```cpp
/*****88  point in polygon ******/
#include <cstdio>
#include <cassert>
#include <algorithm>
using namespace std;
typedef struct { int x, y; } Point;
Point p[100];
inline int direction(const Point &p1, const Point &p2, const Point &p3) {
    return p1.x * (p2.y - p3.y) + p2.x * (p3.y - p1.y) + p3.x * (p1.y - p2.y);
}

inline bool onsegment(const Point &p1, const Point &p2, const Point &p3) {
    Point pmn, pmx;
    pmn.x = min(p1.x, p2.x), pmn.y = min(p1.y, p2.y);
    pmx.x = max(p1.x, p2.x), pmx.y = max(p1.y, p2.y);
    return pmn.x <= p3.x && p3.x <= pmx.x && pmn.y <= p3.y && p3.y <= pmx.y;
}

inline bool intersect(const Point &p1, const Point &p2, const Point &p3, const Point &p4, bool &on) {
    int d1, d2, d3, d4;
    d1 = direction(p3, p4, p1);
    d2 = direction(p3, p4, p2);
    d3 = direction(p1, p2, p3);
    d4 = direction(p1, p2, p4);
    on = false;
    if(((d1 < 0 && d2 > 0) || (d1 > 0 && d2 < 0)) && ((d3 < 0 && d4 > 0) || (d3 > 0 && d4 < 0))) return true;
    if(!d3 && onsegment(p1, p2, p3)) { on = true; return true; }
    if(!d4 && onsegment(p1, p2, p4)) return true;
    if(!d1 && onsegment(p3, p4, p1)) return true;
    if(!d2 && onsegment(p3, p4, p2)) return true;
    return false;
}

int main() {
    int test, cs, cnt, i, n, q;
    Point p0, p1;
    bool on;
    scanf("%d", &test);
    for(cs = 1; cs <= test; cs++) {
        scanf("%d", &n);
        for(i = 0; i < n; i++) scanf("%d %d", &p[i].x, &p[i].y);
        p[n] = p[0];
        printf("Case %d:\n", cs);
        scanf("%d", &q);
        while(q--) {
            scanf("%d %d", &p0.x, &p0.y);
            p1.x = p0.x + 30000, p1.y = p0.y+30001;
            for(i = cnt = 0; i < n; i++) {
                if(intersect(p[i], p[i+1], p0, p1, on)) {
                    if(on) { cnt = 1; break; }
                    cnt++;
                }
            }
            if(cnt & 1) puts("Yes");
            else puts("No");
```

```
56              }
57          }
58          return 0;
59      }
```