5. Clip a line using Cohen-Sutherland algorithm.

```
#include <stdio.h>
#include <GL/glut.h>
#define outcode int
double xmin = 50, ymin = 50, xmax = 100, ymax = 100;
double xvmin = 200, yvmin = 200, xvmax = 300, yvmax = 300;
const int RIGHT = 8;
const int LEFT = 2;
const int TOP = 4;
const int BOTTOM = 1;
outcode ComputeOutCode(double x, double y);

void CohenSutherlandLineClipAndDraw(double x0, double y0, double x1, double y1)
{
	//Outcodes for P0, P1, and whatever point lies outside the clip rectangle
	outcode outcode0, outcode1, outcodeOut;
	bool accept = false, done = false;
	//compute outcodes
	outcode0 = ComputeOutCode(x0, y0);
	outcode1 = ComputeOutCode(x1, y1);
	do {
		if (!(outcode0 | outcode1))	//logical or is 0 Trivially accept & exit
		{
			accept = true;
			done = true;
		}
		else if (outcode0 & outcode1)
			done = true;
		else
		{
			double x, y;
			outcodeOut = outcode0 ? outcode0 : outcode1;
			if (outcodeOut & TOP)
			{
				x = x0 + (x1 - x0) * (ymax - y0) / (y1 - y0);
				y = ymax;
			}
			else if (outcodeOut & BOTTOM)
			{
				x = x0 + (x1 - x0) * (ymin - y0) / (y1 - y0);
				y = ymin;
			}
			else if (outcodeOut & RIGHT)
			{
				y = y0 + (y1 - y0) * (xmax - x0) / (x1 - x0);
				x = xmax;
			}
			else
			{
				y = y0 + (y1 - y0) * (xmin - x0) / (x1 - x0);
				x = xmin;
			}
			if (outcodeOut == outcode0)
			{
				x0 = x;
				y0 = y;
				outcode0 = ComputeOutCode(x0, y0);
```

```
                        }
                        else
                        {
                                x1 = x;
                                y1 = y;
                                outcode1 = ComputeOutCode(x1, y1);
                        }
                }
        } while (!done);

        if (accept)
        {
                double sx = (xvmax - xvmin) / (xmax - xmin);
                double sy = (yvmax - yvmin) / (ymax - ymin);
                double vx0 = xvmin + (x0 - xmin)*sx;
                double vy0 = yvmin + (y0 - ymin)*sy;
                double vx1 = xvmin + (x1 - xmin)*sx;
                double vy1 = yvmin + (y1 - ymin)*sy;
                glColor3f(1.0, 0.0, 0.0);
                glBegin(GL_LINE_LOOP);
                glVertex2f(xvmin, yvmin);
                glVertex2f(xvmax, yvmin);
                glVertex2f(xvmax, yvmax);
                glVertex2f(xvmin, yvmax);
                glEnd();


                glColor3f(0.0, 0.0, 1.0); // draw blue colored clipped line
                glBegin(GL_LINES);
                glVertex2d(vx0, vy0);
                glVertex2d(vx1, vy1);
                glEnd();
        }
}
outcode ComputeOutCode(double x, double y)
{
        outcode code = 0;
        if (y > ymax)                   //above the clip window
                code |= TOP;
        else if (y < ymin)              //below the clip window
                code |= BOTTOM;
        if (x > xmax)                   //to the right of clip window
                code |= RIGHT;
        else if (x < xmin)              //to the left of clip window
                code |= LEFT;
        return code;
}

void display()
{
        double x0 = 30, y0 = 20, x1 = 100, y1 = 150;
        glClear(GL_COLOR_BUFFER_BIT);
        glColor3f(1.0, 0.0, 0.0);
        glBegin(GL_LINES);
        glVertex2d(x0, y0);
        glVertex2d(x1, y1);
        glVertex2d(60, 20);
        glVertex2d(80, 120);
```

```
        glEnd();
        glColor3f(0.0, 0.0, 1.0);
        glBegin(GL_LINE_LOOP);
        glVertex2f(xmin, ymin);
        glVertex2f(xmax, ymin);
        glVertex2f(xmax, ymax);
        glVertex2f(xmin, ymax);
        glEnd();
        CohenSutherlandLineClipAndDraw(x0, y0, x1, y1);
        CohenSutherlandLineClipAndDraw(60, 20, 80, 120);
        glFlush();
}


void myinit()
{
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(1.0, 0.0, 0.0);
        glPointSize(1.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 499.0, 0.0, 499.0);
}
void main(int argc, char** argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(500, 500);
        glutInitWindowPosition(0, 0);
        glutCreateWindow("Cohen Sutherland Line Clipping Algorithm");
        glutDisplayFunc(display);
        myinit();
        glutMainLoop();
}
```

Output: