

Equalmex

It is well known among Romanian noblemen that the beauty of an integer array $a[0], a[1], a[2], \dots, a[m-1]$ is the number of positive integers k for which you can split the array into k disjoint subarrays (sequences of consecutive elements) such that each element is contained in exactly one subarray and all the subarrays have the same minimum excluded element. The minimum excluded element of an integer array is the smallest strictly positive integer (**greater than 0**) that does not appear in the array.

You are given an integer array $v[0], v[1], \dots, v[n-1]$ and q queries of the form (l_i, r_i) , where $0 \leq l_i \leq r_i < n$ for all $0 \leq i < q$.

For each query, you have to find the beauty of the array $v[l_i], v[l_i + 1], \dots, v[r_i]$.

Implementation Details

You should implement the following procedure:

```
std::vector<int> solve(  
    int n, std::vector<int>& v,  
    int q, std::vector<std::pair<int, int>>& queries);
```

- n : the size of the integer array
- v : array of length n , the initial array
- q : the number of queries
- $queries$: array of length q describing the queries
- This procedure should return a vector of q integers containing the answer for each query.
- This procedure is called exactly once for each test case.

Constraints

- $1 \leq n \leq 600\,000$
- $1 \leq q \leq 600\,000$
- $1 \leq v[i] \leq 400\,000$ for all $0 \leq i < n$
- $0 \leq l_i \leq r_i < n$ for all $0 \leq i < q$

Subtasks

1. (4 points) $1 \leq n \leq 10, 1 \leq q \leq 100$
2. (6 points) $1 \leq n, q \leq 100$
3. (17 points) $1 \leq n, q \leq 1\,000$
4. (10 points) $1 \leq n, q \leq 100\,000$ and $1 \leq v[i] \leq 2$ for all $0 \leq i < n$
5. (30 points) $1 \leq n, q \leq 75\,000$
6. (33 points) No additional constraints.

Examples

Example 1

Consider the following call:

```
solve(10, {1, 1, 2, 2, 3, 3, 1, 2, 3, 4}, 2, {{0, 5}, {0, 8}})
```

In this sample $n = 10$ and there are 2 queries for which:

- $l_0 = 0$ and $r_0 = 5$
- $l_1 = 0$ and $r_1 = 8$

For the first query, we can split the interval in only one subarray, which is from position 0 to position 5.

In the second query, k could be either 1 or 2.

A possibility of splitting into 1 subarray is by choosing the subarray from position 0 to position 8. A possibility of splitting into 2 subarrays is by choosing the subarray from position 0 to position 5 and from position 6 to position 8.

The answer for the first query is 1 and for the second query, it is 2, so the call to `solve` will return $\{1, 2\}$.

Sample grader

The sample grader reads the input in the following format:

- line 1: $n\ q$
- line 2: $v[0]\ v[1]\ \dots\ v[n-1]$
- line $3 + i$: $l_i\ r_i$ for all $0 \leq i < q$

and outputs q lines, the result of the call to function `solve` with the corresponding parameters.