


## Count the Floors! (elevator)

Edoardo heard some strange noises coming from his elevator, so he has been tracking which trips the elevator has made in order to find where the problem is. At the beginning, the elevator was at floor  $F_0$ . Then, the elevator made  $N - 1$  trips, reaching after the  $i$ -th trip the floor  $F_i$  ( $1 \leq i \leq N - 1$ ). Edoardo wants to know which is the most visited floor, that is the floor where the elevator passed the most number of times during all the trips. Please note that the floors where the elevator began or ended its trips are counted only once and not twice.



Figure 1: The elevator in Edoardo's house.

Edoardo, however, is not really sure that he tracked the elevator correctly, so he has made  $Q$  consecutive changes to the trips. In particular, the  $j$ -th time he change the value of  $F_{P_j}$  to  $V_j$ . Can you tell Edoardo which is the most visited floor after each change and how many times it has been visited?

 Among the attachments of this task you may find a template file `elevator.*` with a sample incomplete implementation.

### Input

The first line contains the only integer  $N$ . The second line contains  $N$  integers  $F_i$ . The third line contains the only integer  $Q$ . Then  $Q$  lines follow, the  $j$ -th of which contains two integers  $P_j$  and  $V_j$ .

### Output

For each one of the  $Q$  changes, you need to write a single line with two integers: the most visited floor and how many times it has been visited. If there are multiple floors which have been visited the same number of times, write the smallest one.





### Constraints

- $2 \leq N \leq 100\,000$ .

- $1 \leq F_i \leq 200\,000$  for each  $i = 0 \dots N - 1$ .
- $1 \leq Q \leq 100\,000$ .
- $0 \leq P_j \leq N - 1$  for each  $j = 0 \dots Q - 1$ .
- $1 \leq V_j \leq 200\,000$  for each  $j = 0 \dots Q - 1$ .
- $F_i$  are distinct at the beginning and after every update.

## Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 1** (0 points)      Examples.  

- **Subtask 2** (25 points)       $1 \leq N, Q, F_i, V_j \leq 200$ .  

- **Subtask 3** (35 points)       $1 \leq N, Q \leq 1\,000, 1 \leq F_i, V_j \leq 10\,000$ .  

- **Subtask 4** (40 points)      No additional limitations.  


## Examples

input	output
5 4 2 6 1 7 5 0 3 3 4 1 1 4 2 2 5	3 4 5 3 5 3 2 3 2 3

## Explanation

In the **first sample case**, after all the queries,  $F = [3, 1, 5, 4, 2]$ . This means that:

- the first trip visits floors 3, 2 and 1;
- the second trip visits floors 2, 3, 4 and 5 (floor 1 was already counted in the previous trip);
- the third trip visits floor 4;
- the last trip visits floors 3 and 2;

This means that the most visited floor is 2, which is visited three times. Note that also floor 3 has been visited three times, but it is not the smallest.