Міністерство освіти і науки України
Національний технічний університет України
«Київський Політехнічний Інститут імені Ігоря Сікорського»

Факультет прикладної математики
Кафедра «Системного програмування і спеціалізованих комп'ютерних систем»

Лабораторна робота №4
З дисципліни «Комп'ютерна графіка» :
«Фрактали»

Виконав:
студент III курсу,
група КВ-41
Яковенко Максим

Перевірив:
_____

Київ-2016

Постановка задачі:
Відтворити на екрані три групи фракталів:

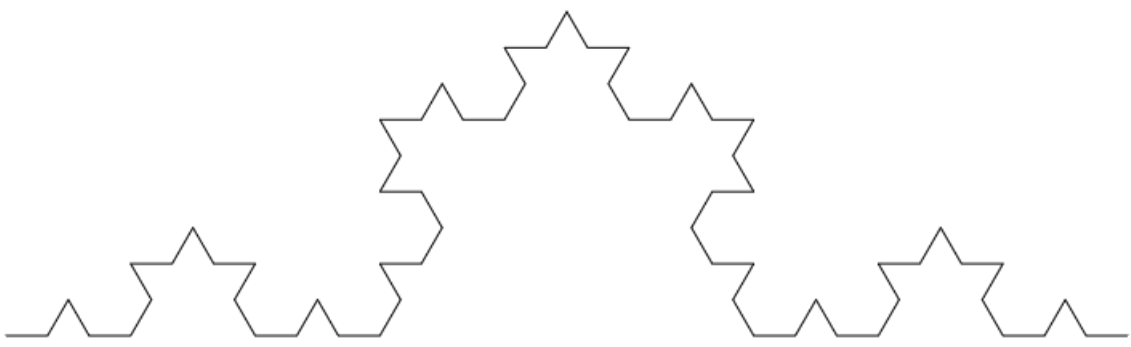
1. Сніжинка та хрест Коха,
2. Хмара Мандельброта,
3. Трикутник і килим Серпінського

Код програми:

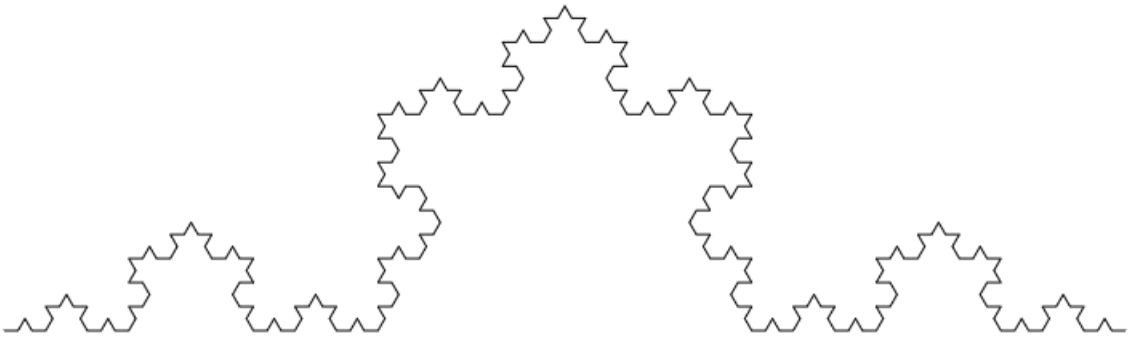
Koch.py

```
from turtle import *
from Tkinter import *
import canvasvg

def snowflake(lengthSide, levels):
    if levels == 0:
        forward(lengthSide)
        return
    lengthSide /= 3.0
    snowflake(lengthSide, levels-1)
    left(60)
    snowflake(lengthSide, levels-1)
    right(120)
    snowflake(lengthSide, levels-1)
    left(60)
    snowflake(lengthSide, levels-1)
if __name__ == "__main__":
    speed(0)
    ht()
    length = 700.0
    penup()
    backward(length/2.0)
    pendown()
    snowflake(length, 3)
    ts = getscreen().getcanvas()
    canvasvg.saveall("Kochlevel3.svg", ts)
    clear()
    penup()
    backward(length/2.0)
    pendown()
    snowflake(length, 4)
    ts = getscreen().getcanvas()
    canvasvg.saveall("Kochlevel4.svg", ts)
    clear()
    penup()
    backward(length/2.0)
    pendown()
    snowflake(length, 5)
    ts = getscreen().getcanvas()
    canvasvg.saveall("Kochlevel5.svg", ts)
```

