God Save the i-th Queen

`ith.c`, `ith.C`, `ith.java`, `ith.p`

Did you know that during the ACM-ICPC World Finals a big chessboard is installed every year and is available for the participants to play against each other? In this problem, we will test your basic chess-playing abilities to verify that you would not make a fool of yourself if you advance to the World Finals.

During the yesterday's Practice Session, you tried to solve the problem of N independent rooks. This time, let's concentrate on queens. As you probably know, the queens may move not only horizontally and vertically, but also diagonally.

You are given a chessboard with $i-1$ queens already placed and your task is to find all squares that may be used to place the i-th queen such that it cannot be captured by any of the others.

Input Specification

The input consists of several tasks. Each task begins with a line containing three integer numbers separated by a space: X, Y, N. X and Y give the chessboard size, $1 \leq X, Y \leq 20\,000$. $N = i-1$ is the number of queens already placed, $0 \leq N \leq X \cdot Y$.

After the first line, there are N lines, each containing two numbers x_k, y_k separated by a space. They give the position of the k-th queen, $1 \leq x_k \leq X$, $1 \leq y_k \leq Y$. You may assume that those positions are distinct, i.e., no two queens share the same square.

The last task is followed by a line containing three zeros.

Output Specification

For each task, output one line containing a single integer number: the number of squares which are not occupied and do not lie on the same row, column, or diagonal as any of the existing queens.

Sample Input

```
8 8 2
4 5
5 5
0 0 0
```

Output for Sample Input

```
20
```