

Homemade Asteroids

Pew pew pew!

Everyone loves Asteroids, the classic arcade game involving senselessly blasting asteroids into submission with a spaceship. In fact, you love it so much that you built your very own version to play at home! Unfortunately, it sucks.

Your version of the game is played on a 2D plane, containing your ship (a dot at coordinates (X_S, Y_S)) and N ($1 \leq N \leq 1000$) stationary, triangular, positive-area asteroids. The i th asteroid has vertices at coordinates (X_{Ai}, Y_{Ai}) , (X_{Bi}, Y_{Bi}) , and (X_{Ci}, Y_{Ci}) . All coordinates in the input are integers with absolute values no greater than 10^9 , and no two objects occupy any of the same space (even on their edges or vertices).

Your game only permits you to fire a single missile, which travels in a straight line, destroying every asteroid that it comes in contact with (even on its edges or vertices). However, it doesn't exactly move very smoothly - instead, it starts at your ship at frame 0, and after every frame, its x-coordinate increases by X_D , and its y-coordinate by Y_D . These variables also have absolute values no greater than 10^9 , and at least one of them is guaranteed to be non-zero. After frame F ($1 \leq F \leq 1000$), the missile simply disappears.

There are T ($1 \leq T \leq 20$) scenarios as described above. For each, you'd like to predict how many different asteroids your missile will be able to take out before frame $F+1$.

Input

First line: 1 integer, T

For each scenario:

First line: 2 integers, N and F

Second line: 4 integers, X_S , Y_S , X_D , and Y_D

Next N lines: 6 integers, X_{Ai} , Y_{Ai} , X_{Bi} , Y_{Bi} , X_{Ci} , and Y_{Ci} , for $i = 1..N$

Output

For each scenario:

1 integer, the number of asteroids that will be destroyed by the missile

Example

Input:

```
1
4 4
4 17 4 -2
```

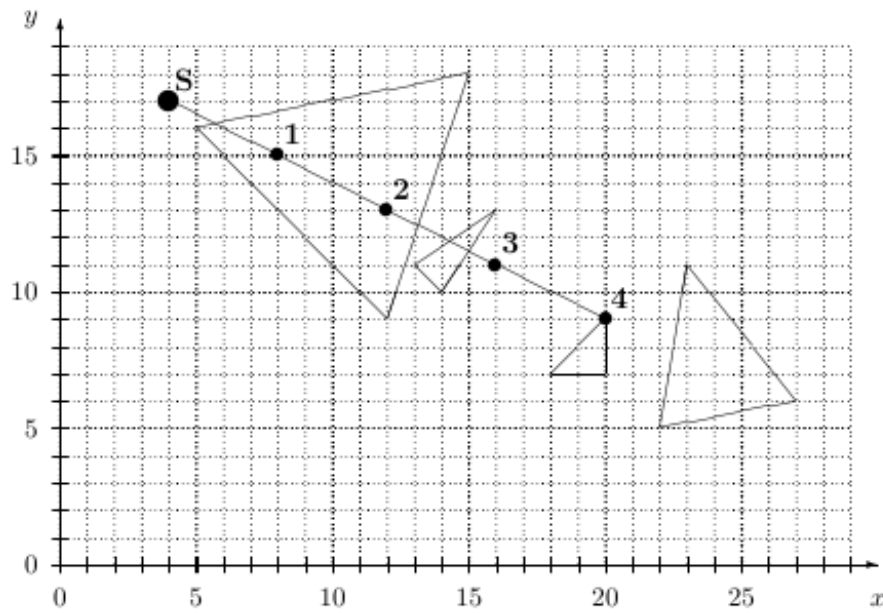
5 16 15 18 12 9
16 13 13 11 14 10
20 9 20 7 18 7
22 5 23 11 27 6

Output:

2

Explanation of Sample:

The following grid shows the layout of the game, with your ship marked with an "S", and the missile's location at each frame marked with that frame's number:



As can be seen, the missile destroys the first asteroid during frame 1, and then the third asteroid during frame 4. It does not destroy the second asteroid, even though its line of fire goes through it, as it does not intersect the asteroid during any of the frames. It also doesn't destroy the last asteroid, as it stops travelling after frame 4.