**Program4:**

#include<stdlib.h>

#include<gl/glut.h>

#include<algorithm>

#include<iostream>

#include<windows.h>

using namespace std;

float x[100], y[100]; //= { 0,0,20,100,100 }, y[] = { 0,100,50,100,0 };

int n, m;

int wx = 500, wy = 500;

static float intx[10] = { 0 };

void draw\_line(float x1, float y1, float x2, float y2) {

Sleep(100);

glColor3f(1, 0, 0);

glBegin(GL\_LINES);

glVertex2f(x1, y1);

glVertex2f(x2, y2);

glEnd();

glFlush();

}

void edgeDetect(float x1, float y1, float x2, float y2, int scanline) {

float temp;

if (y2 < y1) {

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

temp = y1; y1 = y2; y2 = temp;

}

if (scanline > y1 && scanline < y2)

intx[m++] = x1 + (scanline - y1) \* (x2 - x1) / (y2 - y1);

}

void scanfill(float x[], float y[]) {

for (int s1 = 0; s1 <= wy; s1++) {

m = 0;

for (int i = 0; i < n; i++) {

edgeDetect(x[i], y[i], x[(i + 1) % n], y[(i + 1) % n], s1);

}

sort(intx, (intx + m));

if (m >= 2)

for (int i = 0; i < m; i = i + 2)

draw\_line(intx[i], s1, intx[i + 1], s1);

}

}

void display\_filled\_polygon() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glLineWidth(2);

glBegin(GL\_LINE\_LOOP);

for (int i = 0; i < n; i++)

glVertex2f(x[i], y[i]);

glEnd();

scanfill(x, y);

//glFlush();

}

void myInit() {

glClearColor(1, 1, 1, 1);

glColor3f(0, 0, 1);

glPointSize(1);

gluOrtho2D(0, wx, 0, wy);

}

void main(int ac, char\* av[]) {

glutInit(&ac, av);

printf("Enter no. of sides: \n");

scanf\_s("%d", &n);

printf("Enter coordinates of endpoints: \n");

for (int i = 0; i < n; i++)

{

printf("X-coord Y-coord: \n");

scanf("%f %f", &x[i], &y[i]);

}

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(0, 0);

glutCreateWindow("scanline");

glutDisplayFunc(display\_filled\_polygon);

myInit();

glutMainLoop();

}

**Output:**







