CCC '06 S3 - Tin Can Telephone

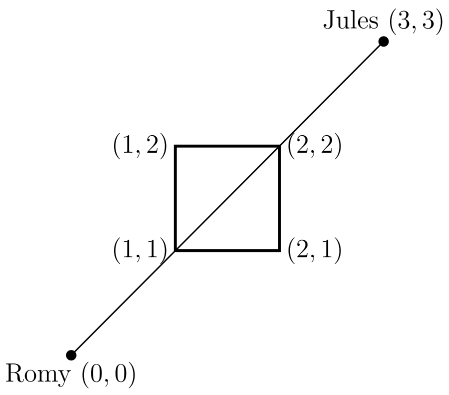
Time limit: 1.0s **Memory limit:** 256M

Canadian Computing Competition: 2006 Stage 1, Senior #3

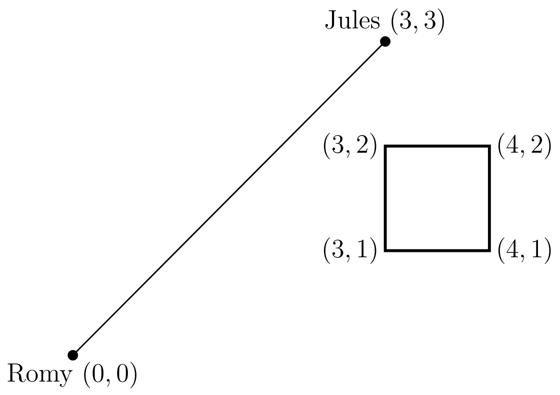
Romy and Jules have been talking with each other on their cell phones. Unfortunately, their parents dislike each other and have decided that they can no longer call each other. In fact, their parents have taken their cell phones away. So, Romy and Jules must find some other way to communicate. After searching the web for ideas, they have decided to build a "tin can" telephone.

Simply, a tin can telephone is two empty soup cans attached to each other with a string. To use it, the string must be stretched tight and then one person speaks while the other person listens. It is important that nothing touches the string so that it can vibrate and transfer sound from one can to the other.

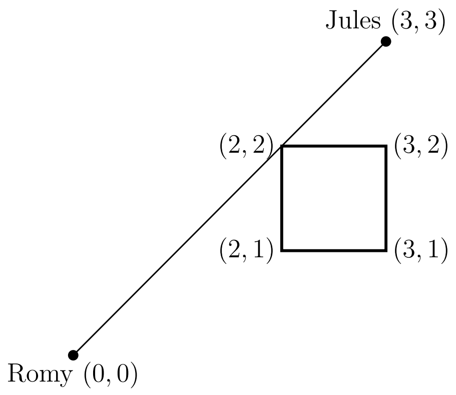
To successfully set up a tin can telephone, Romy and Jules are going to need a clear line of sight between their two bedroom windows. To determine if they can run the string between their rooms, they get out a map that uses simple integer coordinates. Now consider the following three situations:



1. Blocked View



2. Clear View



3. Blocked View (Touching)

In these figures, "Romy" is Romy's window (the grid coordinates $0, 0$) and "Jules" is Jules' window (grid coordinates $3, 3$). In the first figure, there is a building between their windows, and it blocks the line of sight between them. In the second figure, the building doesn't block their view and they can successfully set up a tin can phone. In the third figure, a line drawn from Romy's coordinates to Jules' coordinates would touch the corner of the building. Since the string cannot touch anything, they cannot set up a tin can telephone and the view is considered as blocked.

Input Specification

The input begins with a line containing four integer coordinates representing the locations of Romy's and Jules' windows. That is, the input x_R, y_R, x_J, y_J represents the coordinates (x_R, y_R) for Romy's window, and the coordinates (x_J, y_J) for Jules' window. You may assume that $-1000 \le x_R, x_J, y_R, y_J \le 1000$.

The next line contains a single integer, n $(0 \le n \le 100)$, identifying the number of buildings that will follow on the next n lines. Each building is on a separate line and begins with an integer specifying the number of corners that the building has. This integer is followed by the integer coordinates of the building's corners, in either clockwise or counter-clockwise order. No building has more than 32 corners.

The sample input and output, shown below, corresponds to the first "blocked" situation described previously.

Output Specification

For the input data, output a single number identifying the number of buildings that touch or block the line of sight.

Sample Input

```
0 0 3 3
1
4 1 2 2 2 2 1 1 1
```

Sample Output

```
1
```