Assignment No. 5

Title	Curves and fractals
Aim/Problem Statement	a) Write C++ program to generate snowflake using concept of fractals. **OR** b) Write C++ program to generate Hilbert curve using concept of fractals. **OR** c) Write C++ program to generate fractal patterns by using Koch curves.
CO Mapped	
Pre-requisite	1. Basic programming skills of C++ 2. 64-bit Open source Linux 3. Open Source C++ Programming tool like G++/GCC
Learning Objective	To study curves and fractals

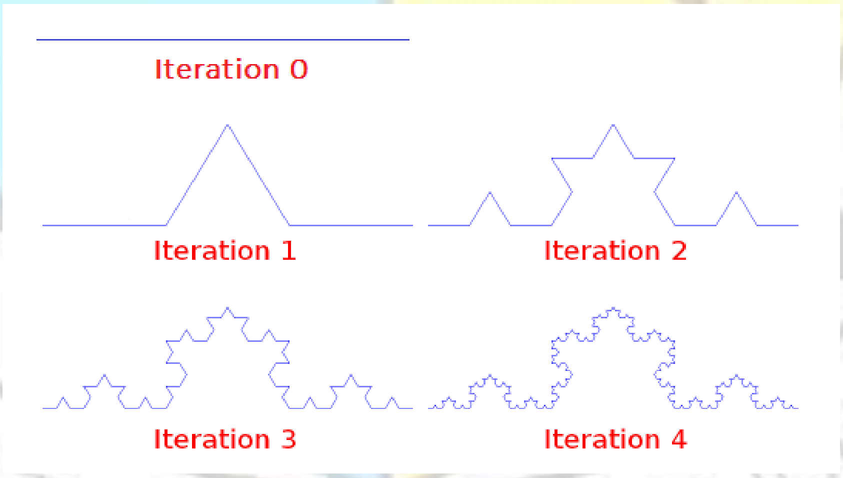
Theory:

Koch Curve:

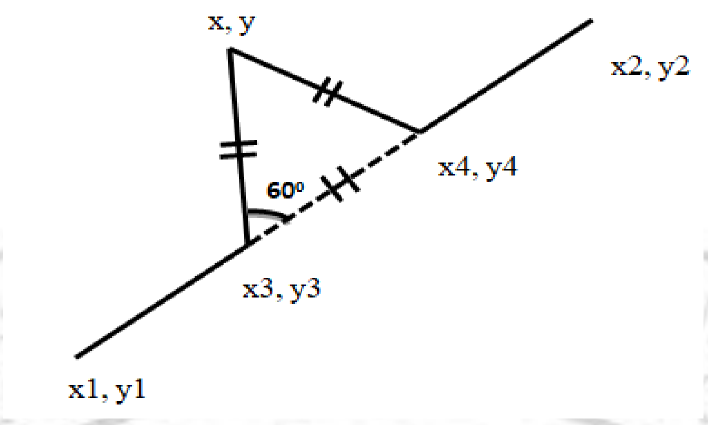
The Koch curve fractal was first introduced in 1904 by Helge von Koch. It was one of the first fractal objects to be described. To create a Koch curve

1. Create a line and divide it into three parts.
2. The second part is now rotated by 60°.
3. Add another part which goes from the end of part 2 to to the beginning of part 3
4. Repeat step 1 to step 3 with each part of the line.

We will get following Koch curve as number of iteration goes on increasing



Step 1: In Iteration 0, we have a horizontal line.
Step 2: In Iteration 1, line is divided into 3 equal parts. Middle part of a line is rotated in 60^0, because it forms a perfect an equilateral triangle as shown below:



Here, (x1,y1) and (x2, y2) is accepted from user.
Now, we can see line is divided into 3 equal segments: segment((x1,y1),(x3,y3)), segment((x3,y3),(x4,y4)),segment((x4,y4),(x2,y2)) in above figure.

