.

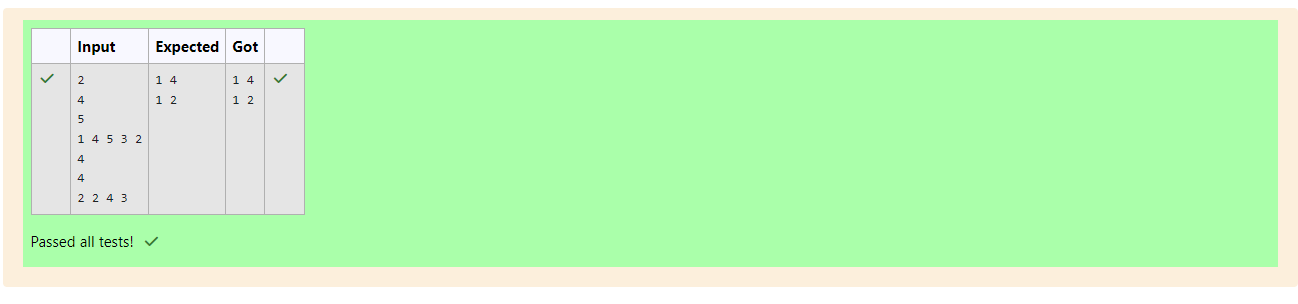
Constraints:   
1 ≤ t ≤ 50   
2 ≤ m ≤ 104   
2 ≤ n ≤ 104   
1 ≤ cost[i] ≤ 104, ∀ i ∈ [1, n]   
There will always be a unique solution.

**Output Format:**   
For each test case, print two space-separated integers denoting the indices of the two flavors purchased, in ascending order.

**Sample Input:**   
2   
4   
5   
1 4 5 3 2   
4   
4   
2 2 4 3   
**Sample Output:**   
1 4   
1 2



Output:



**Q2) problem statement:**

NumerostheArtisthadtwoliststhatwerepermutationsofoneanother. Hewasveryproud. Unfortunately, whiletransportingthemfromoneexhibitiontoanother, somenumberswerelostout ofthefirstlist. Canyoufindthemissingnumbers?

Asanexample, thearraywithsomenumbersmissing, arr = [7, 2, 6, 5, 3, 5, 3]. Theoriginalarrayof numbersbrr = [7, 2, 5, 4, 6, 3, 5, 3]. Thenumbersmissingare [4, 6].

