**Uncle Grandpa and hashset**

Uncle Grandpa defines a multiset as a modification of the concept of a set that, unlike a set, allows for multiple instances for each of its elements. Two multiset are equal if for each value that appear in one set times, it will also appear in the other set exactly times and vice versa.

Uncle Grandpa has two arrays and of equal length. He now wants to process queries of the following form: that requires him to check if the two segments and contain equal multisets of numbers. Please help him!

## Input

The first line contains a single integer – the length of the two array

The second line contains integers – the element of the array

The third line contains integers – the element of the array

The fourth line contains a single integer (– the number of queries to process

Next lines, each line contains four integers

## Output

For each query, print “Yes” if the two segments contain equal multisets of numbers. Otherwise print “No”

## Examples

|  |  |
| --- | --- |
| Input (hashset1.in) | Output (hashset1.out) |
| 5  1 2 2 1 2  2 2 1 1 2  2  1 3 1 3  3 5 2 4 | Yes  No |

## Explanation:

For the 1st query, the segment is and the segment is . Obviously the two set of numbers in these segments are the same.

For the query, the segment is and the segment is . The two set of numbers in these segments are different.

## Hint:

One of the possible way to hash a set of numbers is by the size of the set. For example, the two set [3,4,5] and [3,4] are clearly not equal to each other because their sizes are different, but the two set [3,4] and [4,5] are also not equal to each other despite having the same size.

Another possible way to hash is by the sum of the all the elements. For example, the two set [3,4] and [4,5] are not equal to each other because their sums are different. Yet the two set [3,3] and [6], despite having the same sum, are also not equal to each other.

From the two examples, you can see that each hash function will have its wrong cases. One of the ways to lower the possibility of hash-collision is to combine several hash functions together (for example, compare both the sum and the size of the two sets). This problem aims to help you guys learn this extremely useful way of hashing! We will also provide you with all the testcases for you to test locally!

## Note:

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided

## Skeleton File

You are given the skeleton file Hashset.java. You should see the following contents when you open the file:

|  |
| --- |
| /\*\*  \* Name :  \* Matric. No :  \*/  import java.util.\*;  public class Hashset {  private void run() {  }  public static void main(String args[]) {  Hashset runner = new Hashset();  runner.run();  }  } |