

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

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

```
#include <iostream>
using namespace std;
int a[200][200],x[128],y[128];
int main() {
    //freopen("0.in","r",stdin);
    int i,j,k,n,ans=0;
    cin>>n;
    for(i=0;i<n;++i){
        cin>>x[i]>>y[i];
        a[x[i]+100][y[i]+100]++;
    }
    for(i=0;i<n;++i)
    for(j=0;j<n;++j)
    for(k=j+1;k<n;++k)
    if((x[i]!=x[j] || y[i]!=y[j])&&(x[i]!=x[k] || y[i]!=y[k])&&
(x[j]-x[i])*(x[k]-x[i])+(y[j]-y[i])*(y[k]-y[i])==0&&
(x[j]-x[i])*(x[j]-x[i])+(y[j]-y[i])*(y[j]-y[i])==(x[k]-x[i])*(x[k]-x[i])+(y[k]-y[i])*(y[k]-y[i]))){
        int tx=x[j]+x[k]-x[i],ty=y[j]+y[k]-y[i];
        ans+=a[tx+100][ty+100];
    }
    cout<<ans/4;
    return 0;
}
```