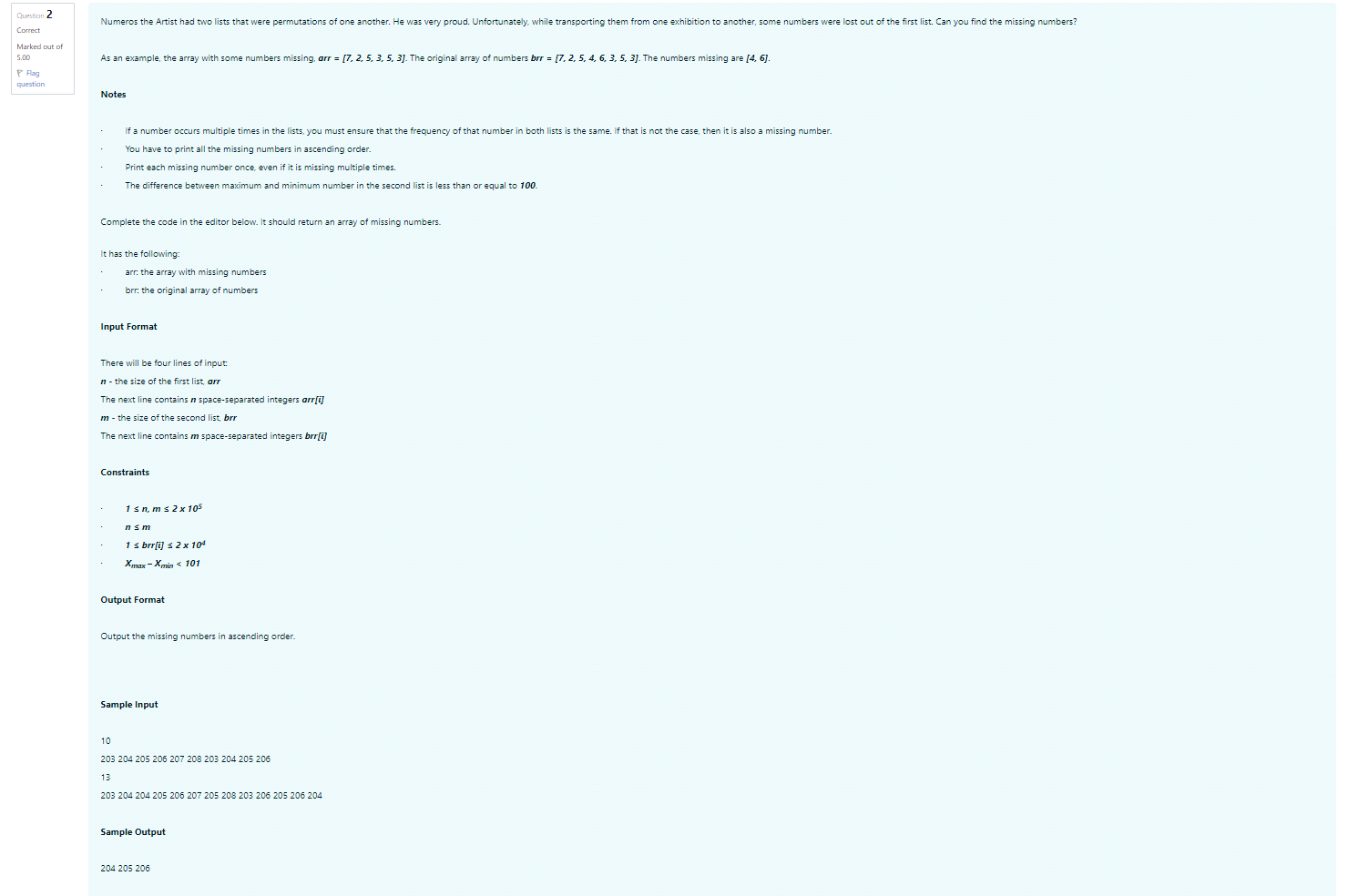
**Notes:**   
Ifanumberoccursmultipletimesinthelists, youmustensurethatthefrequencyofthatnumberin bothlistsisthesame. Ifthatisnotthecase, thenitisalsoamissingnumber.

