D. Little Elephant and Broken Sorting

The Little Elephant loves permutations of integers from 1 to n very much. But most of all he loves sorting them. To sort a permutation, the Little Elephant repeatedly swaps some elements. As a result, he must receive a permutation $1, 2, 3, ..., n$.

This time the Little Elephant has permutation $p_1, p_2, ..., p_n$. Its sorting program needs to make exactly m moves, during the i-th move it swaps elements that are at that moment located at the a_i-th and the b_i-th positions. But the Little Elephant's sorting program happened to break down and now on every step it can equiprobably either do nothing or swap the required elements.

Now the Little Elephant doesn't even hope that the program will sort the permutation, but he still wonders: if he runs the program and gets some permutation, how much will the result of sorting resemble the sorted one? For that help the Little Elephant find the mathematical expectation of the number of permutation inversions after all moves of the program are completed.

We'll call a pair of integers i, j $(1 \leq i < j \leq n)$ an *inversion* in permutatuon $p_1, p_2, ..., p_n$, if the following inequality holds: $p_i > p_j$.

Input

The first line contains two integers n and m $(1 \leq n, m \leq 1000, n > 1)$ — the permutation size and the number of moves. The second line contains n distinct integers, not exceeding n — the initial permutation. Next m lines each contain two integers: the i-th line contains integers a_i and b_i $(1 \leq a_i, b_i \leq n, a_i \neq b_i)$ — the positions of elements that were changed during the i-th move.

Output

In the only line print a single real number — the answer to the problem. The answer will be considered correct if its relative or absolute error does not exceed 10^{-6}.

Examples

input

```
2 1
1 2
1 2
```

output

```
0.500000000
```

input

```
4 3
1 3 2 4
1 2
2 3
1 4
```

output

```
3.000000000
```