```
#include <stdio.h>
#include <GL/glut.h>

double xmin, ymin, xmax, ymax; //50 50 100 100
double xvmin, yvmin, xvmax, yvmax; //200 200 300 300

int n;

struct line_segment {
        int x1;
        int y1;
        int x2;
        int y2;
};
struct line_segment ls[10];
int cliptest(double p, double q, double* u1, double* u2)
{
        double r;
        if (p) r = q / p;  // to check whether p
        if (p < 0.0)    // potentially entry point, update te
        {
                if (r > * u1)* u1 = r;
                if (r > * u2) return(false); // line portion is
outside
        }
        else
                if (p > 0.0)    //  Potentially leaving point, update
tl
                {
                        if (r < *u2)* u2 = r;
                        if (r < *u1) return(false); // line portion is
outside
                }
                else
                        if (p == 0.0)
                        {
                                if (q < 0.0) return(false); // line
parallel to edge but outside
                        }
        return(true);
}

void LiangBarskyLineClipAndDraw(double x0, double y0, double x1,
double y1)
{
        double dx = x1 - x0, dy = y1 - y0, u1 = 0.0, u2 = 1.0;
        //draw a red colored viewport
        glColor3f(1.0, 0.0, 0.0);
        glBegin(GL_LINE_LOOP);
        glVertex2f(xvmin, yvmin);
        glVertex2f(xvmax, yvmin);
        glVertex2f(xvmax, yvmax);
        glVertex2f(xvmin, yvmax);
        glEnd();
        if (cliptest(-dx, x0 - xmin, &u1, &u2))  // inside test wrt
left edge
```

```
                if (cliptest(dx, xmax - x0, &u1, &u2)) // inside test
wrt right edge
                        if (cliptest(-dy, y0 - ymin, &u1, &u2)) //
inside test wrt bottom edge
                                if (cliptest(dy, ymax - y0, &u1, &u2))
// inside test wrt top edge
                                {
                                        if (u2 < 1.0)
                                        {
                                                x1 = x0 + u2 * dx;
                                                y1 = y0 + u2 * dy;
                                        }
                                        if (u1 > 0.0)
                                        {
                                                x0 = x0 + u1 * dx;
                                                y0 = y0 + u1 * dy;
                                        }
                                        // Window to viewport mappings
                                        double sx = (xvmax - xvmin) /
(xmax - xmin); // Scale parameters
                                        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(0.0, 0.0, 1.0); //
draw blue colored clipped line
                                        glBegin(GL_LINES);
                                        glVertex2d(vx0, vy0);
                                        glVertex2d(vx1, vy1);
                                        glEnd();
                                }
}// end of line clipping

void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        //draw the line with red color
        glColor3f(1.0, 0.0, 0.0);
        for (int i = 0; i < n; i++)
        {
                glBegin(GL_LINES);
                glVertex2d(ls[i].x1, ls[i].y1);
                glVertex2d(ls[i].x2, ls[i].y2);
                glEnd();
        }
        //draw a blue colored window
        glColor3f(0.0, 0.0, 1.0);
        glBegin(GL_LINE_LOOP);
        glVertex2f(xmin, ymin);
        glVertex2f(xmax, ymin);
```

```
        glVertex2f(xmax, ymax);
        glVertex2f(xmin, ymax);
        glEnd();
        for (int i = 0; i < n; i++)
                LiangBarskyLineClipAndDraw(ls[i].x1, ls[i].y1,
ls[i].x2, ls[i].y2);
        glFlush();
}

void myinit()
{
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glColor3f(1.0, 0.0, 0.0);
        glLineWidth(2.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        gluOrtho2D(0.0, 499.0, 0.0, 499.0);
}

int main(int argc, char** argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(500, 500);
        glutInitWindowPosition(0, 0);

        printf("Enter window coordinates: (xmin ymin xmax ymax) \n");
        scanf("%lf%lf%lf%lf", &xmin, &ymin, &xmax, &ymax);
        printf("Enter viewport coordinates: (xvmin yvmin xvmax yvmax)
\n");
        scanf("%lf%lf%lf%lf", &xvmin, &yvmin, &xvmax, &yvmax);
        printf("Enter no. of lines:\n");
        scanf("%d", &n);

        for (int i = 0; i < n; i++)
        {
                printf("Enter coordinates: (x1 y1 x2 y2)\n");
                scanf("%d%d%d%d", &ls[i].x1, &ls[i].y1, &ls[i].x2,
&ls[i].y2);
        }
        glutCreateWindow("Liang Barsky Line Clipping Algorithm");
        glutDisplayFunc(display);
        myinit();
        glutMainLoop();
}
```