Overtakings (g1)

You are watching a Formula 1 race, and you know that there are N drivers who are participating in the race. Each driver's car has a distinct number from 1 to N.

Figure 1: A Formula 1 race.

At first, you are given the starting order of the cars, described by an array of N values: at position i we have the car numbered C_i. You also know that there were Q overtakings which took place during the race, each of them indicated by an integer X: car number X is overtaking the car in front of them.

For example, if we have the array $C = [2, 3, 1]$, the car number 2 is leading the race, and car 1 is at the back. After the query "3" the array becomes $[3, 2, 1]$; here 2 is overtaken by 3.

For each operation print the car number that is **overtaken the highest number of times** after all the operations made so far. If there are more such cars, print the one with the smallest number.

> ☞ Among the attachments of this task you may find a template file `g1.*` with a sample incomplete implementation.

Input

The first line of the input has N and Q, representing the number of racers and the number of overtakings.

The second line contains N integers, $C_1, C_2, \ldots, C_N$, the initial order of the cars.

The next Q lines will contain an integer each, representing a car who overtakes another racer. It is guaranteed that a query won't contain the car that is currently at the first place.

Output

You will have to output Q lines

For each query you have to output one line containing the car number that is **overtaken the highest number of times** after all the operations made so far. If there are more such cars, print the one with the smallest number.

Constraints

- $1 \leq N, Q \leq 200\,000$.
- $1 \leq C_i \leq N$ for each $i = 1 \ldots N$.
- No two cars start from the same position.

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** (38 points) $N, Q \leq 2000$.

– **Subtask 3** (62 points) No additional limitations.

Examples

input	output
7 8 5 2 3 4 1 7 6 1 1 1 4 3 7 4 6	4 3 2 3 3 4 4 4

Explanation

In the **sample case**, the car order will look like this after each query:

- $C = [5, 2, 3, 1, 4, 7, 6]$
- $C = [5, 2, 1, 3, 4, 7, 6]$
- $C = [5, 1, 2, 3, 4, 7, 6]$
- $C = [5, 1, 2, 4, 3, 7, 6]$
- $C = [5, 1, 2, 3, 4, 7, 6]$
- $C = [5, 1, 2, 3, 7, 4, 6]$
- $C = [5, 1, 2, 3, 4, 7, 6]$
- $C = [5, 1, 2, 3, 4, 6, 7]$