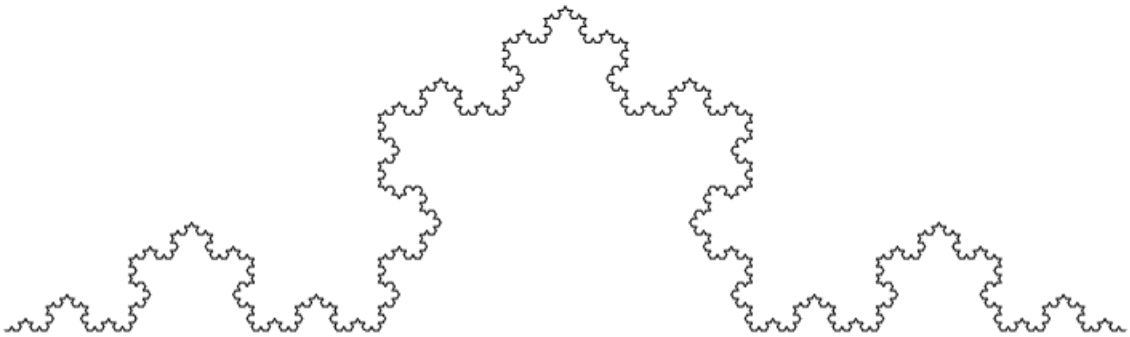
Результат роботи:
Рівень 3



Рівень 4:



Рівень 5:

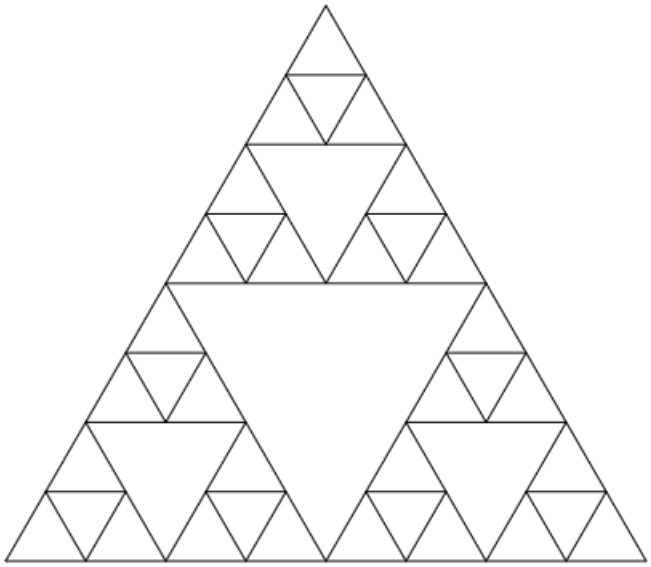


Sierpinsky.py

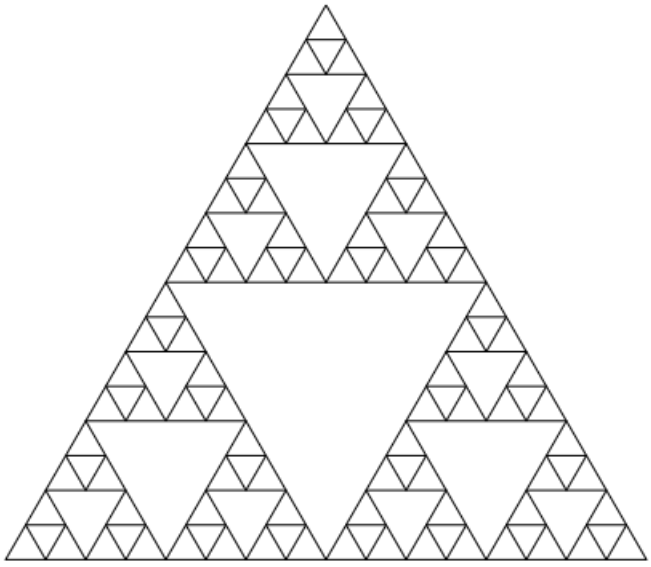
```
from numpy import *
import turtle
from Tkinter import *
import canvasvg

def Left(turn, point, fwd, angle, turt):
        turt.left(angle)
        return [turn, point, fwd, angle, turt]
def Right(turn, point, fwd, angle, turt):
        turt.right(angle)
        return [turn, point, fwd, angle, turt]
def Forward(turn, point, fwd, angle, turt):
        turt.forward(fwd)
        return [turn, point, fwd, angle, turt]

def DrawSierpinskiTriangle(level, ss=400):
        # typical values
        turn = 0            # initial turn (0 to start horizontally)
        angle=60.0                  # in degrees

        turtle.hideturtle()
        turtle.screensize(ss,ss)
        turtle.penup()
        turtle.degrees()

        fwd0        = float(ss)
        point=array([-fwd0/2.0, -fwd0/2.0])

        decode    = {'-':Left, '+':Right, 'X':Forward, 'H':Forward}
        axiom     = 'H--X--X'

        turtle.goto(point[0], point[1])
        turtle.pendown()
        turtle.hideturtle()
        turt=turtle.getpen()
        startposition=turt.clone()
```

```
        fwd      = fwd0/(2.0**level)
        path     = axiom
        for i in range(0,level):
                path=path.replace('X','XX')
                path=path.replace('H','H--X++H++X--H')

        # Draw it.
        for i in path:
                [turn, point, fwd, angle, turt]=decode[i](turn, point, fwd, angle, turt)

turtle.speed(0)
DrawSierpinskiTriangle(3)
ts = turtle.getscreen().getcanvas()
canvasvg.saveall("Sierpinski3.svg", ts)
turtle.clear()
DrawSierpinskiTriangle(4)
ts = turtle.getscreen().getcanvas()
canvasvg.saveall("Sierpinski4.svg", ts)
turtle.clear()
DrawSierpinskiTriangle(5)
ts = turtle.getscreen().getcanvas()
canvasvg.saveall("Sierpinski5.svg", ts)
```

