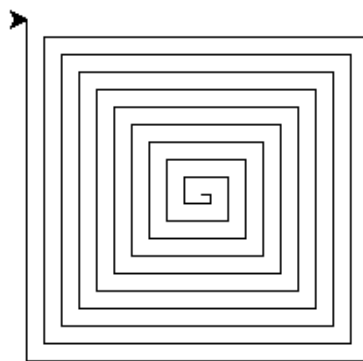


Примеры решения задач

1. Квадратная спираль



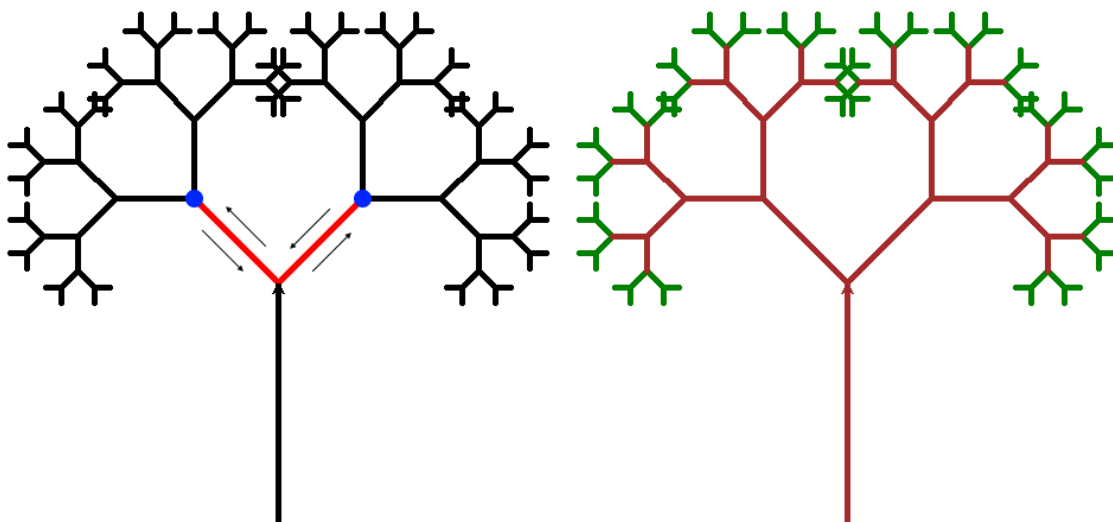
```
import turtle

LENGTH = 5 # начальная длина стороны
DELTA = 10 # увеличение стороны
INC = 20 # количество увеличений

for i in range(INC):
    for k in range(2):
        turtle.forward(LENGTH)
        turtle.right(90)
    LENGTH += DELTA
```

2. Простое фрактальное дерево

Фракталы – это рекурсивные геометрические фигуры или структуры, обладающие свойством самоподобия.



```

import turtle

MAX_DEPTH = 6 # максимальная глубина рекурсии
LENGTH = 100 # начальная длина ветви

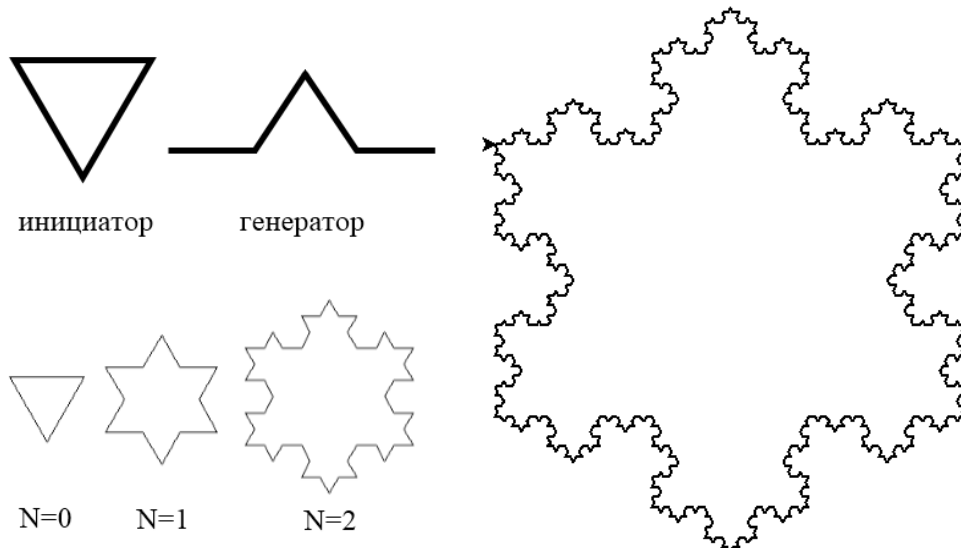
# функция окрашивания ветвей в зависимости от глубины рекурсии
def setcolor(depth):
    if depth > 2:
        turtle.color('brown')
    else:
        turtle.color('green')

# функция рисования ветвей
def tree(depth, length):
    if depth == 0: return 0
    turtle.left(45)
    turtle.forward(length)
    tree(depth-1, length/1.5)
    setcolor(depth)
    turtle.back(length)
    turtle.right(90)
    turtle.forward(length)
    tree(depth-1, length/1.5)
    setcolor(depth)
    turtle.back(length)
    turtle.left(45)

# настройки пера
turtle.width(5)
turtle.color('brown')

turtle.left(90) # Черепаха смотрит вверх
turtle.back(200) # ствол дерева
turtle.forward(200)
tree(MAX_DEPTH, LENGTH) # рисование ветвей дерева
    
```

3. Снежинка Коха треугольная



```
import turtle

# функция построения кривой Коха
def koch_curve(N, k):
    if N == 0:
        turtle.forward(k)
        return
    # генератор
    koch_curve(N - 1, k / 2) # рекурсивный вызов вместо движения вперед
    turtle.left(60)
    koch_curve(N - 1, k / 2)
    turtle.right(120)
    koch_curve(N - 1, k / 2)
    turtle.left(60)
    koch_curve(N - 1, k / 2)

# функция построения снежинки Коха
def koch_snowflake(N, k):
    for i in range(3): # инициатор
        koch_curve(N, k) # сторона треугольника - кривая Коха
        turtle.right(120)

turtle.width(2)
koch_snowflake(4, 70)
```