

## CG PRACTICAL 2

Name : Jadhav Prajakta Ganesh  
class: S.E. Div : A  
Roll No.: 43

```
#include <iostream>
#include <GL/glut.h>
using namespace std;
// Define region codes for the 9 regions (left to right, top to bottom)
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
// Define the coordinates of the clipping window
const int X_MIN = 50;
const int X_MAX = 250;
const int Y_MIN = 50;
const int Y_MAX = 250;
int a1, b1, a2, b2;
// Function to calculate the region code of a point
int calculateRegionCode(int x, int 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;
}
// Function to clip a line segment using Cohen-Sutherland algorithm and draw it
void cohenSutherlandAndDraw() {
    int code1 = calculateRegionCode(a1, b1);
    int code2 = calculateRegionCode(a2, b2);
    bool accept = false;
    while (true) {
        if (!(code1 | code2)) { // Both endpoints inside the window
            accept = true;
            break;
        } else if (code1 & code2) { // Both endpoints outside the same edge

            break;
        } else {
            int codeOut = code1 ? code1 : code2;
            int x, y;
            if (codeOut & TOP) { // Clip against top edge

$$x = a1 + (a2 - a1) * (Y\_MAX - b1) / (b2 - b1);$$


```

```

y = Y_MAX;
} else if (codeOut & BOTTOM) { // Clip against bottom edge
x = a1 + (a2 - a1) * (Y_MIN - b1) / (b2 - b1);
y = Y_MIN;
} else if (codeOut & RIGHT) { // Clip against right edge
y = b1 + (b2 - b1) * (X_MAX - a1) / (a2 - a1);
x = X_MAX;
} else if (codeOut & LEFT) { // Clip against left edge
y = b1 + (b2 - b1) * (X_MIN - a1) / (a2 - a1);
x = X_MIN;
}
if (codeOut == code1) {
a1 = x;
b1 = y;
code1 = calculateRegionCode(a1, b1);
} else {
a2 = x;
b2 = y;
code2 = calculateRegionCode(a2, b2);
}
}
}
if (accept) {
glBegin(GL_LINES);
glVertex2i(a1, b1);
glVertex2i(a2, b2);
glEnd();
}
}
// Display callback function
void display() {
glClear(GL_COLOR_BUFFER_BIT);
// Draw the clipping window
glColor3f(0.0, 0.0, 1.0); // Blue color
glBegin(GL_LINE_LOOP);
glVertex2i(X_MIN, Y_MIN);
glVertex2i(X_MAX, Y_MIN);

glVertex2i(X_MAX, Y_MAX);
glVertex2i(X_MIN, Y_MAX);
glEnd();
// Draw the line segment before clipping
glColor3f(1.0, 0.0, 0.0); // Red color
glBegin(GL_LINES);
glVertex2i(a1, b1);
glVertex2i(a2, b2);
glEnd();
// Clip and draw the line segment
glColor3f(0.0, 1.0, 0.0); // Green color
cohenSutherlandAndDraw();
glFlush();
}

```

```
int main(int argc, char** argv) {
// Initialize GLUT
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(400, 400);
glutCreateWindow("aditi");

// Initialize OpenGL settings
glClearColor(1.0, 1.0, 1.0, 1.0); // White background
gluOrtho2D(0, 400, 0, 400);
// Input line coordinates
cout << "Enter the coordinates of the starting point (a1 b1): ";
cin >> a1 >> b1;
cout << "Enter the coordinates of the ending point (a2 b2): ";
cin >> a2 >> b2;
// Display callback function
glutDisplayFunc(display);
// Enter the GLUT event loop
glutMainLoop();
return 0;
```

OUTPUT:

Enter the coordinates of the starting point (a1 b1): 60 60

Enter the coordinates of the ending point (a2 b2): 800 800



