```
/*Write C++ program to implement Cohen Southerland line clipping algorithm*/

#include <iostream>
#include <graphics.h>

// Constants for the region codes
const int INSIDE = 0; // 0000
const int LEFT = 1;   // 0001
const int RIGHT = 2;  // 0010
const int BOTTOM = 4; // 0100
const int TOP = 8;    // 1000

// Window boundaries
const int x_min = 100;
const int y_min = 100;
const int x_max = 400;
const int y_max = 300;

// Function to compute the region code for a point (x, y)
int computeCode(double x, double y) {
    int code = INSIDE;
    if (x < x_min) {
        code |= LEFT;
    } else if (x > x_max) {
        code |= RIGHT;
    }
    if (y < y_min) {
        code |= BOTTOM;
    } else if (y > y_max) {
        code |= TOP;
    }
    return code;
}

// Cohen-Sutherland line clipping algorithm
void cohenSutherlandClip(double x1, double y1, double x2, double y2) {
    int code1 = computeCode(x1, y1);
    int code2 = computeCode(x2, y2);

    bool accept = false;

    while (true) {
        if ((code1 | code2) == 0) {
            accept = true;
            break;
        } else if (code1 & code2) {
            break;
        } else {
            double x, y;
            int codeOut = code1 ? code1 : code2;
            if (codeOut & TOP) {
                x = x1 + (x2 - x1) * (y_max - y1) / (y2 - y1);
                y = y_max;
            } else if (codeOut & BOTTOM) {
                x = x1 + (x2 - x1) * (y_min - y1) / (y2 - y1);
                y = y_min;
            } else if (codeOut & RIGHT) {
                y = y1 + (y2 - y1) * (x_max - x1) / (x2 - x1);
                x = x_max;
            } else if (codeOut & LEFT) {
                y = y1 + (y2 - y1) * (x_min - x1) / (x2 - x1);
                x = x_min;
            }
            if (codeOut == code1) {
                x1 = x;
                y1 = y;
                code1 = computeCode(x1, y1);
            } else {
                x2 = x;
                y2 = y;
                code2 = computeCode(x2, y2);
            }
        }
    }
    if (accept) {
        setcolor(GREEN);
        line(x1, y1, x2, y2);
    }
}

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, nullptr);

    rectangle(x_min, y_min, x_max, y_max);

    double x1 = 50, y1 = 150, x2 = 450, y2 = 250;
    setcolor(RED);
    line(x1, y1, x2, y2);

    cohenSutherlandClip(x1, y1, x2, y2);

    getch();
    closegraph();
    return 0;
}
```