Assignment No.7

# Name: Omkar Manohar Hepat

# Class: CS-B

# Roll No: 31 PRN No: 12211509

# Subject: CGAVR

## Q1. Implementation of Koch Snowflake

#include <GL/glut.h>

#include <cmath>

void drawKochSnowflake(float x1, float y1, float x2, float y2, int depth) {

if (depth == 0) {

glVertex2f(x1, y1);

glVertex2f(x2, y2);

}

else {

float deltaX = x2 - x1;

float deltaY = y2 - y1;

float length = sqrt(deltaX \* deltaX + deltaY \* deltaY);

float unitX = deltaX / length;

float unitY = deltaY / length;

float x3 = x1 + deltaX / 3;

float y3 = y1 + deltaY / 3;

float x4 = (x1 + x2) / 2 + sqrt(3) \* (y1 - y2) / 6;

float y4 = (y1 + y2) / 2 + sqrt(3) \* (x2 - x1) / 6;

float x5 = x1 + deltaX \* 2 / 3;

float y5 = y1 + deltaY \* 2 / 3;

drawKochSnowflake(x1, y1, x3, y3, depth - 1);

drawKochSnowflake(x3, y3, x4, y4, depth - 1);

drawKochSnowflake(x4, y4, x5, y5, depth - 1);

drawKochSnowflake(x5, y5, x2, y2, depth - 1);

}

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_LINES);

glColor3f(1.0, 1.0, 1.0);

drawKochSnowflake(-0.5f, -0.433f, 0.5f, -0.433f, 4);

drawKochSnowflake(0.5f, -0.433f, 0.0f, 0.433f, 4);

drawKochSnowflake(0.0f, 0.433f, -0.5f, -0.433f, 4);

glEnd();

glFlush();

}

void init() {

glClearColor(0.0, 0.0, 0.0, 1.0);

gluOrtho2D(-1.0, 1.0, -1.0, 1.0);

}

int main(int argc, char\*\* argv) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(800, 800);

glutCreateWindow("Koch Snowflake Fractal");

glutDisplayFunc(display);

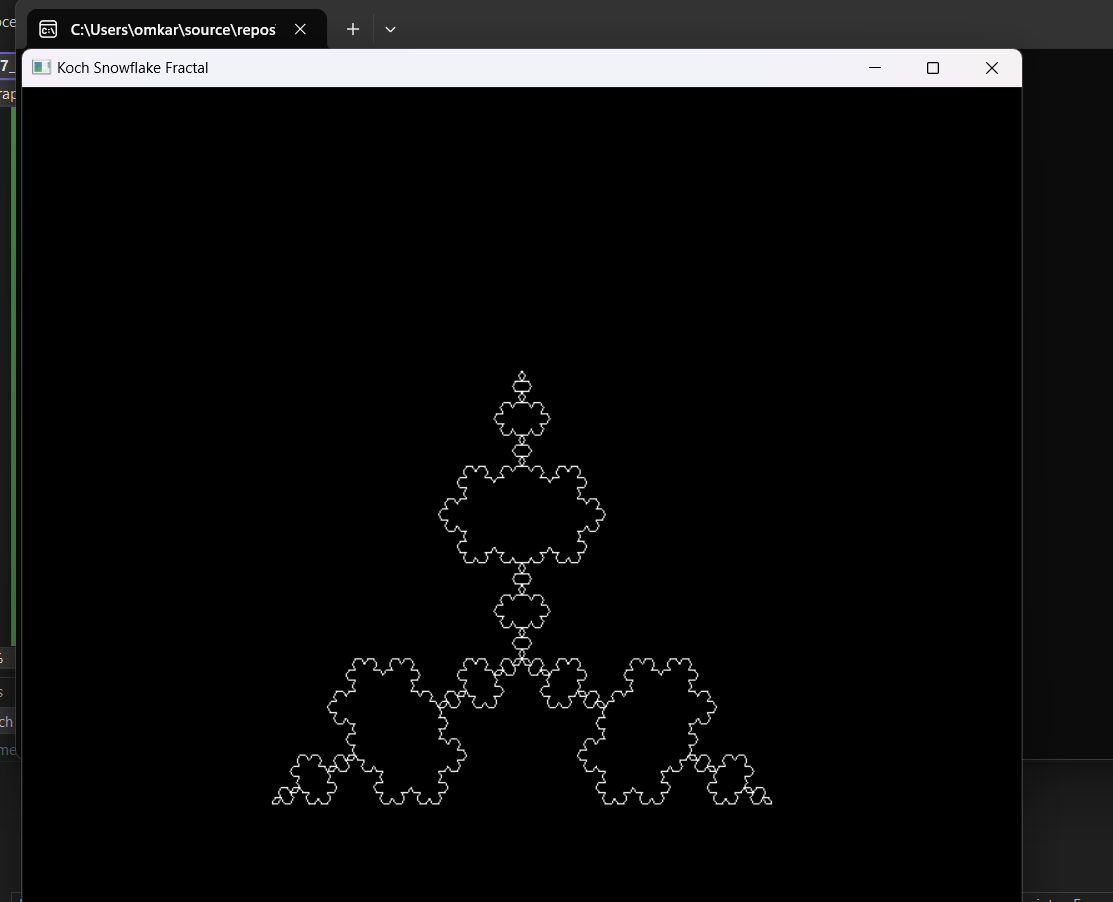
init();