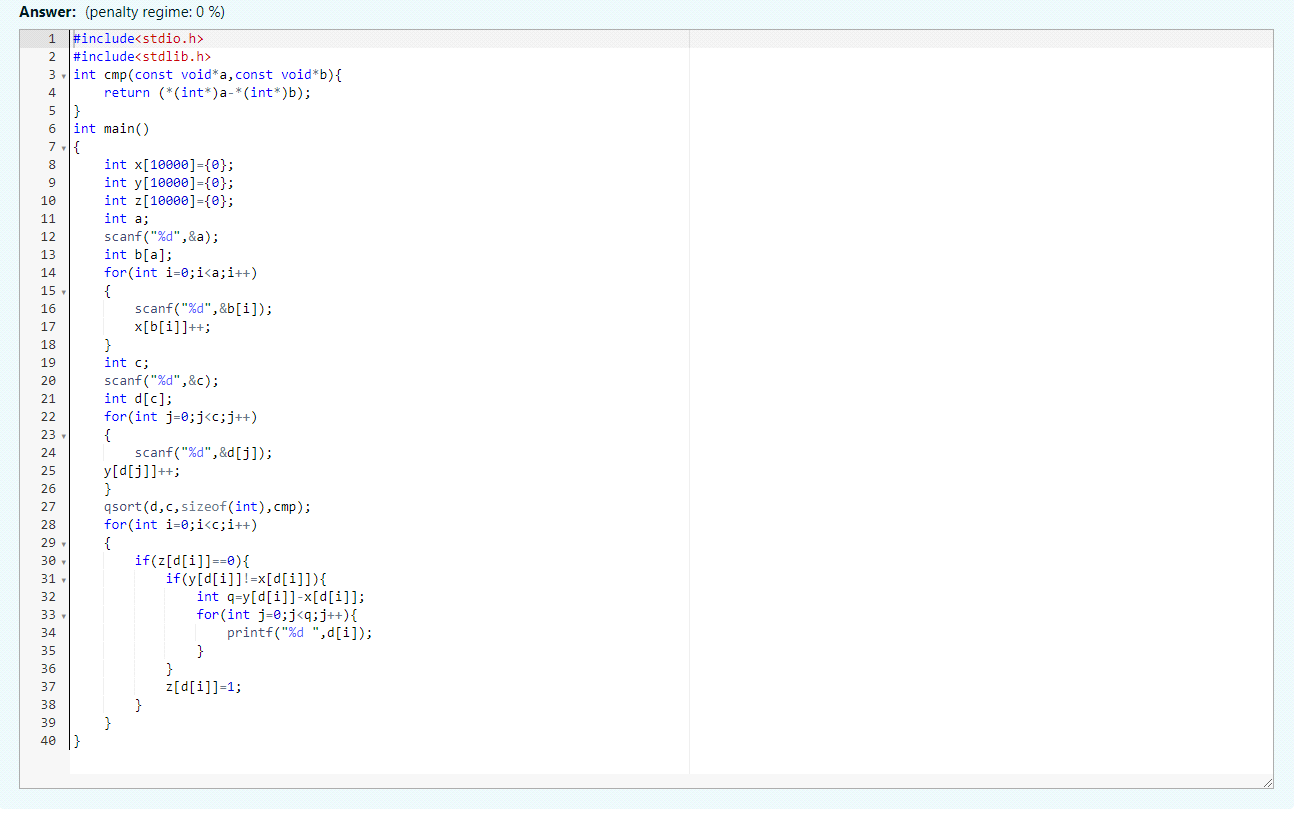
Youhavetoprintallthemissingnumbersinascendingorder.Printeachmissingnumberonce, evenif itismissingmultipletimesThedifferencebetweenthemaximumandminimumnumberinthe secondlistislessthanorequalto100.

Completethecodeintheeditorbelow. Itshouldreturnasortedarrayofmissingnumbers.

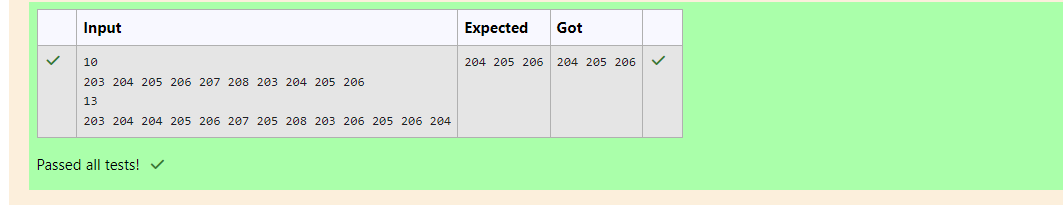
Ithasthefollowing:   
arr: thearraywithmissingnumbers   
brr: theoriginalarrayofnumbers   
**InputFormat:**   
Therewillbefourlinesofinput:   
1. n - thesizeofthefirstlist, arr   
2. Thenextlinecontainsnspace-separatedintegersarr[i]   
3. m - thesizeofthesecondlist, brr   
4. Thenextlinecontainsmspace-separatedintegersbrr[i]   
Constraints:   
1 ≤ n, m ≤ 2 × 105, 1 ≤ arr[i], brr[i] ≤ 2 × 104, Xmax − Xmin < 101 **OutputFormat:**   
Outputthemissingnumbersinascendingorder.

**SampleInput:**   
10   
203204205206207208203204205206   
13   
203204204205206207205208203206205206204 **SampleOutput:**   
204205206





Output:



**Q3) ProblemStatement:**

WatsongivesSherlockanarrayofintegers. Hischallengeistofindanelementofthearraysuchthat thesumofallelementstotheleftisequaltothesumofallelementstotheright. Forinstance, given thearrayarr = [5, 6, 8, 11], 8isbetweentwosubarraysthatsumto11. Ifyourstartingarrayis [1], thatelementsatisfiestheruleasleftandrightsumsto0. Youwillbegivenarraysofintegersand mustdeterminewhetherthereisanelementthatmeetsthecriterion.

