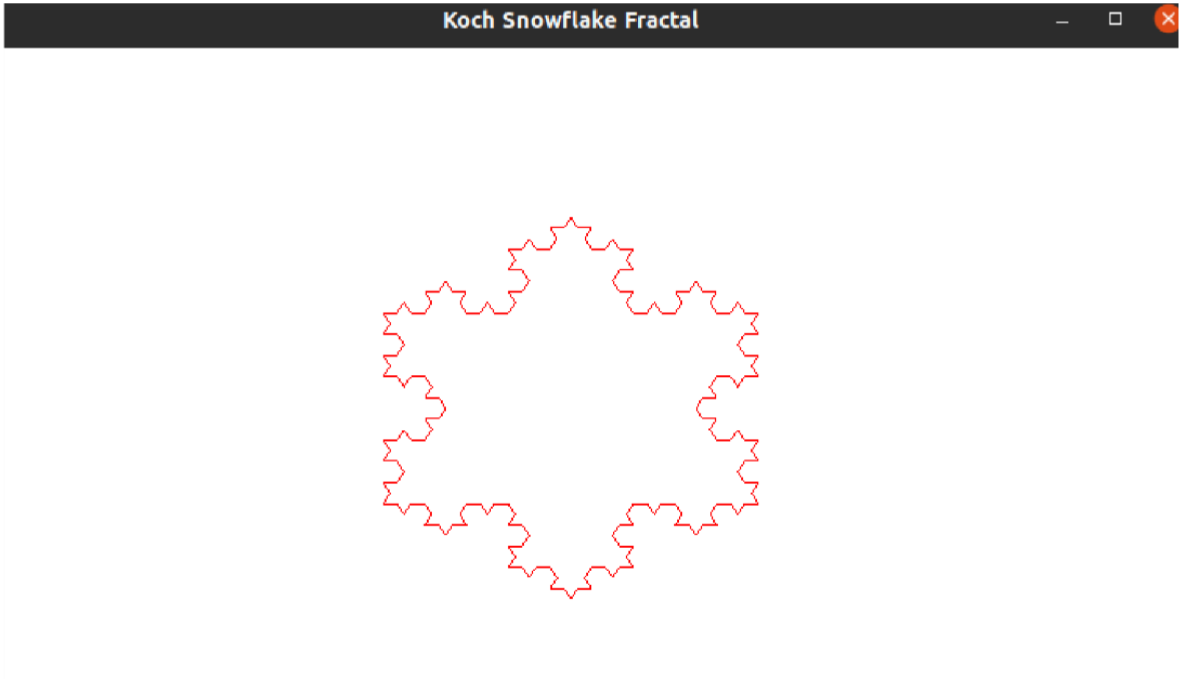
ASSIGNMENT NO 5

Code :-

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


int depth = 3; // Adjust this to change the level of recursion


void drawKochSnowflake(int depth, double x1, double y1, double x5, double y5)
{ if (depth == 0) {
// Draw a line segment from (x1, y1) to (x5,
y5) glBegin(GL_LINES); glVertex2d(x1, y1);
glVertex2d(x5, y5); glEnd();
}
else { double deltaX =
x5 - x1; double deltaY
= y5 - y1;


// Calculate new points
double x2 = x1 + deltaX / 3;
double y2 = y1 + deltaY / 3;


double x3 = 0.5 * (x1 + x5) + (sqrt(3) / 6) * (y1 - y5);
double y3 = 0.5 * (y1 + y5) + (sqrt(3) / 6) * (x5 - x1);


double x4 = x1 + 2 * deltaX / 3;
double y4 = y1 + 2 * deltaY / 3; //
Recursively draw smaller segments
drawKochSnowflake(depth - 1, x1,
y1, x2, y2);
drawKochSnowflake(depth - 1, x2,
y2, x3, y3);
drawKochSnowflake(depth - 1, x3,
y3, x4, y4);
drawKochSnowflake(depth - 1, x4,
y4, x5, y5);
}
}


void display() {
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1.0f, 0.0f, 0.0f); // Black color


double x1 = -0.3; double
y1 = 0.3 * sqrt(3);
double x2 = 0.3; double
y2 = 0.3 * sqrt(3);
double x3 = 0.0; double
y3 = 0.0;


drawKochSnowflake(depth, x1, y1, x2, y2);
drawKochSnowflake(depth, x2, y2, x3, y3);
drawKochSnowflake(depth, x3, y3, x1, y1);


glFlush();
}


int main(int argc, char** argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(800, 800);
glutCreateWindow("Koch Snowflake Fractal");


glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // White background
glutDisplayFunc(display);

glutMainLoop();
return 0;
```

```
}
```

Output :



Koch Snowflake Fractal