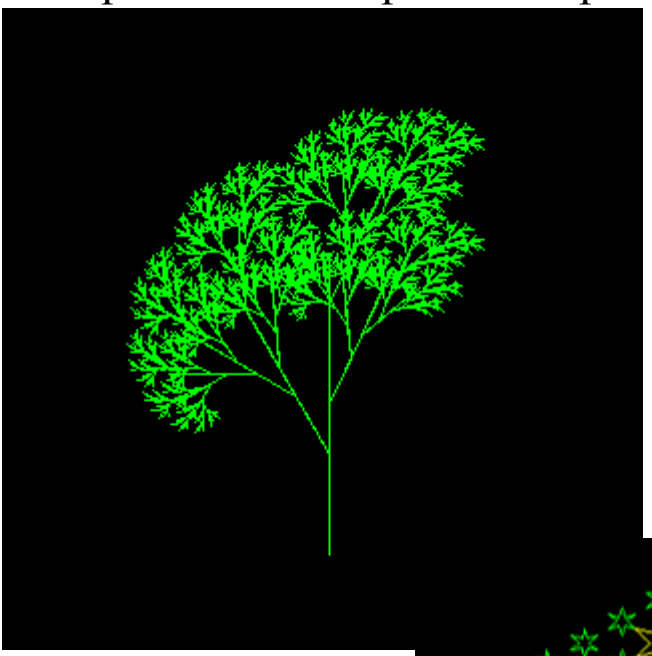
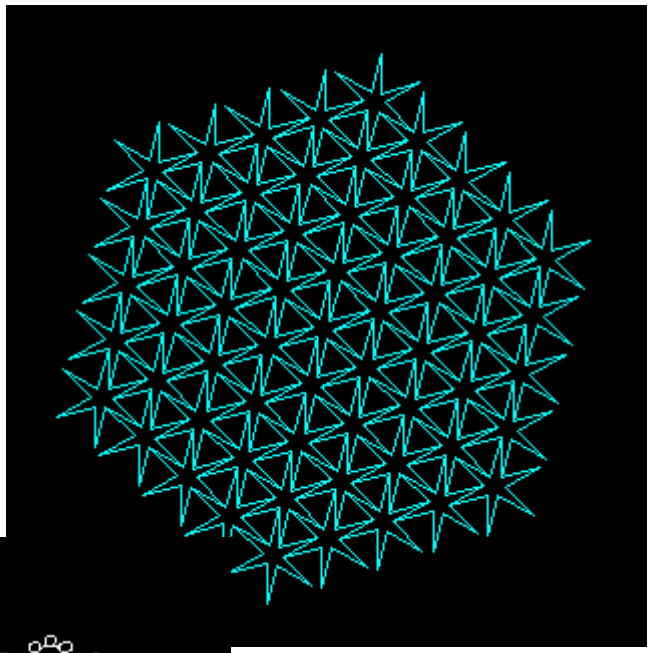
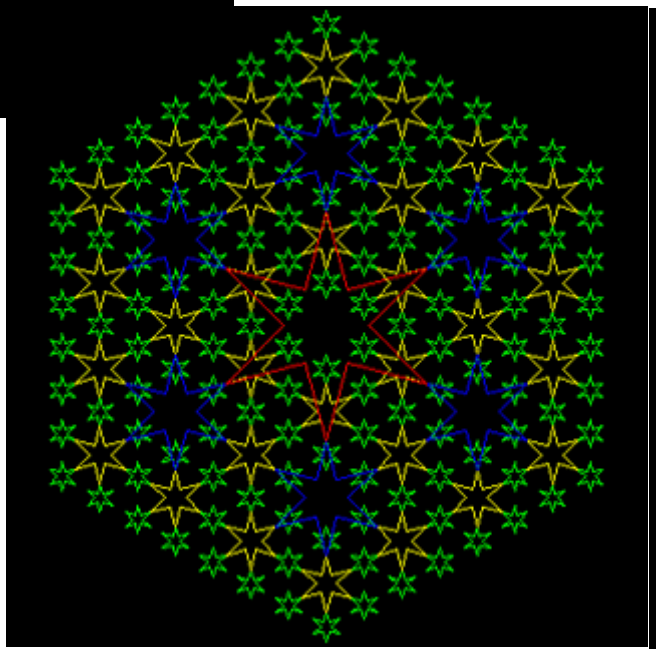
Рекурсии в алгоритмах

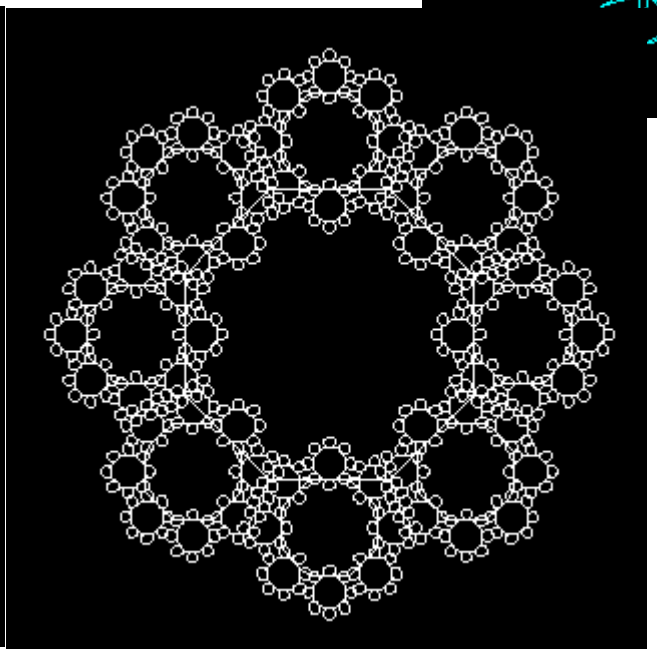
Рекурсия - это такая организация алгоритма, при которой процедура обращается к самой себе. Сама процедура называется рекурсивной.

