E. Cup Trick

The employees of the F company have lots of ways to entertain themselves. Today they invited a famous magician who shows a trick with plastic cups and a marble.

The point is to trick the spectator's attention. Initially, the spectator stands in front of a line of n plastic cups. Then the magician places a small marble under one cup and shuffles the cups. Then the spectator should guess which cup hides the marble.

But the head coder of the F company isn't easy to trick. When he saw the performance, he noticed several important facts:

- each cup contains a mark — a number from 1 to n; all marks on the cups are distinct;
- the magician shuffles the cups in m operations, each operation looks like that: take a cup marked x_i, sitting at position y_i in the row of cups (the positions are numbered from left to right, starting from 1) and shift it to the very beginning of the cup row (on the first position).

When the head coder came home after work he wanted to re-do the trick. Unfortunately, he didn't remember the starting or the final position of the cups. He only remembered which operations the magician performed. Help the coder: given the operations in the order they were made find at least one initial permutation of the cups that can go through the described operations in the given order. Otherwise, state that such permutation doesn't exist.

Input

The first line contains integers n and m $(1 \le n, m \le 10^6)$. Each of the next m lines contains a couple of integers. The i-th line contains integers x_i, y_i $(1 \le x_i, y_i \le n)$ — the description of the i-th operation of the magician. Note that the operations are given in the order in which the magician made them and the coder wants to make them in the same order.

Output

If the described permutation doesn't exist (the programmer remembered wrong operations), print -1. Otherwise, print n distinct integers, each from 1 to n: the i-th number should represent the mark on the cup that initially is in the row in position i.

If there are multiple correct answers, you should print the lexicographically minimum one.

Examples

input
```
2 1
2 1
```
output
```
2 1
```

input
```
3 2
1 2
1 1
```
output
```
2 1 3
```

input
3 3 1 3 2 3 1 3
output
-1

input
3 3 1 3 2 3 1 3
output
-1