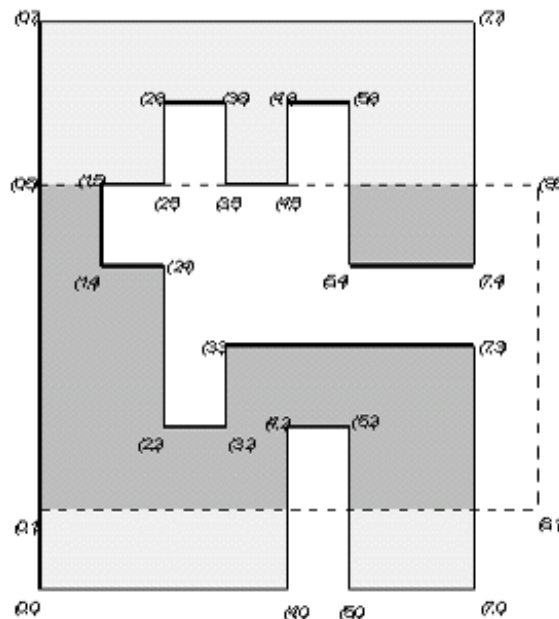


Window

We have a polygon chosen in the cartesian coordinate system. Sides of the polygon are parallel to the axes of coordinates. Every two consecutive sides are perpendicular and coordinates of every vertex are integers. We have also given a window that is a rectangle whose sides are parallel to the axes of coordinates. The interior of the polygon (but not its periphery) is coloured red. What is the number of separate red fragments of the polygon that can be seen through the window?

Example

Look at the figure below:



There are two separate fragments of the polygon that can be seen through the window.

Task

Write a program that for each test case:

- reads descriptions of a window and a polygon;
- computes the number of separate red fragments of the polygon that can be seen through the window;
- outputs the result.

Input

The number of test cases t is in the first line of input, then t test cases follow separated by an empty line.

In the first line of a test case there are four integers x_1, y_1, x_2, y_2 from the range $[0..10000]$, separated by single spaces. The numbers x_1, y_1 are the coordinates of the top-left corner of the window. The numbers x_2, y_2 are the coordinates of the bottom-right corner of the window. The next line of the input file contains one integer n , $4 \leq n \leq 5000$, which equals the number of

vertices of the polygon. In the following n lines there are coordinates of polygon's vertices given in anticlockwise direction, i.e. the interior of the polygon is on the left side of its periphery when we move along the sides of the polygon according to the given order. Each line contains two integers x, y separated by a single space, $0 \leq x \leq 10000$, $0 \leq y \leq 10000$. The numbers in the i -th line, are coordinates of the i -th vertex of the polygon.

Output

For each test case you should output one line with the number of separate red fragments of the polygon that can be seen through the window.

Example

Sample input:

```
1
0 5 8 1
24
0 0
4 0
4 2
5 2
5 0
7 0
7 3
3 3
3 2
2 2
2 4
1 4
1 5
2 5
2 6
3 6
3 5
4 5
4 6
5 6
5 4
7 4
7 7
0 7
```

Sample output:

```
2
```