

# 小水獭和三点共线

## 题目描述

小水獭正在补习计算几何。

二维平面上有  $n$  个不同的点，小水獭请你帮它判断是否存在三点不共线？

## 输入格式

第一行一个正整数  $t$  ( $1 \leq t \leq 10$ )，表示数据组数。

对于每组数据，第一行一个正整数  $n$  ( $3 \leq n \leq 10^5$ )，含义同题目描述。

接下来  $n$  行，每行两个整数  $x_i, y_i$  ( $-10^9 \leq x_i, y_i \leq 10^9$ )，表示第  $i$  个点的坐标。

保证任意两点坐标不同。

## 输出格式

对于每组数据，输出一行一个字符串：

- 如果存在三点不共线，输出 `how?`；
- 否则，输出 `boo how! boo how!`。

## 题目分析

- 题目要求对二维平面上  $n$  个不同的点，判断其中是否存在三点不共线。可将题目转化为，任取这  $n$  个点中的两个，记作  $p_1$  和  $p_2$ ，分别观察其他的  $n-2$  个点是否与  $p_1$ 、 $p_2$  共线，若这  $n-2$  个点均与  $p_1$ 、 $p_2$  共线，则  $n$  个点均共线；否则，存在三点不共线。
- 可以使用一个结构体来记录点的坐标

```
struct Point {
    int x,y;
} p[maxn],p1,p2;
```

判断三点是否共线，即判断两条线段的斜率是否相等，将斜率相等的式子稍作转换

$$(p[i].x-p1.x)*(p1.y-p2.y) == (p1.x-p2.x)*(p[i].y-p1.y)$$

对于题目中待判断的  $n-2$  个点，用一个整型 `ans` 记录结果

```
for (int i=3;i<=n;i++)
{
    if ((p[i].x-p1.x)*(p1.y-p2.y) != (p1.x-p2.x)*(p[i].y-p1.y))
        ans=0;
}
```

# 代码

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;

const int maxn = 100005;
struct Point {
    int x,y;
} p[maxn],p1,p2;

int main() {
    int t,n;
    scanf("%d",&t);
    while(t--) {
        scanf("%d",&n);
        for(int i = 1; i <= n; i++)
            scanf("%d%d",&p[i].x,&p[i].y);
        int ans=1;
        p1=p[1];
        p2=p[2];
        for (int i=3;i<=n;i++)
        {
            if ((p[i].x-p1.x)*(p1.y-p2.y) != (p1.x-p2.x)*(p[i].y-p1.y))
                ans=0;
        }
        if (ans)
            printf("boo how! boo how!\n");
        else
            printf("how?\n");
    }
    return 0;
}
```