

Back To Basics

Project

Data Structures

Algorithms

Due date: 3 May, 2020

Problem Statement: Sitting at home in the lock-down, you are remembering the old times. Well, so are the TAs! Its time to brush up the simple concept of old times. Given a sorted array of integers of N elements and T test cases. Each test case will have a single integer X and your task is to find if and where the element exist in the array. If multiple instance of elements exist, return the lowest index. This time again, the array won't be given explicitly, but you can make some special queries to get the element at a given valid index. However, you limited special queries as well as time. Find the zero-based index where the element is present or -1 if its not.

Note

The array is sorted, either ascending or descending. You only have 20 special query per test case! BE EXTRA SURE OF THE INTERACTION. IT IS CASE SENSITIVE AND SPACE SENSITIVE. DO NOT PRINT ANY EXTRA LINES AND DO NOT PRINT ANY OTHER DEBUGGING STATEMENTS.

Input

First fixed line will contain N , the number of elements in an array. Second fixed line will contain T , the number of test cases. For each test case, a single integer X will be provided.

Interaction

To get the element at index i , output "READ i "(without quotes), and the OJ will return integer at given index i if i is valid. Note that value returned can be directly inputted as an integer. NOTE: AFTER USING cout, ALWAYS USE `fflush(NULL)`; statement in C++. If you don't use that, there will be issues with input.

Output

Once you find the lowest index i at which X is present, output "OUTPUT i "(without quotes), and if the element doesn't exist, output "OUTPUT -1"(without quotes).

Constraints

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

$$1 \leq Q \leq 10^4$$

$$-10^9 \leq a[i], X \leq 10^9$$

$$specialqueries \leq 20$$

Time Limit: 5 sec

Memory Limit: 256 MB

Sample Test Case

Array	5437897 842387 1000 200 0 0 -10000 -13773 -99999999
Interaction Sample 1(on the above arrays)	
9 3 2015 5437897 842387 1000 200 0 0 -10000 -13773 -99999999 0 5437897 842387 1000 200 0 0 -1053 5437897 842387 1000 200 0 0 -10000 -13773 -99999999	READ 0 READ 1 READ 2 READ 3 READ 4 READ 5 READ 6 READ 7 READ 8 OUTPUT -1 READ 0 READ 1 READ 2 READ 3 READ 4 OUTPUT 4 READ 0 READ 1 READ 2 READ 3 READ 4 READ 5 READ 6 READ 7 READ 8 OUTPUT -1
Interaction Sample 2(on the above arrays)	
9 3 2015 0 -10000 842387 1000 0 0 -1053 0 842387 -10000 0	READ 4 READ 6 READ 1 READ 2 OUTPUT -1 READ 4 OUTPUT 4 READ 4 READ 1 READ 6 READ 5 OUTPUT -1

Explanation

The file has 2 queries as given above. There is no fixed input, i.e. copy pasting the left side of table to test your code won't work.

The above three interactions are the examples of three different types of code on the same input. Be sure to stick to the format of interaction and any deviation from that(like extra space anywhere, or invalid

index i etc) will result in WRONG-ANSWER.

The first interaction is linear search, but it won't work for larger constraints.

The second interaction is some other search, probably randomized, and it might not work for larger constraints.