Get Common Elements - 1

`Easy`

> 1. You are given a number n1, representing the size of array a1.
> 2. You are given n1 numbers, representing elements of array a1.
> 3. You are given a number n2, representing the size of array a2.
> 4. You are given n2 numbers, representing elements of array a2.
> 5. You are required to print all elements of a2 which are also present in a1 (in order of their occurence in a2). Make sure to not print duplicates (a2 may have same value present many times).

Constraints

> 1 <= n1, n2 <= 100 0 <= a1[i], a2[i] < 10

`Time complexity should be O(n)`

Format

Input

> A number n1 n1 number of elements line separated A number n2 n2 number of elements line separated

Output

> All relevant elements of a2 in separate lines (no duplicacy)

Example

> Sample Input

```
9
5
5
9
8
5
5
8
0
3
18
9
7
1
0
3
```

```
6
5
9
1
1
8
0
2
4
2
9
1
5
```

Sample Output

```
9
0
3
5
8
```

Solution

```
import java.io.*;
import java.util.*;

public class GetCommonElements {

    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
        int a1[] = new int[n1];
        for (int i = 0; i < n1; i++) {
            a1[i] = sc.nextInt();
        }
        int n2 = sc.nextInt();
        int a2[] = new int[n2];
        for (int i = 0; i < n2; i++) {
            a2[i] = sc.nextInt();
        }
        // frequency map of a1[]
        HashMap<Integer, Integer> hm = new HashMap<Integer, Integer>();
        for (int ele : a1) {
            hm.put(ele, hm.getOrDefault(ele, 0) + 1); // it will add the
element if it is not present in the map
        }
        // frequency map of a2[]
        for (int ele : a2) {
            if (hm.containsKey(ele)) {
                System.out.println(ele);
                // remove
                hm.remove(ele); // it will remove the element if it is
present in the map
            }
        }
        sc.close();
    }
}
```