

9. Develop a menu driven program to fill the polygon using scan line algorithm

```
#include <stdlib.h>
#include <stdio.h>
#include <GL/glut.h>
float x1, x2, x3, x4, y1, y2, y3, y4;
int fillFlag = 0;

void edgedetect(float x1, float y1, float x2, float y2, int* le, int* re)
{
    float mx, x, temp;
    int i;
    if ((y2 - y1) < 0) // we swap so we don't get a negative slope
    {
        temp = y1;
        y1 = y2;
        y2 = temp;
        temp = x1;
        x1 = x2;
        x2 = temp;
    }
    if ((y2 - y1) != 0) // to find the slope if not zero
        mx = (x2 - x1) / (y2 - y1);

    else
        mx = x2 - x1;
    // slope is mx
    x = x1;

    for (i = y1; i <= y2; i++) // follow along the edge of the polygon
    {
        if (x < le[i])
            le[i] = x;
        if (x > re[i])
            re[i] = x;
        x += mx;
    }
}

void draw_pixel(int x, int y)
{
    glColor3f(1.0, 1.0, 0.0);
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}

void scanfill(float x1, float y1, float x2, float y2, float x3, float y3, float x4,
float y4)
{
    int le[500], re[500]; // left edge and right edge
    int i, y;
    for (i = 0; i < 500; i++)
    {
        le[i] = 500;
        re[i] = 0;
    }

    edgedetect(x1, y1, x2, y2, le, re);
    edgedetect(x2, y2, x3, y3, le, re);
    edgedetect(x3, y3, x4, y4, le, re);
    edgedetect(x4, y4, x1, y1, le, re);
}
```

```

        for (y = 0; y < 500; y++)
        {
            for (i = le[y]; i < re[y]; i++)
                draw_pixel(i, y);
        }
    }

void display()
{
    x1 = 200.0; y1 = 200.0; x2 = 100.0; y2 = 300.0; x3 = 200.0; y3 = 400.0; x4 =
300.0; y4
    = 300.0;
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 0.0, 1.0);
    glBegin(GL_LINE_LOOP);
    glVertex2f(x1, y1);
    glVertex2f(x2, y2);
    glVertex2f(x3, y3);
    glVertex2f(x4, y4);
    glEnd();

    if (fillFlag == 1)
        scanfill(x1, y1, x2, y2, x3, y3, x4, y4);
    glFlush();
}

void init()
{
    glClearColor(0.0, 0.0, 0.0, 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 fillMenu(int option)
{
    if (option == 1)
        fillFlag = 1;
    if (option == 2)
        fillFlag = 2;
    display();
}

void main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Filling a Polygon using Scan-line Algorithm");
    init();
    glutDisplayFunc(display);
    glutCreateMenu(fillMenu);
    glutAddMenuEntry("Fill Polygon", 1);
    glutAddMenuEntry("Empty Polygon", 2);
    glutAttachMenu(GLUT_RIGHT_BUTTON);
    glutMainLoop();
}

```