В жизни вам не раз приходилось сталкиваться с рекурсией. Вспомните хотя бы стихотворение "У попа была собака" или то, как, сидя в поезде, вы ловили свое отражение в зеркале, которое отражалось в зеркале напротив, которое отражалось в зеркале напротив...



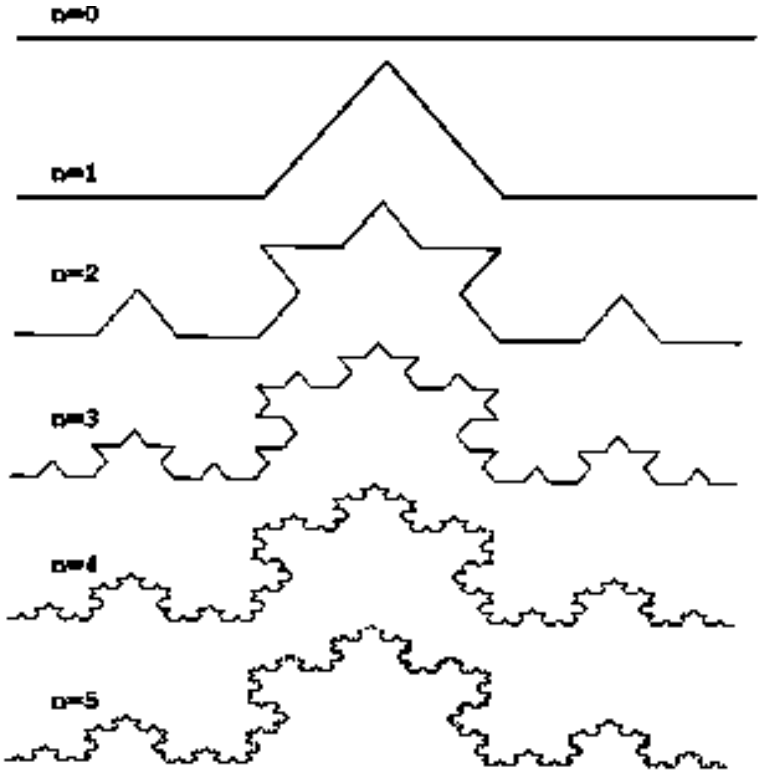






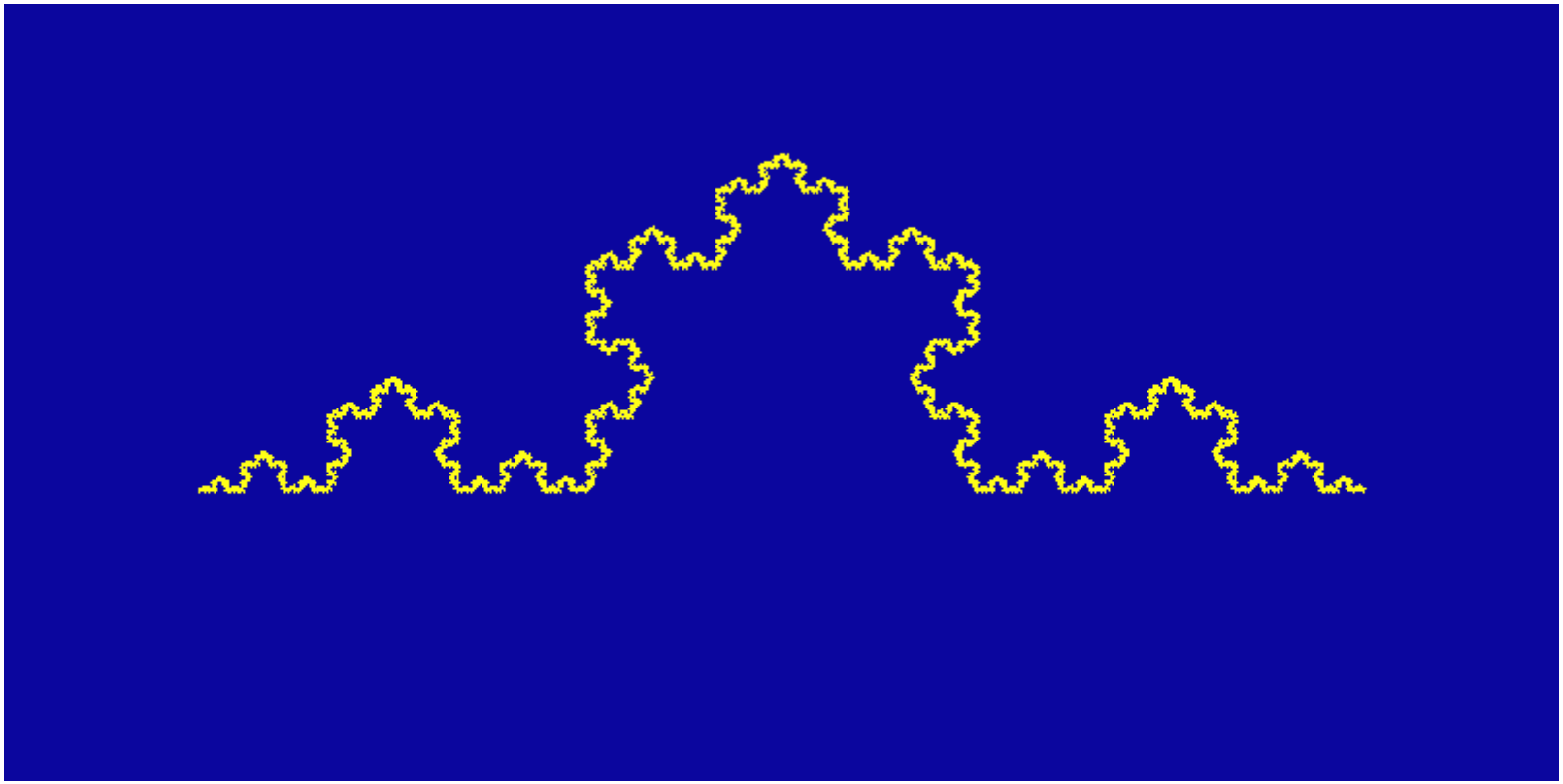
1. Построение триадной кривой Кох.

