

**SSN COLLEGE OF ENGINEERING**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**  
**EX NO: 7 – Cohen Sutherland Line Clipping Algorithm**

---

Name : Mohamed Hashim G  
RegNo : 185001094

---

**AIM**

To write a program in C++ using OpenGL to perform line clipping using Cohen Sutherland Line Clipping Algorithm.

**ALGORITHM:**

1. Read the endpoints (x1,y1),(x2,y2) and window co-ordinates xwmin,xwmax,ywmin,ywmax.
2. Calculate region code for endpoints(x,y) as follows:
3. Calculate slope m as (y2-y1)/(x2-x1).
4. Region code of (x1,y1) is c1 and region code of (x2,y2) is c2.
5. If c1==0 and c2==0 : Line Completely inside
6. If c1&c2 == 0 : Line partially inside and partially outside.
  - a. Find the window boundary where line intersects and calculate new endpoints.
7. If c1&c2 > 0:Line completely outside.

**CODE:**

```
#include<gl/glut.h>
#include<iostream>
#include<utility>
using namespace std;

pair<int, int> P1, P2;
int X1, X2, Y1, Y2;
int xwmin, xwmax, ywmin, ywmax;

void myInit()
{
    glClearColor(0, 0, 0, 1);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPointSize(10);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 1000.0, 0.0, 1000.0);
}
```

```

void drawWindow()
{
    glBegin(GL_LINE_LOOP);
    glVertex2d(xwmin, ywmin);
    glVertex2d(xwmax, ywmin);
    glVertex2d(xwmax, ywmax);
    glVertex2d(xwmin, ywmax);
    glEnd();
}

void drawOriginal()
{
    glBegin(GL_LINES);
    glVertex2d(P1.first, P1.second);
    glVertex2d(P2.first, P2.second);
    glEnd();
}

int getRC(pair<int, int>& P)
{
    int rc = 0;
    if (P.first < xwmin) rc |= 1;
    else if (P.first > xwmax) rc |= 1 << 1;
    if (P.second < ywmin) rc |= 1 << 2;
    else if (P.second > ywmax) rc |= 1 << 3;
    return rc;
}

void findIntersection(pair<int, int>& P, double m, int rc) {

    if (rc == 0) return;
    // y = ywmax
    if ((rc >> 3)&1) {
        //x = X1 + (y-Y1)/m
        P.second = ywmax;
        P.first = X1 + (ywmax - Y1) / m;
        return;
    }
    //y = ywmin
    if ((rc >> 2) & 1) {
        //x = X1 + (y-Y1)/m
        P.second = ywmin;
        P.first = X1 + (ywmin - Y1) / m;
        return;
    }
    // x= xwmax
    if ((rc >> 1) & 1) {
        //y = Y1 + (x-X1)*m
        P.first = xwmax;

```

```

        P.second = Y1 + (xwmax - X1) * m;
        return;
    }
    // x= xwmin
    if (rc & 1) {
        //y =Y1 + (x-X1)*m
        P.first = xwmin;
        P.second = Y1 + (xwmin - X1) * m;
        return;
    }
}

void PerformClipping(pair<int, int>& P1, pair<int, int>& P2)
{
    int rc1 = getRC(P1), rc2 = getRC(P2);
    //Checking for trivial OR
    if (int(rc1 | rc2) == 0) {

        glBegin(GL_LINES);
        glVertex2d(P1.first, P1.second);
        glVertex2d(P2.first, P2.second);
        glEnd();
        return;
    }
    else if (int(rc1 & rc2) != 0) return;
    double m = (Y2-Y1) * 1.0 / (X2-X1);
    findIntersection(P1, m, rc1);
    findIntersection(P2, m, rc2);
    PerformClipping(P1, P2);
}

void myDisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0f, 0.0f, 1.0f);
    drawWindow();
    glColor3f(0.0f, 1.0f, 0.0f);
    drawOriginal();
    glColor3f(1.0f, 0.0f, 0.0f);
    PerformClipping(P1, P2);
    glFlush();
}

int main(int argc, char* argv[])
{
    cout << "Enter window properties:" << endl;
    cout << "XWmin YWmin: ";
    cin >> xwmin >> ywmin;
    cout << "XWmax YWmax: ";
    cin >> xwmax >> ywmax;
}

```

```

int x, y;
cout << endl << "Enter point p1(x,y) :";
cin >> x >> y;
P1.first = x;
P1.second = y;
X1 = x;
Y1 = y;
cout << "Enter point p2(x,y) :";
cin >> x >> y;
P2.first = x;
P2.second = y;
X2 = x;
Y2 = y;

glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowPosition(750, 0);
glutInitWindowSize(750, 750);
glutCreateWindow("7-Cohen Sutherland");
glClearColor(0, 0, 0, 1);
glutDisplayFunc(myDisplay);
myInit();
glutMainLoop();
return 1;
}

