Experiment No: 11

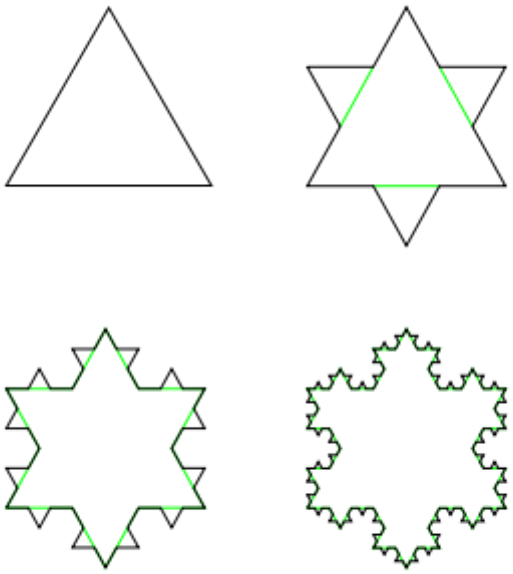
Title: Write Write C++/Java program to generate fractal patterns by using Koch curves.

Software Required: Ubuntu, GCC

Theory:

The Koch curve (also known as the Koch star, or Koch island) is a mathematical curve and one of the earliest fractal curves to have been described. It is based on the Koch curve, which appeared in a 1904 paper titled "On a continuous curve without tangents, constructible from elementary geometry".

The progression for the area of the snowflake converges to 8/5 times the area of the original triangle, while the progression for the snowflake's perimeter diverges to infinity. Consequently, the snowflake has a finite area bounded by an infinitely long line.



The Koch snowflake can be constructed by starting with an equilateral triangle, then recursively altering each line segment as follows:

1.divide the line segment into three segments of equal length.
2. draw an equilateral triangle that has the middle segment from step 1 as its base and points outward.
3 .remove the line segment that is the base of the triangle from step 2.

After one iteration of this process, the resulting shape is the outline of a hexagram. The Koch snowflake is the limit approached as the above steps are followed over and over again. The Koch curve originally described by Helge von Koch is constructed with only one of the three sides of the original triangle. In other words, three Koch curves make a Koch snowflake.

Algorithm:

```
#include<graphics.h>
#include<iostream>
#include<math.h>

using namespace std;

void koch(int x1, int y1, int x2, int y2, int it)
{
      float angle = 60/180;
      int x3 = (2*x1+x2)/3;
      int y3 = (2*y1+y2)/3;

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

      int x = x3 + (x4-x3)*cos(angle)+(y4-y3)*sin(angle);
      int y = y3 - (x4-x3)*sin(angle)+(y4-y3)*cos(angle);

      if(it > 0)
      {
            koch(x1, y1, x3, y3, it-1);
            koch(x3, y3, x, y, it-1);
            koch(x, y, x4, y4, it-1);
            koch(x4, y4, x2, y2, it-1);
      }
      else
      {
           line(x1, y1, x3, y3);
```

```
            line(x3, y3, x, y);
            line(x, y, x4, y4);
            line(x4, y4, x2, y2);
           // delay(50);
      }
}

int main(void)
{
      int gd = DETECT, gm;
      initgraph(&gd, &gm, NULL);

      int x1,x2,y1,y2,x3,y3,it;

      cout<<"Enter coordinates of initial line(x1,y1,x2,y2,x3,y3):";
      cin>>x1>>y1>>x2>>y2>>x3>>y3;

      //int x1 = 100, y1 = 100, x2 = 400, y2 = 400;

      cout<<"Enter number of iterations:";
      cin>>it;

      //it=1,2,3,4,....

      koch(x1, y1, x2, y2, it);
      koch(x2, y2, x3, y3, it);
      koch(x3, y3, x1, y1, it);

      delay(500000);
      closegraph();
      return 0;
}
```

Conclusion: We implemented fractal patterns by using Koch curves.

Questions:

1. What is the meaning of fractal?
2. What are the different types of fractals?
3. What is the logic to generate Koch curve?
4. What is cups and joins?
5. List out the examples of fractals.

6. Why fractals are very important in computer graphics?
7. What is the difference between fractals and curves?
8. What are the different types of curves?
9. What is Bezier curve?
10. What are the different applications of fractals?