Рассмотрим триадную кривую Кох. Построение кривой начинается с отрезка единичной длины - это 0-е поколение кривой Кох. Далее каждое звено (в нулевом поколении один отрезок) заменяется на образующий элемент, обозначенный на рис.1 через n=1. В результате такой замены получается следующее поколение кривой Кох. В 1-ом поколении - это кривая из четырех прямолинейных звеньев, каждое длиной по 1/3. Для получения 3-го поколения проделываются те же действия - каждое звено заменяется на уменьшенный образующий элемент. Итак, для получения каждого последующего поколения, все звенья предыдущего поколения необходимо заменить уменьшенным образующим элементом. Кривая n-го поколения при любом конечном n называется предфракталом. На рисунке представлены пять поколений кривой.

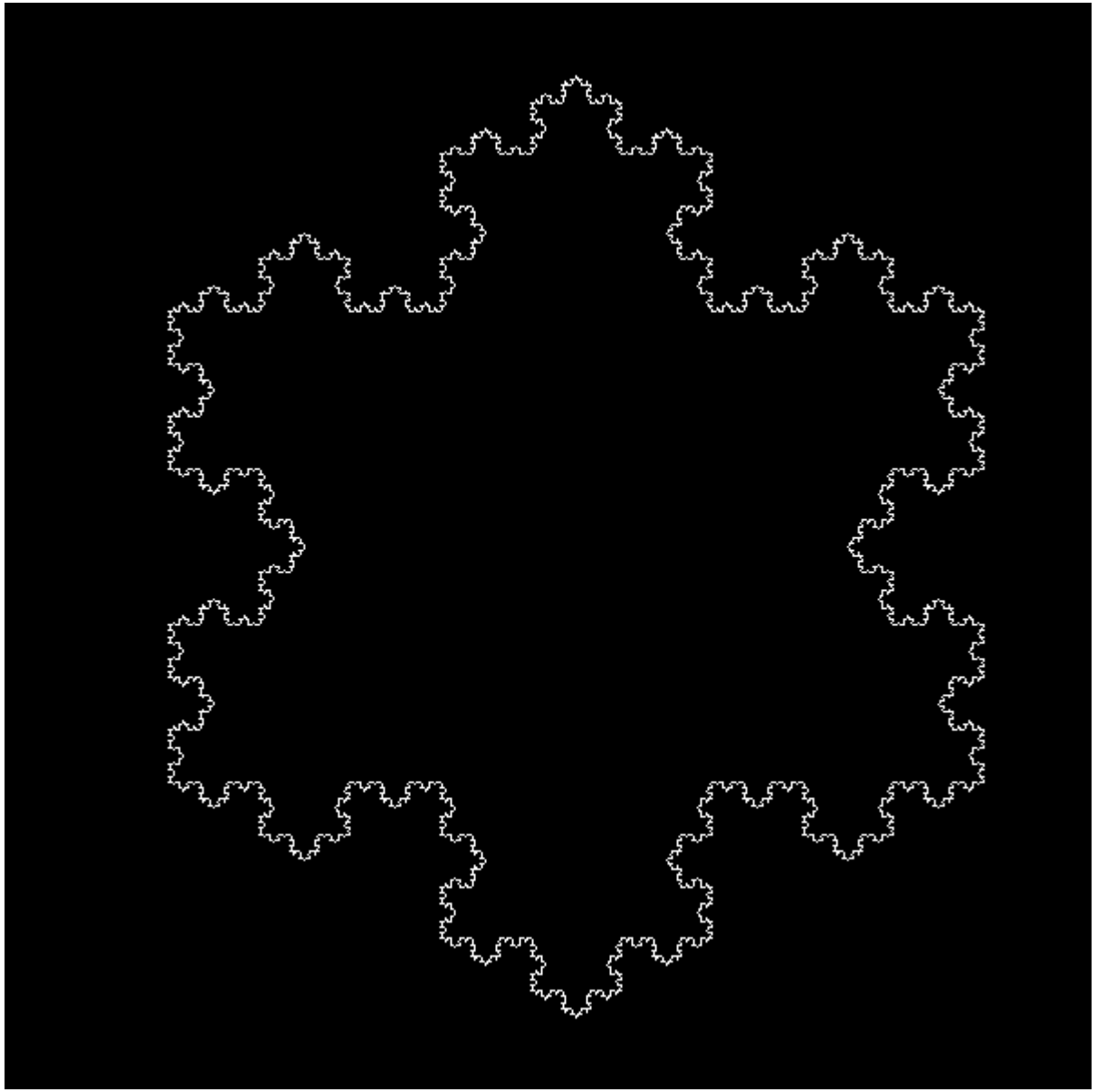


В kTurtle наберите следующую программу, разберите, как она работает и попробуйте изменить параметры. Посмотрите, что получилось. Результат сохраните под именем logo17.logo

```
сброс
НРХ 800, 400
ИДИ 100, 250
НЦХ 11, 6, 158
НЦП 255, 255, 23
НШП 3
спрячь

ПР 90

выучи кох дл, ст [
если ст>0 [
  ст=ст-1
  дл=дл/3
  кох дл, ст
  ЛВ 60
  кох дл, ст
  ПР 120
  кох дл, ст
  ЛВ 60
  кох дл, ст
  ] иначе [
    ВП 3*дл
    ]
]

кох 200, 6
```

