Problem ID: balloongrid

You are in a pub with your friends, planning your next trip around the world together, when a strangely dressed woman approaches you. “I have something you might be interested in. You look just the type,” she says. She then continues to tell you about the most famous programming contest in the world: The Taurida Subregional 2017. Fingers will be burned and minds broken during that contest! It is held outside, rain or shine, but a real programmer can take it.

Of course you would like to travel to this *Taurida* as soon as possible and cheer for the glorious contestants! Your heart drops as you figure out where it will be: Crimea. Foreign office strictly advises against all travel to that region. Fortunately, all programming contests award teams with a balloon for each problem they solve, so you can just fly directly over the contest site and compute the scoreboard by counting the teams’ balloons.

Conveniently, the teams’ work stations are arranged in a 10x10 grid and each work station is 5m by 5m. There are 100 teams, numbered 0 to 99 in row-major order to give them unique IDs. You have quickly measured all the balloons’ coordinates, now you just need to write a program that computes the number of solved problems per team!

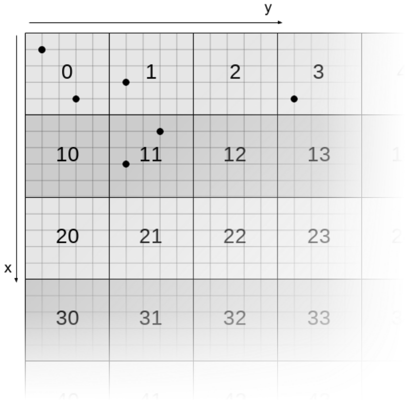

Figure 1: Visualization of Sample Input 1.

Input

The input consists of:

- One line with one integer n ($1 \leq n \leq 1\,000$), the number of balloons.
- n lines, each with two integers x, y ($0 < x, y < 50$), specifying a balloon at row x and column y.

It is guaranteed that no balloon’s coordinates lie on a line separating two adjacent work stations.

Output

For every team that has solved at least one problem, print one line with two integers, the first being the team’s ID and the second being the number of problems they have solved so far.

Sample Input 1

```
6
1 1
4 3
3 6
4 16
6 8
8 6
```

Sample Output 1

```
0 2
1 1
3 1
11 2
```

Sample Input 2

```
3
1 1
1 1
3 3
```

Sample Output 2

```
0 3
```