B1. Shave Beaver!

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The Smart Beaver has recently designed and built an innovative nanotechnologic all-purpose beaver mass shaving machine, "Beavershave 5000". Beavershave 5000 can shave beavers by families! How does it work? Very easily!

There are n beavers, each of them has a unique id from 1 to n. Consider a permutation $a_1, a_2, ..., a_n$ of n these beavers. Beavershave 5000 needs one session to shave beavers with ids from x to y (inclusive) if and only if there are such indices $i_1 < i_2 < ... < i_k$, that $a_{i_1} = x$, $a_{i_2} = x + 1$, ..., $a_{i_{k-1}} = y - 1$, $a_{i_k} = y$. And that is really convenient. For example, it needs one session to shave a permutation of beavers $1, 2, 3, ..., n$.

If we can't shave beavers from x to y in one session, then we can split these beavers into groups $[x, p_1]$, $[p_1 + 1, p_2]$, ..., $[p_m + 1, y]$ $(x \le p_1 < p_2 < ... < p_m < y)$, in such a way that the machine can shave beavers in each group in one session. But then Beavershave 5000 needs $m + 1$ working sessions to shave beavers from x to y.

All beavers are restless and they keep trying to swap. So if we consider the problem more formally, we can consider queries of two types:

- what is the minimum number of sessions that Beavershave 5000 needs to shave beavers with ids from x to y, inclusive?
- two beavers on positions x and y (the beavers a_x and a_y) swapped.

You can assume that any beaver can be shaved any number of times.

Input

The first line contains integer n — the total number of beavers, $2 \le n$. The second line contains n space-separated integers — the initial beaver permutation.

The third line contains integer q — the number of queries, $1 \le q \le 10^5$. The next q lines contain the queries. Each query i looks as $p_i\, x_i\, y_i$, where p_i is the query type (1 is to shave beavers from x_i to y_i, inclusive, 2 is to swap beavers on positions x_i and y_i). All queries meet the condition: $1 \le x_i < y_i \le n$.

- to get 30 points, you need to solve the problem with constraints: $n \le 100$ (subproblem B1);
- to get 100 points, you need to solve the problem with constraints: $n \le 3 \cdot 10^5$ (subproblems B1+B2).

Note that the number of queries q is limited $1 \le q \le 10^5$ in both subproblem B1 and subproblem B2.

Output

For each query with $p_i = 1$, print the minimum number of Beavershave 5000 sessions.

Examples

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

```
2
1
3
5
```