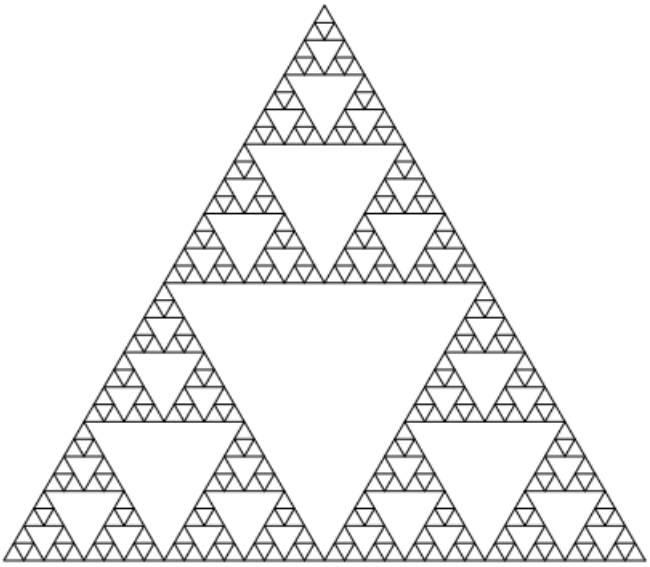
Результат роботи:
Рівень 3:



Рівень 4:



Рівень 5:

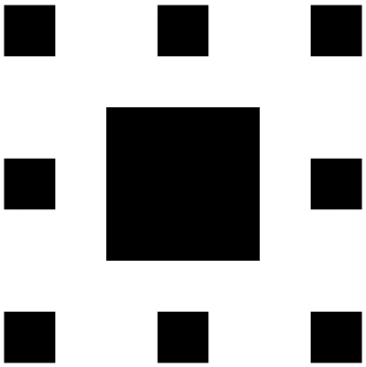


Sierpinskicarpet.py

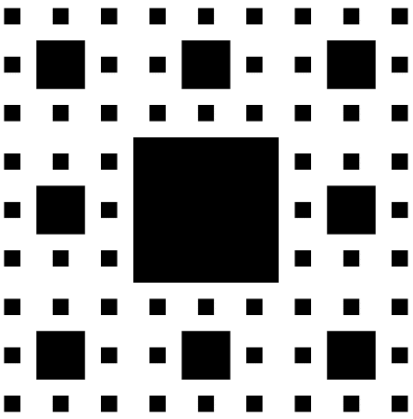
```
import numpy as np
from PIL import Image

numLevels = 6
imageSize = 3**numLevels

# Create the image and fill it with white
img = np.empty([imageSize, imageSize, 3], dtype=np.uint8)
img.fill(255)
color = np.array([0, 0, 0], dtype=np.uint8)

for level in range(0, numLevels + 1):
        stepSize = 3**(numLevels - level)
        for x in range(0, 3**level):
                if x % 3 == 1:
                        for y in range(0, 3**level):
                                if y % 3 == 1:
                                        img[y*stepSize:(y+1)*stepSize, x*stepSize:(x+1)*stepSize] = color

        # Send to PIL and save
        outputFilename = "sierpinski%d.bmp" % level
        Image.fromarray(img).save(outputFilename)
```

Результат роботи:

Рівень 2:



Рівень 3:



Mandelbrot.py

```
from PIL import Image

max_iteration = 1000
x_center = -1.0
y_center =  0.0
size = 750

im = Image.new("RGB", (size,size))
for i in xrange(size):
    for j in xrange(size):
        x,y = ( x_center + 4.0*float(i-size/2)/size,
                y_center + 4.0*float(j-size/2)/size)

        a,b = (0.0, 0.0)
        iteration = 0

        while (a**2 + b**2 <= 4.0 and iteration < max_iteration):
            a,b = a**2 - b**2 + x, 2*a*b + y
            iteration += 1
        if iteration == max_iteration:
            color_value = 255
        else:
            color_value = iteration*10 % 255
        im.putpixel( (i,j), (color_value, color_value, color_value))

im.save("mandelbrot.png", "PNG")
```

Результат роботи: