

B. Problems for Round

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

There are n problems prepared for the next Codeforces round. They are arranged in ascending order by their difficulty, and no two problems have the same difficulty. Moreover, there are m pairs of similar problems. Authors want to split problems between two division according to the following rules:

- Problemset of each division should be non-empty.
- Each problem should be used in exactly one division (yes, it is unusual requirement).
- Each problem used in division 1 should be harder than any problem used in division 2.
- If two problems are similar, they should be used in different divisions.

Your goal is count the number of ways to split problem between two divisions and satisfy all the rules. Two ways to split problems are considered to be different if there is at least one problem that belongs to division 1 in one of them and to division 2 in the other.

Note, that the relation of similarity **is not** transitive. That is, if problem i is similar to problem j and problem j is similar to problem k , it doesn't follow that i is similar to k .

Input

The first line of the input contains two integers n and m ($2 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$) — the number of problems prepared for the round and the number of pairs of similar problems, respectively.

Each of the following m lines contains a pair of similar problems u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$). It's guaranteed, that no pair of problems meets twice in the input.

Output

Print one integer — the number of ways to split problems in two divisions.

Examples

input
5 2 1 4 5 2
output
2

input
3 3 1 2 2 3 1 3
output
0

input
3 2 3 1 3 2
output

Note

In the first sample, problems 1 and 2 should be used in division 2, while problems 4 and 5 in division 1. Problem 3 may be used either in division 1 or in division 2.

In the second sample, all pairs of problems are similar and there is no way to split problem between two divisions without breaking any rules.

Third sample reminds you that the similarity relation is not transitive. Problem 3 is similar to both 1 and 2, but 1 is not similar to 2, so they may be used together.