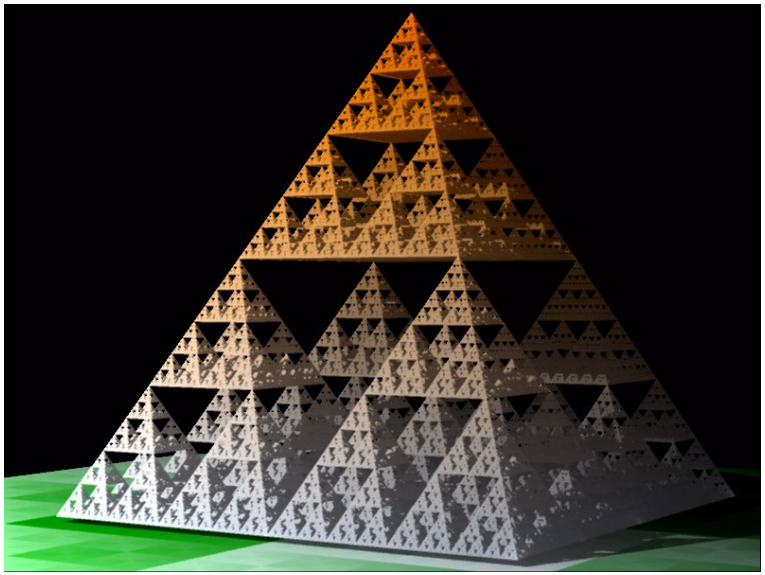


Если три раза повторить триадную кривую Коха с поворотом, то получится «снежинка Коха»:



Исправьте предыдущую программу до получения такого результата и сохраните под именем logo18.logo

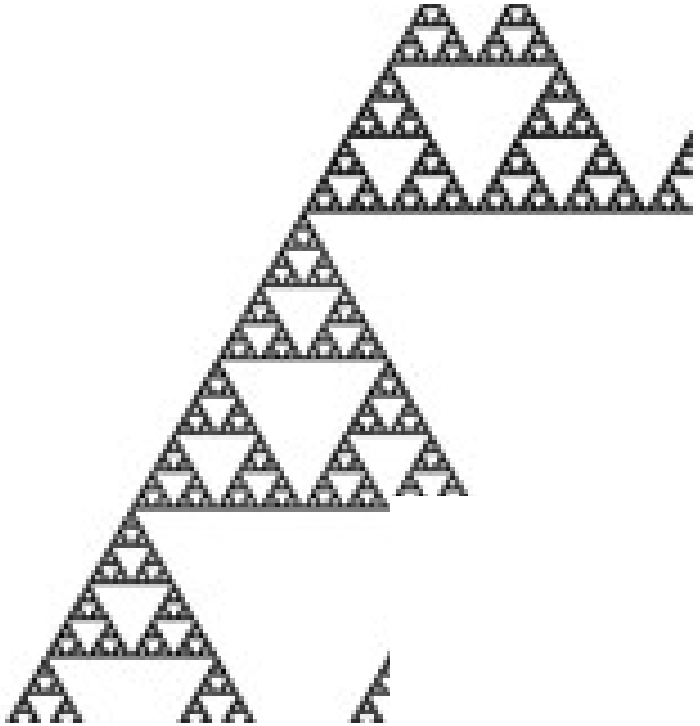
2. Построение треугольника Серпинского.

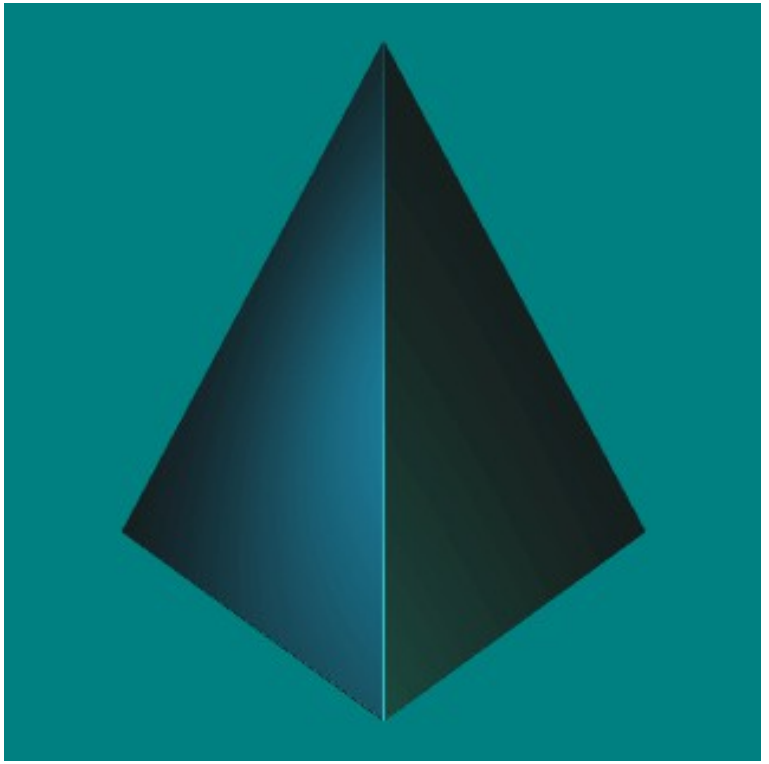


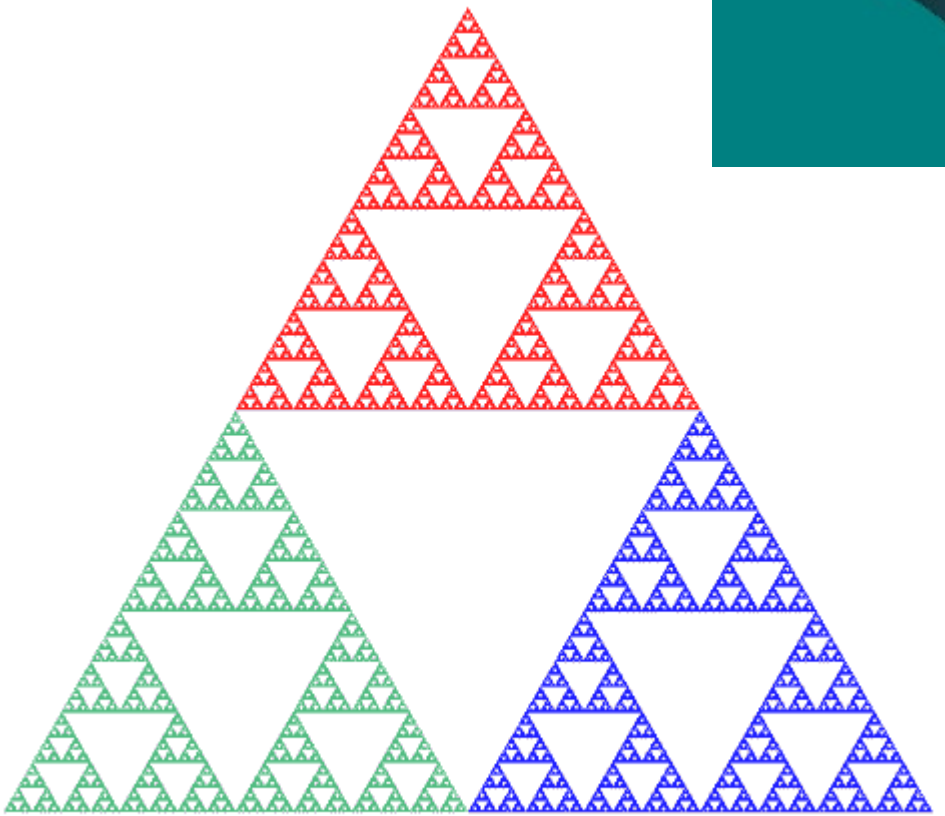
Треугольник Серпинского — фрактал, один из друмерных аналогов множества Кантора, предложенный польским математиком Серпинским в 1915 году. Также известен как «решётка» или «салфетка» Серпинского.

Берётся сплошной равносторонний треугольник, на первом шаге из центра удаляется внутренность срединного треугольника. На втором шаге удаляется три срединных треугольника из трёх оставшихся треугольников и т. д. После бесконечного повторения этой процедуры, от сплошного треугольника остаётся подмножество — треугольник Серпинского.

Треугольник Серпинского можно также получить по следующему алгоритму:
1. Взять три точки на плоскости, и нарисовать треугольник.
2. Случайно выбрать любую точку внутри треугольника, и продвинуться на половину расстояния от этой точки к любой из трех вершин треугольника.
3. Отметить текущую позицию.
4. Повторить с шага 2.

