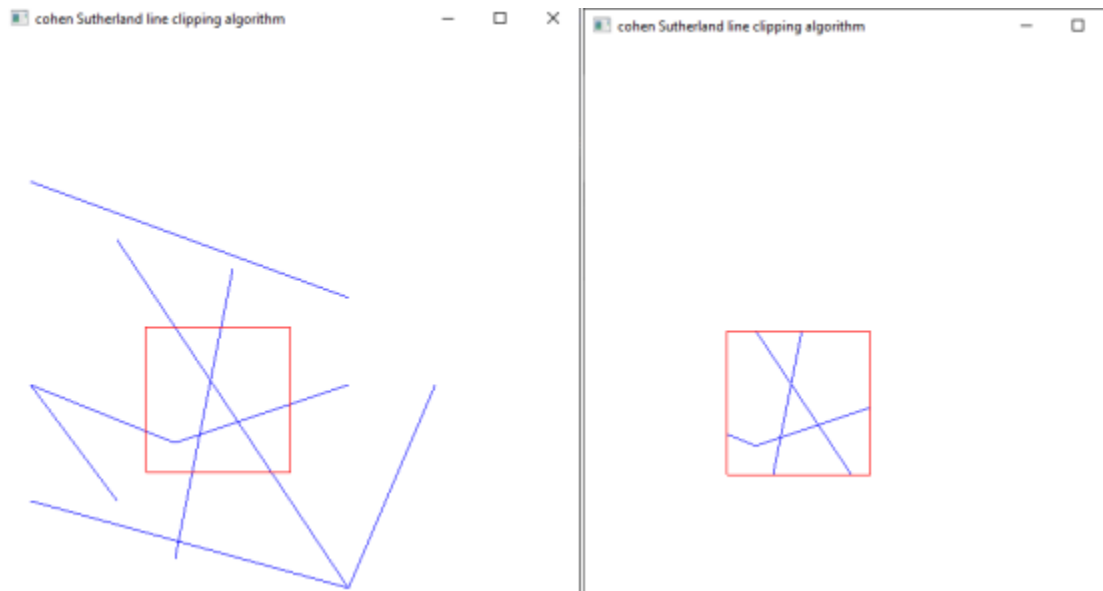


Practical-7

AIM: To write a program to implement line clipping algorithm by using Cohen-Sutherland Line clipping algorithm.



Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <iostream>
#include <conio.h>
#include <dos.h>
#define ESCAPE 27
#define outcode int
#include <GL/gl.h>
#include <stdlib.h>

#include <windows.h>
#include <stdlib.h>
namespace gp7 {
    double xmin = 50, ymin = 50, xmax = 100, ymax = 100; // Window boundaries
    const int TOP = 1;
    const int BOTTOM = 2;
    const int RIGHT = 4;
    const int LEFT = 8;
    outcode ComputeOutCode(double x, double y);
    void CohenSutherlandLineClipAndDraw(double x0, double y0, double x1, double
y1)
```

```

{
    outcode outcode0, outcode1, outcodeOut;
    bool accept = false, done = false;

    outcode0 = ComputeOutCode(x0, y0);
    outcode1 = ComputeOutCode(x1, y1);
    do
    {
        if (!(outcode0 | outcode1))
        {
            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)
    {
        //draw a red colored viewport
    }
}

```

```

        glColor3f(1.0, 0.0, 0.0);
        glBegin(GL_LINE_LOOP);
        glVertex2f(xmin + 200, ymin + 200);
        glVertex2f(xmax + 200, ymin + 200);
        glVertex2f(xmax + 200, ymax + 200);
        glVertex2f(xmin + 200, ymax + 200);
        glEnd();
        glColor3f(0.0, 0.0, 1.0); // draw blue colored clipped line
        glBegin(GL_LINES);
        glVertex2d(x0 + 200, y0 + 200);
        glVertex2d(x1 + 200, y1 + 200);
        glEnd();
    }
}

outcode ComputeOutCode(double x, double y)
{
    outcode code = 0;
    if (y > ymax)
        code |= TOP;
    else if (y < ymin)
        code |= BOTTOM;
    if (x > xmax)
        code |= RIGHT;
    else if (x < xmin)
        code |= LEFT;
    return code;
}

void display()
{
    double x0 = 120, y0 = 10, x1 = 40, y1 = 130;
    glClear(GL_COLOR_BUFFER_BIT);
    //draw the line with red color
    glColor3f(1.0, 0.0, 0.0);
    glBegin(GL_LINES);
    glVertex2d(x0, y0);
    glVertex2d(x1, y1);
    glVertex2d(10, 70);
    glVertex2d(250, 90);
    glEnd();
    //draw a blue colored window
    glColor3f(0.0, 0.0, 1.0);
    glBegin(GL_LINE_LOOP);
    glVertex2d(xmin, ymin);
    glVertex2d(xmax, ymin);
    glVertex2d(xmax, ymax);
    glVertex2d(xmin, ymax);
    glEnd();
    CohenSutherlandLineClipAndDraw(x0, y0, x1, y1);

    CohenSutherlandLineClipAndDraw(10, 70, 250, 90);
    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("21172012015_Malay Patel");
        glutDisplayFunc(display);
        myInit();
        glutMainLoop();
    }
}
```

OUTPUT:

