**Program 7:**

#include <stdio.h>

#include <GL/glut.h>

#include "Header.h"

double xmin\_9 = 50, ymin\_9 = 50, xmax\_9 = 100, ymax\_9 = 100; // Window boundaries

double xvmin\_9 = 200, yvmin\_9 = 200, xvmax\_9 = 300, yvmax\_9 = 300; // Viewport boundaries

int cliptest(double p, double q, double\* t1, double\* t2)

{

double t = q / p;

if (p < 0.0) // potentially enry point, update te

{

if (t > \* t1) \*t1 = t;

if (t > \* t2) return(false); // line portion is outside

}

else

if (p > 0.0) // Potentially leaving point, update tl

{

if (t < \*t2) \*t2 = t;

if (t < \*t1) 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, te = 0.0, tl = 1.0;

if (cliptest(-dx, x0 - xmin\_9, &te, &tl)) // inside test wrt left edge

if (cliptest(dx, xmax\_9 - x0, &te, &tl)) // inside test wrt right edge

if (cliptest(-dy, y0 - ymin\_9, &te, &tl)) // inside test wrt bottom edge

if (cliptest(dy, ymax\_9 - y0, &te, &tl)) // inside test wrt top edge

{

if (tl < 1.0)

{

x1 = x0 + tl \* dx;

y1 = y0 + tl \* dy;

}

if (te > 0.0)

{

x0 = x0 + te \* dx;

y0 = y0 + te \* dy;

}

// Window to viewport mappings

double sx = (xvmax\_9 - xvmin\_9) / (xmax\_9 - xmin\_9); // Scale parameters

double sy = (yvmax\_9 - yvmin\_9) / (ymax\_9 - ymin\_9);

double vx0 = xvmin\_9 + (x0 - xmin\_9) \* sx;

double vy0 = yvmin\_9 + (y0 - ymin\_9) \* sy;

double vx1 = xvmin\_9 + (x1 - xmin\_9) \* sx;

double vy1 = yvmin\_9 + (y1 - ymin\_9) \* sy;

//draw a red colored viewport

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_LINE\_LOOP);

glVertex2f(xvmin\_9, yvmin\_9);

glVertex2f(xvmax\_9, yvmin\_9);

glVertex2f(xvmax\_9, yvmax\_9);

glVertex2f(xvmin\_9, yvmax\_9);

glEnd();

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\_LabPA9()

{

double x0 = 60, y0 = 20, x1 = 80, y1 = 120;

glClear(GL\_COLOR\_BUFFER\_BIT);

//draw the line with red color

glColor3f(1.0, 0.0, 0.0);

//bres(120,20,340,250);

glBegin(GL\_LINES);

glVertex2d(x0, y0);

glVertex2d(x1, y1);

glEnd();

//draw a blue colored window

glColor3f(0.0, 0.0, 1.0);

glBegin(GL\_LINE\_LOOP);

glVertex2f(xmin\_9, ymin\_9);

glVertex2f(xmax\_9, ymin\_9);

glVertex2f(xmax\_9, ymax\_9);

glVertex2f(xmin\_9, ymax\_9);

glEnd();

LiangBarskyLineClipAndDraw(x0, y0, x1, y1);

glFlush();

}

void myinit\_LabPA9()

{

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 LabPA\_9\_main(int argc, char\*\* argv)

{

//int x1, x2, y1, y2;

//printf("Enter End points:");

//scanf\_s("%d%d%d%d", &x1,&x2,&y1,&y2);

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(0, 0);

glutCreateWindow("Liang Barsky Line Clipping Algorithm");

glutDisplayFunc(display\_LabPA9);

myinit\_LabPA9();

glutMainLoop();

}

**OUTPUT**

