```cpp
#include <bits/stdc++.h>
using namespace std;
#define MOD (int)(1e9+7)
#define ll long long
```

Useful define:

```cpp
#define pi 3.14159265358979323846
```

Length of the union of segments:

```cpp
//given n segments on a line, each described by a pair of coordinates
//we have to find the length of their union

int length_union(const vector<pair<int, int>> &a) {
    int n = a.size();
    vector<pair<int, bool>> x(n*2);
    for (int i = 0; i < n; i++) {
        x[i*2] = {a[i].first, false};
        x[i*2+1] = {a[i].second, true};
    }

    sort(x.begin(), x.end());

    int result = 0;
    int c = 0;
    for (int i = 0; i < n * 2; i++) {
        if (i > 0 && x[i].first > x[i-1].first && c > 0)
            result += x[i].first - x[i-1].first;
        if (x[i].second)
            c--;
        else
            c++;
    }
    return result;
}
```

Area of triangle:

```cpp
int signed_area_parallelogram(point2d p1, point2d p2, point2d p3) {
    return cross(p2 - p1, p3 - p2);
}

double triangle_area(point2d p1, point2d p2, point2d p3) {
    return abs(signed_area_parallelogram(p1, p2, p3)) / 2.0;
}

bool clockwise(point2d p1, point2d p2, point2d p3) {
    return signed_area_parallelogram(p1, p2, p3) < 0;
}

bool counter_clockwise(point2d p1, point2d p2, point2d p3) {
    return signed_area_parallelogram(p1, p2, p3) > 0;
}

//or
double triangleArea(double l1 , double l2 , double l3){
    double s =(l1+l2+l3)/2.0;
    double ans = sqrtl(s*(s-l1)*(s-l2)*(s-l3));
    return ans;
}
```