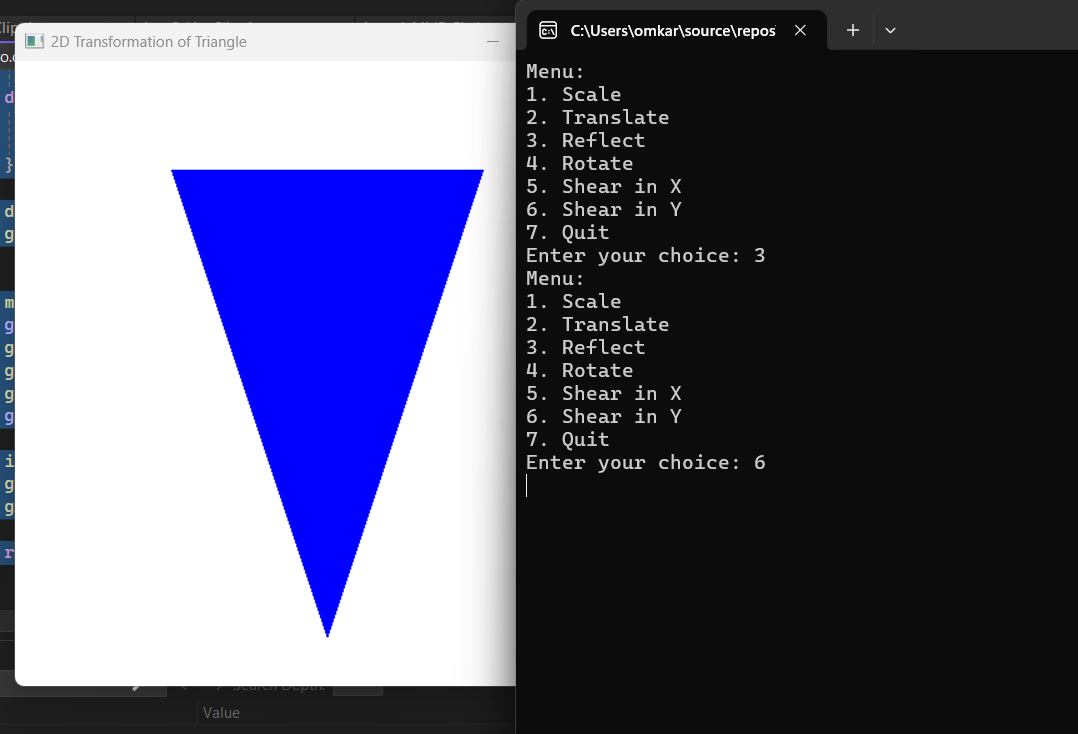
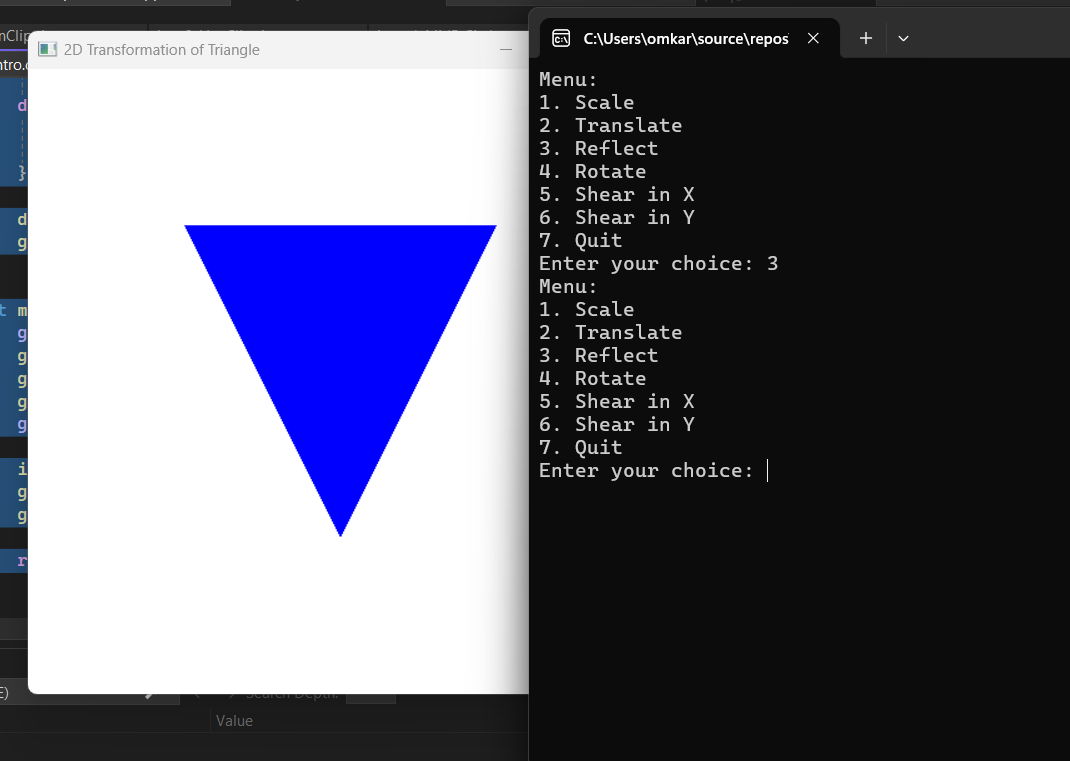
