In the nearby kindergarten they recently made up an attractive game of strength and agility that kids love.

The surface for the game is a large flat area divided into N×N squares.

The children lay large spongy cues onto the surface. The sides of the cubes are the same length as the sides of the squares. When a cube is put on the surface, its sides are aligned with some square. A cube may be put **on another cube** too.

Kids enjoy building forts and hiding them, but they always leave behind a huge mess. Because of this, prior to closing the kindergarten, the teachers rearrange **all** the cubes so that they occupy a rectangle on the surface, with **exactly one** cube on every square in the rectangle.

In one moving, a cube is taken off the top of a square to the top of any other square.

Write a program that, given the state of the surface, calculates the smallest number of moves needed to arrange all cubes into a rectangle.

INPUT

The first line contains the integers N and M ($1 \le N \le 100$, $1 \le M \le N^2$), the dimensions of the surface and the number of cubes currently on the surface.

Each of the following M lines contains two integers R and C ($1 \le R, C \le N$), the coordinates of the square that contains the cube.

OUTPUT

Output the smallest number of moves. A solution will always exist.

EXAMPLES

input	input	input
3 2 1 1 1 1	4 3 2 2 4 4 1 1	5 8 2 2 3 2 4 2 2 4 3 4 4 4 2 3 2 3
output	**output**	**output**
1	2	3

In the first example, it suffices to move one of the cubes from (1, 1) to (1, 2) or (2, 1).

In the third example, a cube is moved from (2, 3) to (3, 3), from (4, 2) to (2, 5) and from (4, 4) to (3, 5).