#include<stdio.h>

#include<graphics.h>

#include<stdlib.h>

#define SIN 0.86602540 // sin(60 degrees)

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

{

int xx, yy, x[5], y[5], lx, ly, offx = 50, offy = 300;

lx = (x2-x1)/3;

ly = (y2-y1)/3;

x[0] = x1; // Store point p0

y[0] = y1;

x[4] = x2; // Store point p4

y[4] = y2;

x[1] = x[0] + lx; // Store point p1

y[1] = y[0] + ly;

x[3] = x[0] + 2\*lx; // Store point p3

y[3] = y[0] + 2\*ly;

xx = x[3] - x[1]; // Translate point p2 to origin

yy = y[3] - y[1];

x[2] = xx\*(0.5) + yy\*(SIN); // Perform rotation for point p3

y[2] = - xx\*(SIN) + yy\*(0.5);

x[2] = x[2] + x[1]; // Perform inverse translation

y[2] = y[2] + y[1];

if(m>0)

{

koch(x[0], y[0], x[1], y[1], m-1); // Recursive call to Draw part1

koch(x[1], y[1], x[2], y[2], m-1); // Recursive call to Draw part2

koch(x[2], y[2], x[3], y[3], m-1); // Recursive call to Draw part3

koch(x[3], y[3], x[4], y[4], m-1); // Recursive call to Draw part4

}

else

{

line(offx + x[0], offy + y[0] , offx + x[1], offy + y[1]);

line(offx + x[1], offy + y[1] , offx + x[2], offy + y[2]);

line(offx + x[2], offy + y[2] , offx + x[3], offy + y[3]);

line(offx + x[3], offy + y[3] , offx + x[4], offy + y[4]);

}

}

void main()

{

int n, gd, gm;

int x1 = 0, x2 = 550, y1= 0, y2 = 0;

/\* Initialise graphics mode

---------------------------------- \*/

printf("\n Enter the level of curve generation : ");

scanf("%d", &n);

detectgraph(&gd,&gm);

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

koch(x1, y1, x2, y2, n); // Draw Koch curve

getch();

closegraph();

}

// OUTPUT -