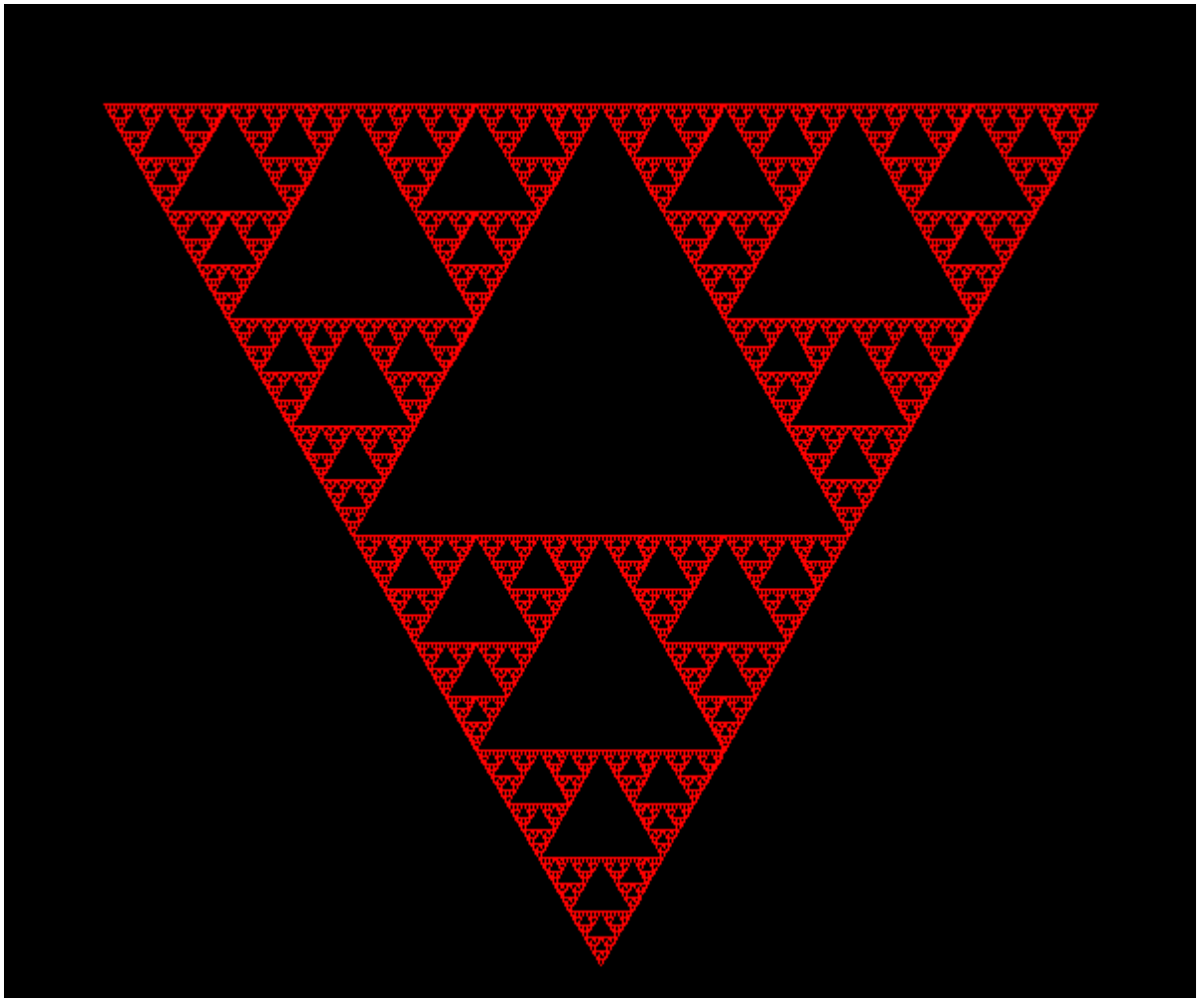




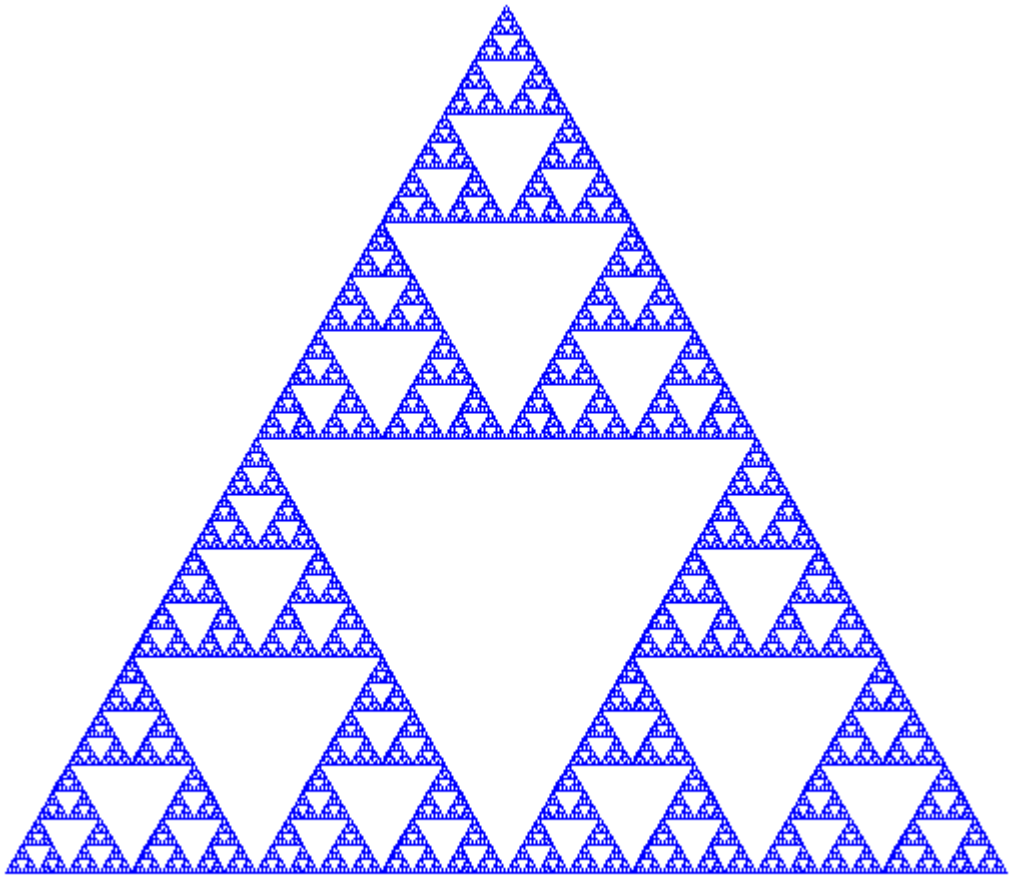


В kTurtle наберите программу, разберите, как она работает и попробуйте изменить параметры. Посмотрите, что получилось. Результат сохраните под именем logo19.logo

```
сброс
НРХ 600, 500
ИДИ 50, 50
НЦХ 0,0,0
НЦП 0,0,255
спрячь

ПР 90

выучи треуг дл [
если дл>2 [
  повтори 3 [
    треуг дл/2
    ВП дл
    ПР 120
          ]
        ]
]

треуг 500
```



