

题目

B 小水獭和三点共线

时间限制：1000ms 内存限制：65536kb

通过率：170/188 (90.43%) 正确率：170/520 (32.69%)

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 ($3 \leq n \leq 105$)，含义同题目描述。

接下来 n 行，每行两个整数 x_i, y_i ($-109 \leq x_i, y_i \leq 109$)，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!`。

输入样例

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

输出样例

```
boo how! boo how!  
how?
```

思路

这道题思路比较清晰，先找出两个点之后将其余n-2个点遍历，通过叉积可以判断三点是否在一条直线上，如果出现叉积不为0的情况则存在三点不共线，反之则全部共线。

时间复杂度

$O(n)$

程序示例

```
#include <iostream>  
using namespace std;  
typedef long long ll;  
struct vertex  
{  
    ll x, y;  
};  
ll crossMul(vertex p0, vertex p1, vertex p2)  
{  
    return ((p1.x - p0.x) * (p2.y - p0.y) - (p1.y - p0.y) * (p2.x - p0.x));  
}  
vertex list[100005];  
int main(int argc, char const *argv[])  
{  
    ll t;  
    cin >> t;  
    int n;  
    while (t--)  
    {  
        scanf("%d", &n);  
        for(int i = 1; i <= n; i++)  
        {  
            cin >> list[i].x >> list[i].y;  
        }  
        bool flag = true;  
        for(int i = 3; i <= n; i++)  
        {  
            if(crossMul(list[1], list[2], list[i]) != 0)  
            {  
                flag = false;  
                break;  
            }  
        }  
        if(flag)  
        {  
            printf("boo how! boo how!\n");  
        }  
        else  
        {  
            ;  
        }  
    }  
}
```

```
        printf("how?\n");  
    }  
}  
return 0;  
}
```