Coordinates of middle two points will be calculated as follows:

```
x3 = (2*x1+x2)/3;
y3 = (2*y1+y2)/3;
x4 = (x1+2*x2)/3;
y4 = (y1+2*y2)/3;
```

In our curve, middle segment((x3,y3),(x4,y4)) will not be drawn. Now, in order to find out coordinates of the top vertex (x,y) of equilateral triangle, we have rotate point (x4,y4) with respect to arbitrary point (x3,y3) by angle of 60 degree in anticlockwise direction. After performing this rotation, we will get rotated coordinates (x, y) as:

$$x = x_3 + (x_4 - x_3) * \cos\theta + (y_4 - y_3) * \sin\theta$$
$$y = y_3 - (x_4 - x_3) * \sin\theta + (y_4 - y_3) * \cos\theta$$

Step 3: In iteration 2, you will repeat step 2 for every segment obtained in iteration1.

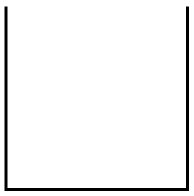
In this way, you can generate Koch curve for any number of iterations.

The Hilbert curve

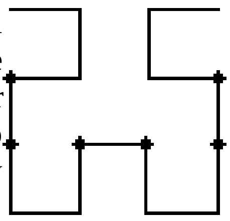
The Hilbert curve is a space filling curve that visits every point in a square grid with a size of 2×2, 4×4, 8×8, 16×16, or any other power of 2. It was first described by David Hilbert in 1892. Applications of the Hilbert curve are in image processing: especially image compression and dithering. It has advantages in those operations where the coherence between neighbouring pixels is important. The Hilbert curve is also a special version of a quadtree; any image processing function that benefits from the use of quadtrees may also use a Hilbert curve.

Cups and joins

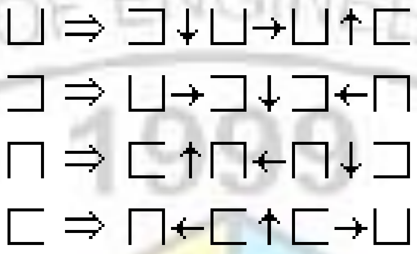
The basic elements of the Hilbert curves are what I call "cups" (a square with one open side) and "joins" (a vector that joins two cups). The "open" side of a cup can be top, bottom, left or right. In addition, every cup has two end-points, and each of these can be the "entry" point or the "exit" point. So, there are eight possible varieties of cups. In practice, a Hilbert curve uses only four types of cups. In a similar vein, a join has a direction: up, down, left or right.



A first order Hilbert curve is just a single cup (see the figure on the left). It fills a 2×2 space. The second order Hilbert curve replaces that cup by four (smaller) cups, which are linked together by three joins (see the figure on the right; the link between a cup and a join has been marked with a fat dot in the figure). Every next order repeats the process or replacing each cup by four smaller cups and three joins.



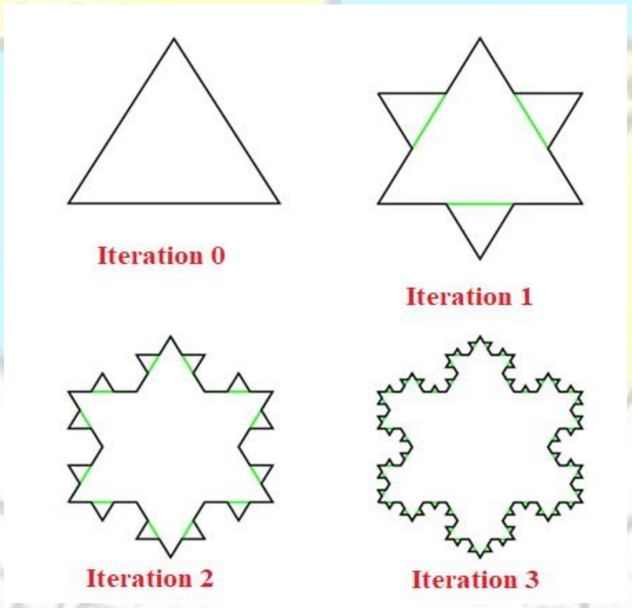
Cup subdivision rules



The function presented below (in the "C" language) computes the Hilbert curve. Note that the curve is symmetrical around the vertical axis. It would therefore be sufficient to draw half of the Hilbert curve.

Snowflake curve:

Snowflake curve is drawn using koch curve iterations. In koch curve, we just have a single line in the starting iteration and in snowflake curve, we have an equilateral triangle. Draw an equilateral triangle and repeat the steps of Koch curve generation for all three segments of an equilateral triangle.



Conclusion:

Questions:

1. What is the importance of curves and fractals in computer graphics?
2. What are applications of curves and fractals?