

构成的正方形数量

题目描述：

输入 N 个互不相同的二维整数坐标，求这 N 个坐标可以构成的正方形数量。(内积为零的两个向量垂直)

输入描述：

第一行输入为 N ， N 代表坐标数量， N 为正整数。 $N \leq 100$

之后的 K 行输入为坐标 $x\ y$ 以空格分隔， x, y 为整数， $-10 \leq x, y \leq 10$

输出描述：

输出可以构成的正方形数量

示例 1

输入：

3

1 3

2 4

3 1

输出：

0

说明：

3 个点不足以构成正方形

示例 2

输入：

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

输出：

```
1
```

说明：

此 4 点可构成正方形

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import java.util.Scanner;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
```

```
        int num = Integer.parseInt(in.nextLine());
```

```
        List<String> list = new ArrayList<>();
```

```
        for (int i = 0; i < num; i++) {
```

```
            list.add(in.nextLine());
```

```
        }
```

```
        int count = 0;
```

```
        if(num <= 3){
```

```
            System.out.println(count);
```

```
            return;
```

```

}

for (int i = 0; i < list.size(); i++) {

    String str = list.get(i);

    String[] split = str.split(" ");

    int x1 = Integer.parseInt(split[0]);

    int y1 = Integer.parseInt(split[1]);

    for (int j = i + 1; j < list.size(); j++) {

        String str1 = list.get(j);

        String[] split1 = str1.split(" ");

        int x2 = Integer.parseInt(split1[0]);

        int y2 = Integer.parseInt(split1[1]);


        int x31 = x1 - (y1-y2);

        int y31 = y1 + (x1-x2);

        int x41 = x2 - (y1-y2);

        int y41 = y2 + (x1-x2);


        int x32 = x1 + (y1-y2);

        int y32 = y1 - (x1-x2);

        int x42 = x2 + (y1-y2);

        int y42 = y2 - (x1-x2);
    }
}

```

```
        if(list.contains(x31 + "" + y31) && list.contains(x41 + "" + y41)){

            count++;

        }

        if(list.contains(x32 + "" + y32) && list.contains(x42 + "" + y42)){

            count++;

        }

    }

    System.out.println(count/4);

}

}
```