



# Lower Bound-STL

by saikiran9194

Problem

Submissions

Leaderboard

Discussions

You are given  $N$  integers in the sorted order. Then you are given  $Q$  queries. In each query you will be given an integer and you have to tell whether that integer is present in the array, if so you have to tell at which index it is present and if it is not present you have to tell the index at which the smallest integer that is just greater than the given number is present.

Lower bound is a function that can be used with a sorted vector. Learn how to use lower bound to solve this problem by [Clicking Here](#).

## Input Format

The first line of the input contains the number of integers  $N$ . The next line contains  $N$  integers in sorted order. The next line contains  $Q$ , the number of queries. Then  $Q$  lines follow each containing a single integer  $Y$ .

**If the same number is present multiple times, you have to print the first index at which it occurs.**

**The input is such that you always have an answer for each query.**

## Constraints

$$1 \leq N \leq 10^5$$

$$1 \leq X_i \leq 10^9, \text{ where } x_i \text{ is } i^{\text{th}} \text{ element in the array.}$$

$$1 \leq Q \leq 10^5$$

$$1 \leq Y \leq 10^9$$

## Output Format

For each query you have to print "Yes"(without the quotes)if the number is present and at which index(1-based) it is present separated by a space. If the number is not present you have to print "No"(without the quotes) followed by the index of the next smallest number just greater than that number.

You have to output each query in a new line.

## Sample Input

```
8
1 1 2 2 6 9 9 15
4
1
4
9
15
```

## Sample Output

```
Yes 1
No 5
Yes 6
Yes 8
```

[f](#) [t](#) [in](#)



Submissions:16104



Max Score:15

Difficulty: Easy

Rate This Challenge:



Current Buffer (saved locally, editable)  

C++14  

```
1 #include <iostream>
2 #include <list>
3 #include <vector>
4 #include <algorithm>
5 #include <iterator>
6 using namespace std;
7 #define uli unsigned long int
8 #define ulli unsigned long long int
9
10 int main()
11 {
12     ios_base::sync_with_stdio(false);
13     cin.tie(0);
14     cout.tie(0);
15
16     int size; cin >> size;
17
18     vector<int> vec; vec.reserve(size);
19     copy_n(istream_iterator<int>(cin), size, back_inserter(vec));
20
21     int queryCount; cin >> queryCount;
22
23     vector<int> queries; queries.reserve(queryCount);
24     copy_n(istream_iterator<int>(cin), queryCount, back_inserter(queries));
25
26     for(auto& q : queries)
27     {
28         auto it = lower_bound(begin(vec), end(vec), q);
29         cout << (*it == q ? "Yes " : "No ") << distance(begin(vec), it) + 1 << endl;
30     }
31
32     return 0;
33 }
34
35
36
```

Line: 36 Col: 1

[Upload Code as File](#)☐ Test against custom input

Run Code

Submit Code

## Accepted

✓ Test Case #0

✓ Test Case #3

✓ Test Case #6

✓ Test Case #9

✓ Test Case #12

✓ Test Case #15

✓ Test Case #1

✓ Test Case #4

✓ Test Case #7

✓ Test Case #10

✓ Test Case #13

✓ Test Case #16

✓ Test Case #2

✓ Test Case #5

✓ Test Case #8

✓ Test Case #11

✓ Test Case #14