B题题解

题目分析：

判断是否存在三点不共线，用叉积可以很方便地判断，只要取前两组向量相减得到基础向量，之后直接相乘即可。

易错点：

1. 题目看着不难，但是但是做题的时候卡了好一阵子……问题出在判断上，图方便在确定基准向量后直接对读入的向量做叉积判断，然后就直接break出去输出了***，改成flag就秒过了。
2. 注意数据范围，叉积开long long

代码：

```
#include <bits/stdc++.h>

#include <cmath>
#include <cstring>
#include <queue>
using namespace std;
#define lld long long
#define MAX_ARC 250010
#define MAX_VEX 510
#define INF_LLD 0x3f3f3f3f3f3f3f3f
#define INF_INT 0x3f3f3f3f
#define QC std::ios::sync_with_stdio(false), cin.tie(0)
struct vec {
  lld x, y;
};
long long Cross(vec A, vec B) {
  long long s = (long long)A.x * (long long)B.y;
  long long t = (long long)A.y * (long long)B.x;
  return s - t;  // A->B左转为正
}
int main() {
  QC;
  int t;
  cin >> t;
  while (t--) {
    int n;
    cin >> n;
    vec a, b, s;
    cin >> a.x >> a.y >> b.x >> b.y;
    s.x = b.x - a.x;
    s.y = b.y - a.y;
    bool flag = false;
    for (size_t i = 0; i < n - 2; i++) {
      cin >> b.x >> b.y;
      b.x -= a.x;
```

```
      b.y -= a.y;
      if (Cross(s, b)) {
        flag = true;
      }
    }
    if (flag)
      cout << "how?" << endl;
    else
      cout << "boo how! boo how!" << endl;
  }

  return 0;
}
```