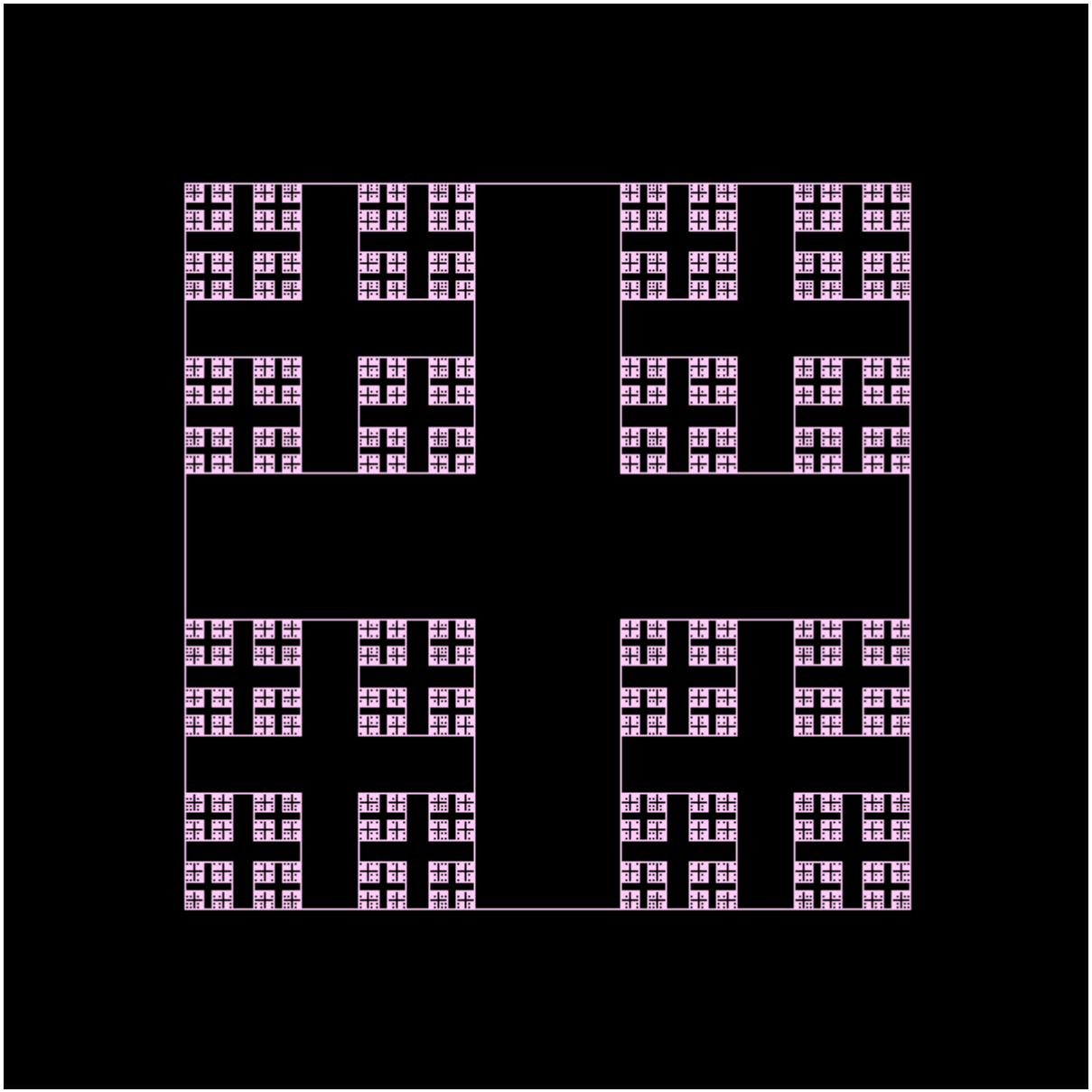
Переверните треугольник. Для этого исправьте предыдущую программу и сохраните под именем logo20.logo



3. Построение квадрата Серпинского

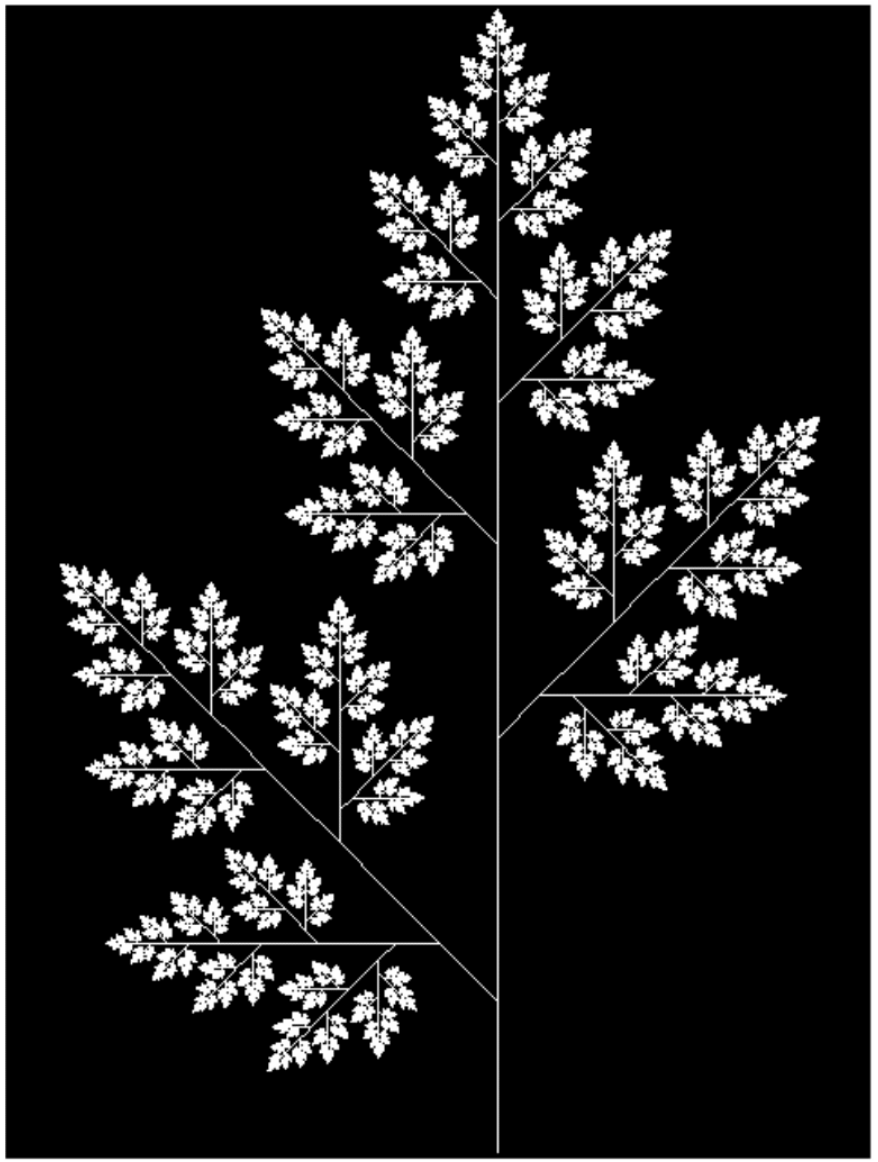
В kTurtle наберите программу, разберите, как она работает и попробуйте изменить параметры. Посмотрите, что получилось. Результат сохраните под именем logo21.logo

