

Problem G: Battlecheap

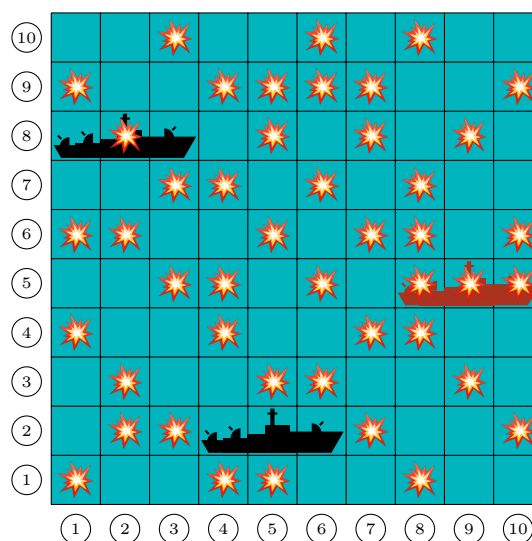
Time limit: 3 seconds

You are bored to death during a very long flight. While surfing through the collection of crappy video games provided by the in-flight entertainment system, you have stumbled upon something interesting: the game *Battleship*! Remember that this is a game for two players played on a 10 by 10 grid; your goal is to position your fleet and try to sink the enemy ships before your own are sunk.

So wouldn't it be fun to try play *Battleship* while you are flying in a plane? Unfortunately, you'll soon discover that it isn't that much fun after all and the in-flight video game is just a low-budget and buggy clone with several key shortcomings:

- You start by placing your cruiser which is 3 squares long and 1 square wide, but then the remote control doesn't seem to have the button to move to the next ship, so after a while the game automatically starts with just that ship in place.
- The computer can fire at your ships during its turn, but when it's your turn, the button to fire at the computer doesn't seem to be mapped on the remote control either. So after a while you lose your turn and the computer shoots at you again.
- Fortunately, however, the computer seems to have a limit on how many times it plays. After a certain number of shots, if you haven't lost yet (i.e. not all the squares of your cruiser have been hit), then the computer just gives up and declares you as the winner.
- As it turns out, the game's RNG is also completely buggy; from one game to the next, the computer always fires the exact same sequence of squares, no matter what happens or what it hits or where your cruiser was placed.

Of course, as soon as you have figured out the computer's hardcoded sequence of moves, this makes the game an utter disappointment. Still, you are stuck on the plane and you start to wonder, knowing this sequence, how many ways (if any) do you have to win, i.e. how many ways are there to position the cruiser and survive until the computer gives up?



Input

The input file consists of multiple test cases. The first line of the input file consists of a single integer indicating the number of test cases. Each test case follows. The first line consists of a single integer $0 \leq n \leq 100$ indicating how many times the computer fires before giving up. The next n lines describe the successive squares where the computer fires: the i^{th} line consists of two integers $1 \leq r_i \leq 10$ and $1 \leq c_i \leq 10$ giving the row and column of the computer's i^{th} shot. It is guaranteed that the computer never fires twice at the same position in its sequence.

Output

For each test case in the input, your program should produce one line consisting of one integer indicating in how many different ways you can position your cruiser at the beginning of the game and survive until the end. The cruiser must be positioned on 3 squares that are vertically adjacent or horizontally adjacent, and you survive unless the computer's sequence hits all the squares where the cruiser is placed.

Sample Input

```
2
0
10
3 2
4 7
2 2
3 6
3 3
4 9
3 4
3 7
4 8
3 5
```

Sample Output

```
160
155
```