Problem J. Jewel of Data Structure Problems

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

The array $a_1, a_2, \ldots, a_m$ of integers is called **odd** if it has an odd number of inversions, and **even** otherwise. Recall that an inversion is a pair (i, j) with $1 \le i < j \le m$ such that $a_i > a_j$. For example, in the array $[2, 4, 1, 3]$ there are 3 inversions: $(1, 3), (2, 3), (2, 4)$ (since $a_1 > a_3, a_2 > a_3, a_2 > a_4$), so it is **odd**.

Given a permutation $p_1, p_2, \ldots, p_n$ of integers from 1 to n, we call its beauty the length of its longest **odd** subsequence, if it exists, otherwise -1. For example, the beauty of the permutation $(1, 2, 3)$ is -1, because each of its subsequences is **even**, the beauty of $(4, 1, 2, 3)$ is 4, because the whole permutation is **odd**, and the beauty of $(4, 1, 3, 2)$ is 3, because the whole permutation is **even**, and the subsequence $(4, 3, 2)$ is **odd**.

We are given an initial permutation $p_1, p_2, \ldots, p_n$. There will be q update requests to it. After the i-th request we will have to swap p_{u_i} and p_{v_i}.

Find the beauty of the permutation after each request.

Recall that an array b is a subsequence of c if b can be obtained from c by removing some (possibly none or all) elements.

Recall that a permutation of the numbers from 1 to n is an array of length n containing each number from 1 to n exactly once.

Input

The first line contains two integers n, q ($1 \le n, q \le 2 \cdot 10^5$) — the permutation length and the number of queries, respectively.

The next line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$, all p_i are pairwise distinct) — the initial permutation p.

The i-th of the following q lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), indicating that after the i-th query you have to swap p_{u_i} and p_{v_i}.

Output

Print q integers — the permutation beauty after each update request.

Example

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

Note

- After the first query, the permutation is $(1, 2, 3, 4, 5)$. There is no **odd** subsequence in it.
- After the second query, the permutation is $(2, 1, 3, 4, 5)$. The whole permutation is **odd**, because it has exactly one inversion.

- After the third query, the permutation is $(4, 1, 3, 2, 5)$. The whole permutation is **even**, but its subsequence $(4, 3, 2, 5)$ is **odd**.
- After the fourth query, the permutation is $(1, 4, 3, 2, 5)$. The entire permutation is **odd**.
- After the fifth query, the permutation is $(1, 4, 5, 2, 3)$. The entire permutation is **even**, and all its subsequences of length 4 are **even**, but the subsequence $(1, 5, 2)$ is **odd**.
- After the sixth query, the permutation is $(5, 4, 1, 2, 3)$. The entire permutation is **odd**.