

Problem K. Keys

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 512 mebibytes

This is an interactive problem.

There are n keys and n closed locks, both are numbered from 1 to n . Each key i opens exactly one lock i . Consider the following algorithm of opening the locks:

1. Fix a permutation p_1, p_2, \dots, p_n denoting the order of keys.
2. Let x be the current key and y be the current lock. Initially, $x = 1$ and $y = 1$.
3. Try to open lock y by key p_x .
4. If the try was successful, increase y by 1, or finish the algorithm if y equals to n .
5. If x doesn't equal to n increase it by 1, otherwise set it to 1.
6. Go to step 3.

Step 3 takes one second, the others steps are instant.

Determine the total time required to open all the locks for $k + 1$ permutations: a given permutation p and k permutations that are constructed by reversing a given continuous segment of p .

Interaction Protocol

The jury's program outputs a sequence of queries, one after the other. Once the participant's program has printed an answer for a given query, the jury's program will proceed to output the next query.

Input

The first line contains two integers n and k ($2 \leq n \leq 100\,000$; $1 \leq k \leq 100\,000$).

The second line contains n distinct integers p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$).

Each of the next k lines contains two integers a and b ($1 \leq a < b \leq n$). This means a query with the permutation $p_1, p_2, \dots, p_{a-1}, p_b, p_{b-1}, \dots, p_{a+1}, p_a, p_{b+1}, \dots, p_{n-1}, p_n$.

Output

For each of the $k + 1$ given permutations, output a line with a single integer: the number of seconds required to open all the locks.

Examples

standard input	standard output
4 2 1 2 3 4 2 3 1 4	4 8 13
3 4 1 2 3 1 2 1 2 2 3 1 2	3 6 6 5 6