COCI '08 Regional #4 Tablica

Time limit: 1.0s **Memory limit:** 32M

Ivo has an $N \times N$ table. The table has the integers 1 through N^2 inscribed in row-major order. The following operations can be done on the table:

1. Rotate a row – all cells in a single row are rotated right, so that the number in the last column moves to the first.
2. Rotate a column – all cells in a single column are rotated down, so that the number in the last row moves to the first.

Ivo occasionally feels the urge to move a number X to cell (R, C) and proceeds as follows:

- While X is not in column C, rotate the row it is in.
- While X is not in row R, rotate the column it is in.

Here is an example of how to move number 6 to cell $(3, 4)$, start from the initial configuration:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	2	3	4
8	**5**	**6**	**7**
9	10	11	12
13	14	15	16

1	2	3	4
7	**8**	**5**	**6**
9	10	11	12
13	14	15	16

1	2	3	**16**
7	8	5	**4**
9	10	11	**6**
13	14	15	**12**

Ivo wants to move K numbers one after another. Write a program that calculates the number of rotations needed.

Input Specification

The first line contains two integers N $(2 \le N \le 10\,000)$ and K $(1 \le K \le 1000)$, the table dimension and the number of moves.

Each of the following K lines contains three integers X $(1 \le X \le N^2)$, R and C $(1 \le R, C \le N)$, the description of one move Ivo wants to make. Ivo does the moves in the order in which they are given.

Output Specification

Output K lines; for each move, output the number of rotations needed.

Sample Input 1

```
4 1
6 3 4
```

Sample Output 1

```
3
```

Sample Input 2

```
4 2
6 3 4
6 2 2
```

Sample Output 2

```
3
5
```

Sample Input 3

```
5 3
1 2 2
2 2 2
12 5 5
```

Sample Output 3

```
2
5
3
```