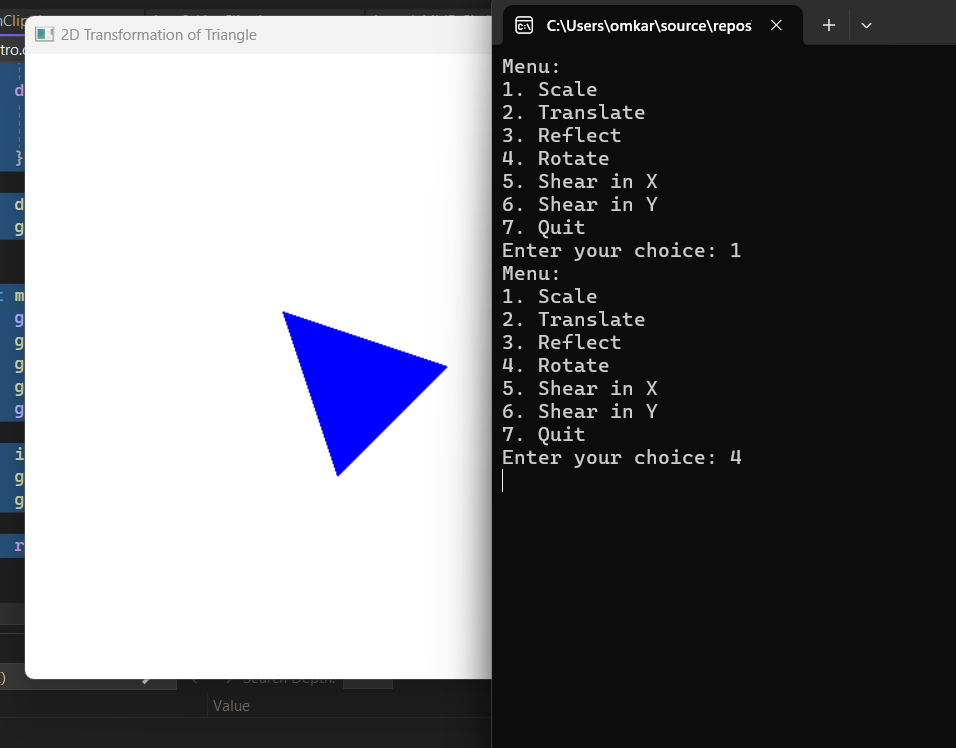
glutMainLoop();

return 0;

}

Output:



****