**Practical No 10.A**

**Implementation of fractal - Koch curve/Snowflake.**

**Aim: Write a program to implement a fractal using Koch curve/Snowflake algorithm.**

**Theory:**

The Koch snowflake (also known as the Koch curve, 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” by the Swedish mathematician Helge von Koch. 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.

**Construction**

**Step1:**

Draw an equilateral triangle. You can draw it with a compass or protractor, or just eyeball it if you don’t want to spend too much time drawing the snowflake. It’s best if the length of the sides is divisible by 3, because of the nature of this fractal. This will become clear in the next few steps



**Step2:**

Divide each side in three equal parts. This is why it is handy to have the sides divisible by three.



**Step3:**

Draw an equilateral triangle on each middle part. Measure the length of the middle third to know the length of the sides of these new triangles.



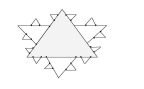
**Step4:**

Divide each outer side into thirds. You can see the 2nd generation of triangles covers a bit of the first. These three-line segments shouldn’t be parted in three.



**Step5:**

Draw an equilateral triangle on each middle part. Note how you draw each next generation of parts that are one 3rd of the mast one.



The Koch curve can be expressed by the following rewrite system (Linden Mayer system):

**Alphabet:** F

**Constants:** +,?

**Axiom:** F

**Production rules:** F? F+F–F+F

Here, F means “draw forward”, – means “turn right 60°”, and + means “turn left 60°”.

To create the Koch snowflake, one would use F++F++F (an equilateral triangle) as the axiom.

**Conclusion: We have implemented a fractal using Koch curve/Snowflake algorithm.**

**Code:**

#include<graphics.h>

#include<conio.h>

#include<math.h>

void koch(int x1, int y1, int x2, int y2, int it)

{

float angle = 60\*M\_PI/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);

} }

int main(void) {

int gd = DETECT, gm;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

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

koch(x1, y1, x2, y2, 4);

getch();

return 0;

}

**Output:**