Completethecodeintheeditorbelow. Itshouldreturnastring, eitherYESifthereisanelement meetingthecriterionorNOotherwise. Ithasthefollowing: arr: anarrayofintegers.

**InputFormat:**   
ThefirstlinecontainsT, thenumberoftestcases.

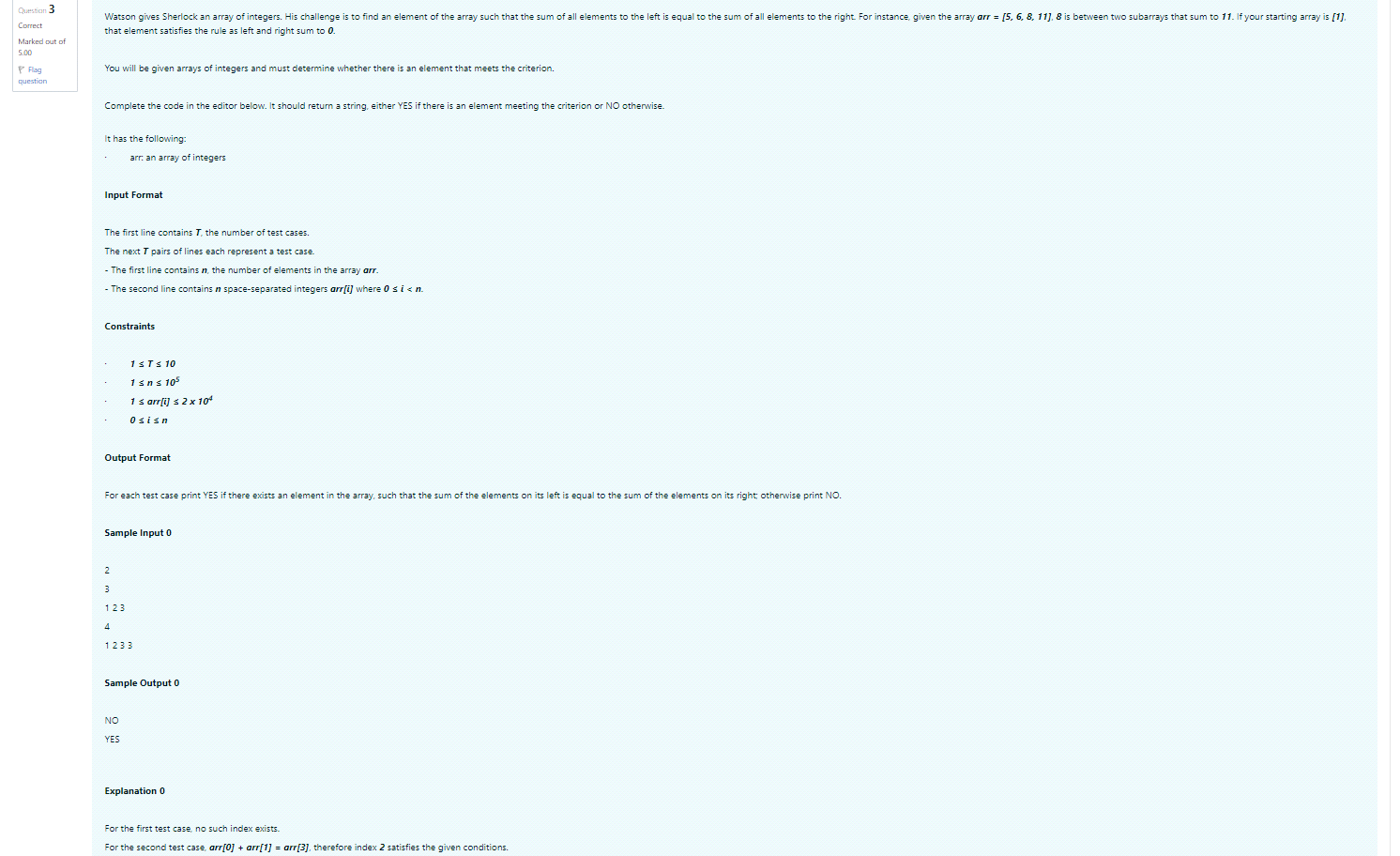
ThenextTpairsoflineseachrepresentatestcase.

