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 <= 100
之后的 K 行输入为坐标 x y以空格分隔，x, y 为整数, -10<=x, y <= 10

输出描述：输出可以构成的正方形数量

补充说明：

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

```
const rl = require("readline").createInterface({ input: process.stdin });
var iter = rl[Symbol.asyncIterator]();
const readline = async () => (await iter.next()).value;

void (async function () {
    function isZhengfang(a, b, c, d) {
        let list = [a, b, c, d];
        let map = new Map();
        let tmp = 0;
        for (let i = 0; i < 3; i++) {
            for (let j = i + 1; j < 4; j++) {
                let l1 = list[i][0] - list[j][0];
                let l2 = list[i][1] - list[j][1];
                let result = l1 * l1 + l2 * l2;
                tmp = result;
                map.set(result, (map.get(result) || 0) + 1);
            }
        }
        return map.size == 2 && (map.get(tmp) == 2 || map.get(tmp) == 4);
    }
```

```
    function count(dots) {
        let sum = 0;
        let len = dots.length;
        if (len > 3) {
            for (let x = 0; x < len - 3; x++) {
                for (let y = x + 1; y < len - 2; y++) {
                    for (let z = y + 1; z < len - 1; z++) {
                        for (let q = z + 1; q < len; q++) {
                            if (isZhengfang(dots[x], dots[y], dots[z], dots[q]))
                                sum++;
                        }
                    }
                }
            }
        }
        console.log(sum);
    }
    // Write your code here
    while ((line = await readline())) {
        let number = Number(line);
        let arr = [];
        for (let i = 0; i < number; i++) {
            let dot = (await readline()).split(" ").map(Number);
            arr.push(dot);
        }
        count(arr)
    }
})();
```