```
#include<graphics.h>
#include<bits/stdc++.h>
using namespace std;

const int INSIDE = 0;
const int LEFT = 1;
const int RIGHT = 2;
const int BOTTOM = 4;
const int TOP = 8;

const int x_max = 10;
const int y_max = 8;
const int x_min = 4;
const int y_min = 4;

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;
}


void cohenSutherlandClip(int x1, int y1, int x2, int y2)
{
    int code1 = computeCode(x1, y1);
    int code2 = computeCode(x2, y2);
    bool accept = false;
    while (true)
    {
        if ((code1 == 0) && (code2 == 0)) {
            // If both endpoints lie within rectangle
            accept = true;
            break;
        }
        else if (code1 & code2) {
            // If both endpoints are outside rectangle in same region
            break;
        }
```

```
else {
    // Some segment of line lies within the rectangle
    int code_out;
    double x, y;

    // At least one endpoint is outside the rectangle, pick it.
    if (code1 != 0)
        code_out = code1;
    else
        code_out = code2;


    // Find intersection point; using formulas y = y1 + slope * (x
    if (code_out & TOP) {
        // point is above the clip rectangle
        x = x1 + (x2 - x1) * (y_max - y1) / (y2 - y1);
        y = y_max;
    }
    else if (code_out & BOTTOM) {
        // point is below the rectangle
        x = x1 + (x2 - x1) * (y_min - y1) / (y2 - y1);
        y = y_min;
    }
    else if (code_out & RIGHT) {
        // point is to the right of rectangle
        y = y1 + (y2 - y1) * (x_max - x1) / (x2 - x1);
        x = x_max;
    }
    else if (code_out & LEFT) {
        // point is to the left of rectangle
        y = y1 + (y2 - y1) * (x_min - x1) / (x2 - x1);
        x = x_min;
    }


    // Now intersection point x, y is found
    // We replace point outside rectangle by intersection point
    if (code_out == code1) {
        x1 = x;
        y1 = y;
        code1 = computeCode(x1, y1);
    }
    else {
        x2 = x;
        y2 = y;
        code2 = computeCode(x2, y2);
```

```
            }
        }
    }

    if (accept) {
        cout << "Line accepted from " << x1 << ", " << y1 << " to " << x2
        // Here the user can add code to display the rectangle along with
    }
    else
        cout << "Line rejected" << endl;


}


int main()
{
    initwindow(400,300,"Clipping");
    int X,Y;
    X = getmaxx();
    Y = getmaxy();
    rectangle(0,0,X,Y);
    line(X/2,0,X/2,Y);
    line(0,Y/2,X,Y/2);

    int xn = X/2;
    int yn = Y/2;

    rectangle(xn+0,yn-100,xn+80,yn-40);

    line(xn-20,yn-90,xn+100,yn-30);
    line(xn-30,yn-60,xn+40,yn-120);

    cohenSutherlandClip(5, 5, 7, 7);
    cohenSutherlandClip(7, 9, 11, 4);
    cohenSutherlandClip(1, 5, 4, 1);

    while(!kbhit())
    {
        delay(200);
    }
    getch();
    return 0;
}
```