```

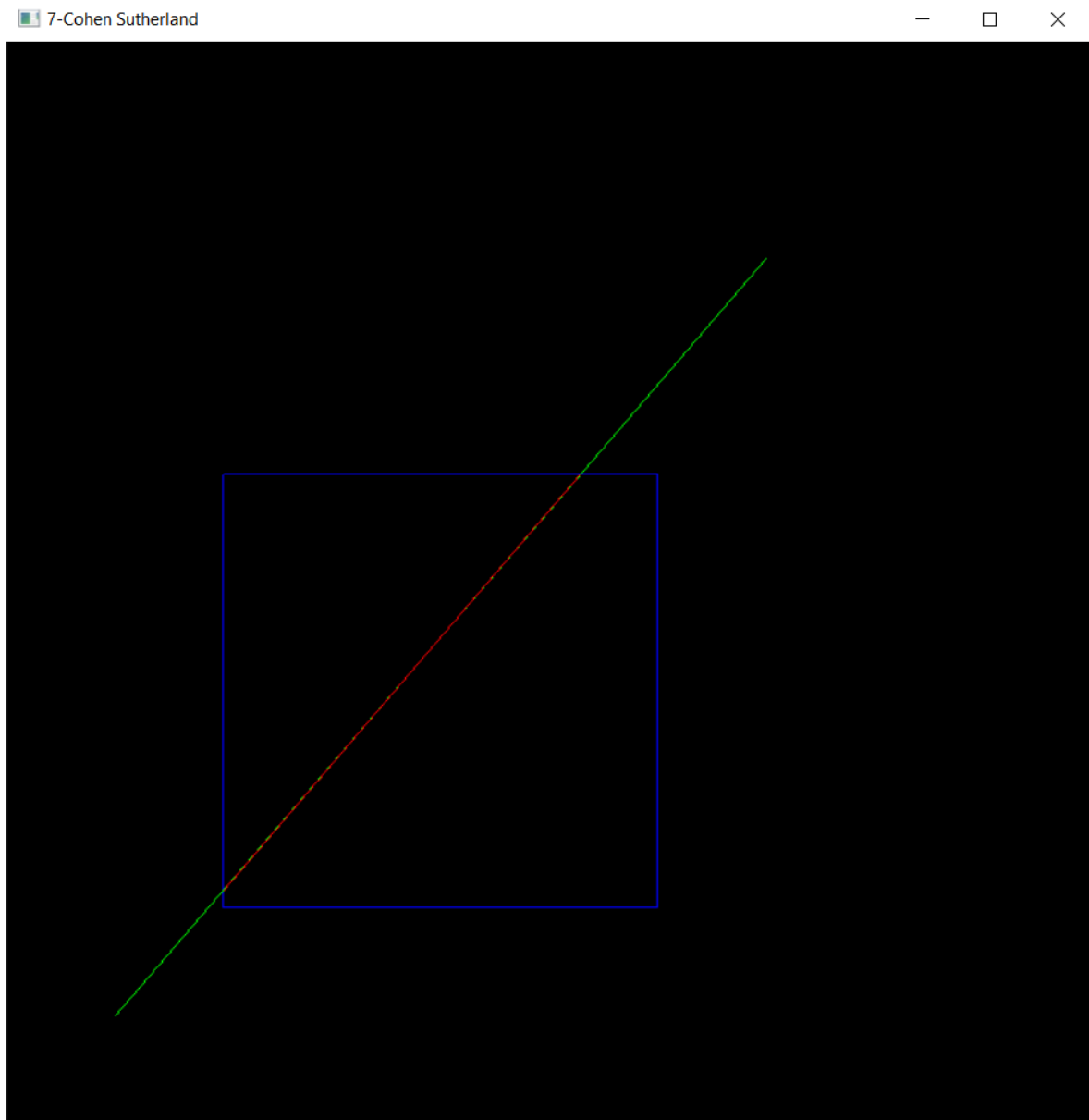
## OUTPUT:

```

C:\Users\acer\Desktop\GM-Lab\ex7>main
Enter window properties:
XWmin YWmin: 200 200
XWmax YWmax: 600 600

Enter point p1(x,y) :100 100
Enter point p2(x,y) :700 800

```



**RESULT:**

Thus line clipping is performed using Cohen Sutherland Line Clipping Algorithm.