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

float x1, x2, x3, x4, y1, y2, y3, y4;    // our polygon has 4 lines - so 8 coordinates

void edgedetect(float x1, float y1, float x2, float y2, int *left_edge, int *right_edge)
{
    float x_slope, x, temp;
    int i;

    if ((y2-y1)<0)    // decide where to start
    {
        temp = y1;
        y1 = y2;
        y2 = temp;

        temp = x1;
        x1 = x2;
        x2 = temp;
    }

    if ((y2-y1)!=0)
        x_slope = (x2 - x1) / (y2 - y1);
    else
        x_slope = x2 - x1;

    x = x1;

    for (i = y1; i <= y2; i++)
    {
        if (x < left_edge[i])
            left_edge[i] = x;

        if (x > right_edge[i])
            right_edge[i] = x;

        x = x + x_slope;
    }
}

void draw_pixel (int x, int y)
{
    glColor3f (1, 1, 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 left_edge[500], right_edge[500];
    int i, y;

    for (i = 0; i <= 500; i++)
    {
        left_edge [i] = 500;
        right_edge [i] = 0;
    }

    edgedetect (x1, y1, x2, y2, left_edge, right_edge);

    edgedetect (x2, y2, x3, y3, left_edge, right_edge);

    edgedetect (x3, y3, x4, y4, left_edge, right_edge);

    edgedetect (x4, y4, x1, y1, left_edge, right_edge);

    for (y = 0; y <= 500; y++)
    {
        if (left_edge[y] <= right_edge[y])
        {
            for (i = left_edge[y]; i <= right_edge[y]; i++)
            {
                draw_pixel (i, y);
                glFlush ();
            }
        }
    }
}

void display()
{
    x1 = 200, y1 = 200;
    x2 = 100, y2 = 300;
    x3 = 200, y3 = 400;
    x4 = 300, y4 = 300;

    glClear (GL_COLOR_BUFFER_BIT);

    glColor3f (0, 0, 1);

    glBegin (GL_LINE_LOOP);
    glVertex2f (x1, y1);
    glVertex2f (x2, y2);
    glVertex2f (x3, y3);
    glVertex2f (x4, y4);
    glEnd ();

    scanfill (x1, y1, x2, y2, x3, y3, x4, y4);
}

void init()
{
    glClearColor (1, 1, 1, 1);
    gluOrtho2D (0, 499, 0, 499);
}

int 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);

    glutMainLoop ();
}
```