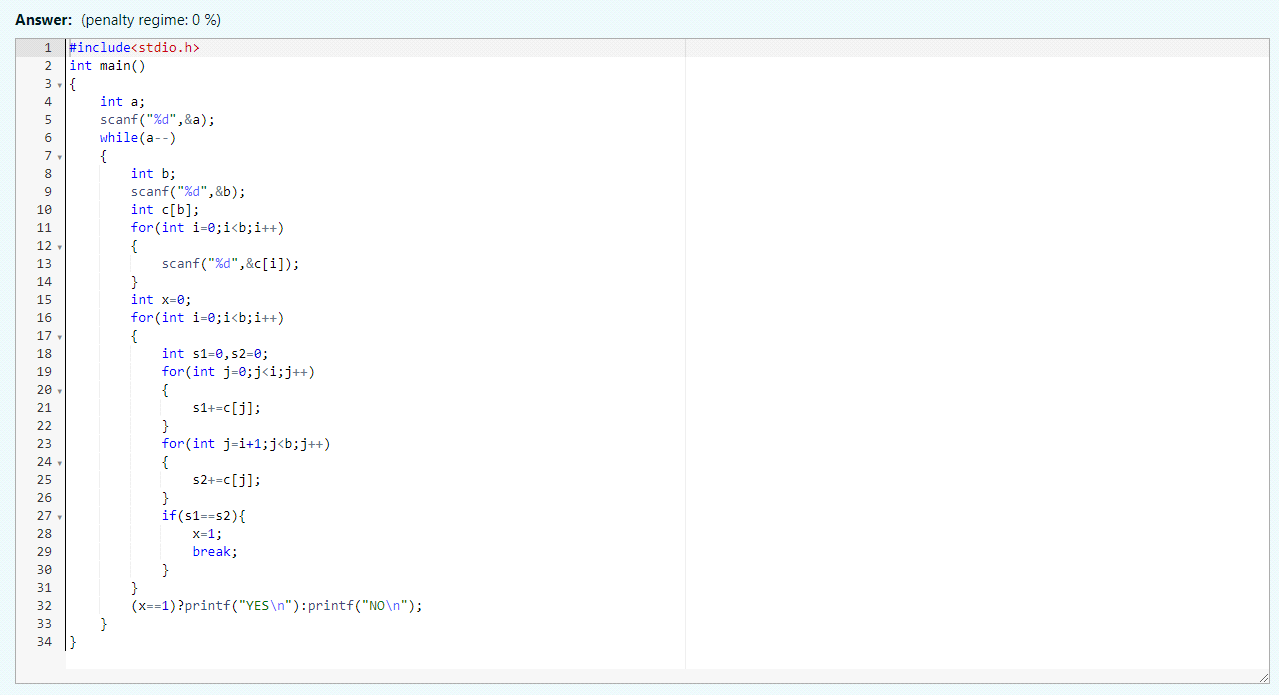
- Thefirstlinecontainsn, thenumberofelementsinthearrayarr.

- Thesecondlinecontainsnspace-separatedintegersarr[i] where0 ≤ i < n.

Constraints:   
1 ≤ T ≤ 10, 1 ≤ n ≤ 105, 1 ≤ arr[i] ≤ 2 x 104, 0 ≤ i ≤ n   
**Output Format:**   
For each test case print YES if there exists an element in the array, such that the sum of the elements on its left is equal to the sum of the elements on its right; otherwise print NO.

**Sample Input 0:**   
2   
3   
1 2 3   
4   
1 2 3 3   
**Sample Output 0:**   
NO   
YES





Output:

