

Practical NO.07

Title:- Generate fractal patterns using i) Koch Curve

Name:-Ujawala Sinha

Roll no.S554

CODE:

```
#include <GL/glut.h>
#include <math.h>

// Initial points of the line
float ax = -200, ay = 0;
float bx = 200, by = 0;

// Recursive depth
int depth = 2;

void drawKochCurve(float x1, float y1, float x2, float y2, int n)
{
    if (n == 0)
    {
        glVertex2f(x1, y1);
        glVertex2f(x2, y2);
    }
    else
    {
        float x3 = (2 * x1 + x2) / 3;
        float y3 = (2 * y1 + y2) / 3;

        float x4 = (x1 + 2 * x2) / 3;
        float y4 = (y1 + 2 * y2) / 3;

        // Calculate peak of the triangle
        float dx = x4 - x3;
        float dy = y4 - y3;

        float x = x3 + (dx * cos(M_PI / 3) - dy * sin(M_PI / 3));
        float y = y3 + (dx * sin(M_PI / 3) + dy * cos(M_PI / 3));

        drawKochCurve(x1, y1, x3, y3, n - 1);
        drawKochCurve(x3, y3, x, y, n - 1);
        drawKochCurve(x, y, x4, y4, n - 1);
        drawKochCurve(x4, y4, x2, y2, n - 1);
    }
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 1.0, 1.0);
```

```

    glBegin(GL_LINES);
    drawKochCurve(ax, ay, bx, by, depth);
    glEnd();
    glFlush();
}

void init()
{
    glClearColor(0.0, 0.0, 0.0, 1.0); // black background
    gluOrtho2D(-300, 300, -200, 200); // setting coordinate system
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(600, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Koch Curve - OpenGL in C++");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

Output

svpm@svpm-HP-EliteDesk-800-G2-SFF:~\$ g++ koch.cpp -lGL -lGLU -lglut svpm@svpm-HP-EliteDesk-800-G2-SFF:~\$./a.out