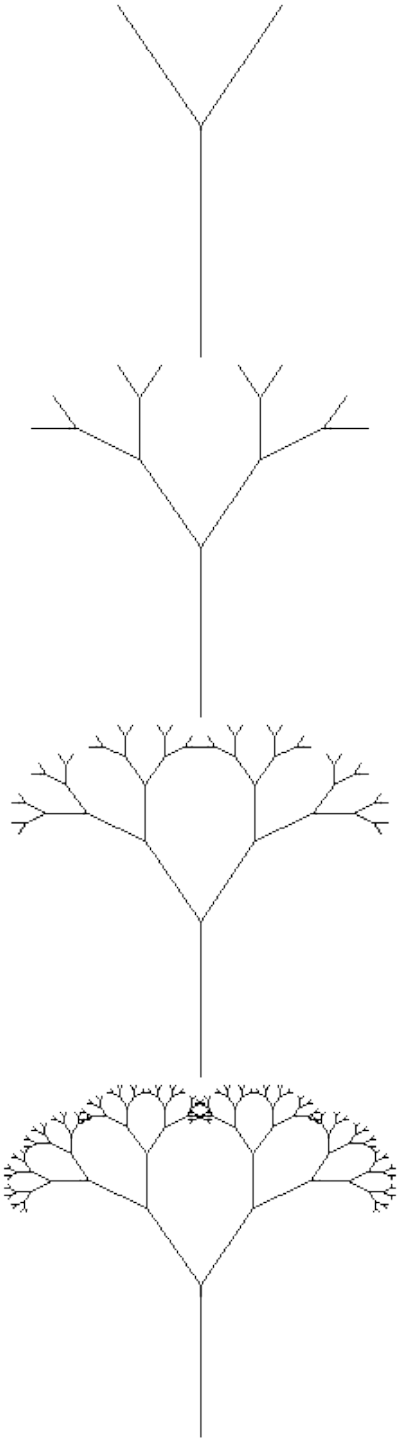
```
сброс
НРХ 600, 600
ИДИ 100, 500
НЦХ 0,0,0
НЦП 255, 204, 249
спрячь

ПР 90

выучи кв дл
[
если дл>1 [
  повтори 4 [
    кв дл*0.4
    ВП дл
    ЛВ 90
          ]
        ]
]

кв 401
```

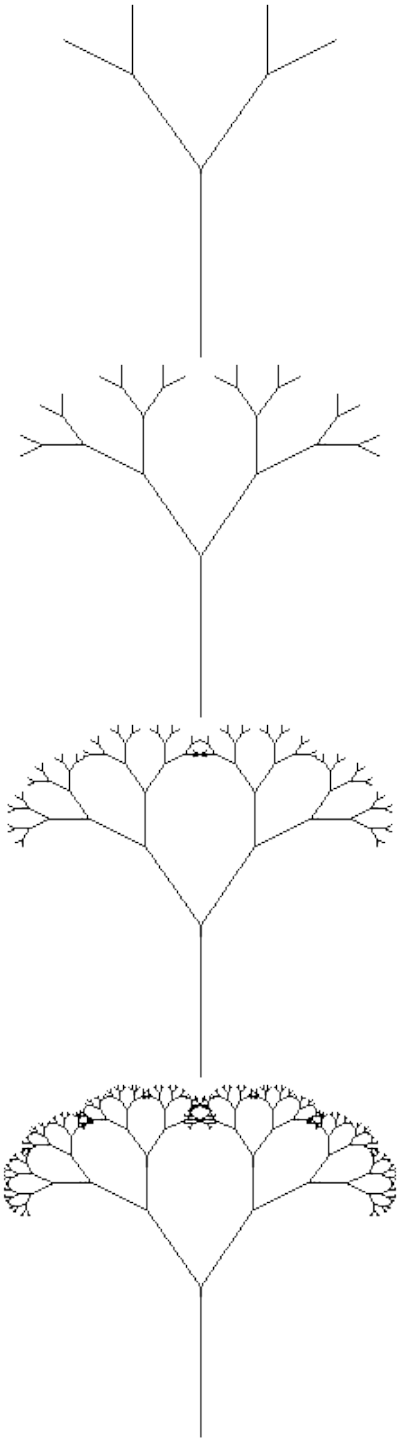


4. Построение деревьев.

Дерево - широко известный пример рекурсивной графики.

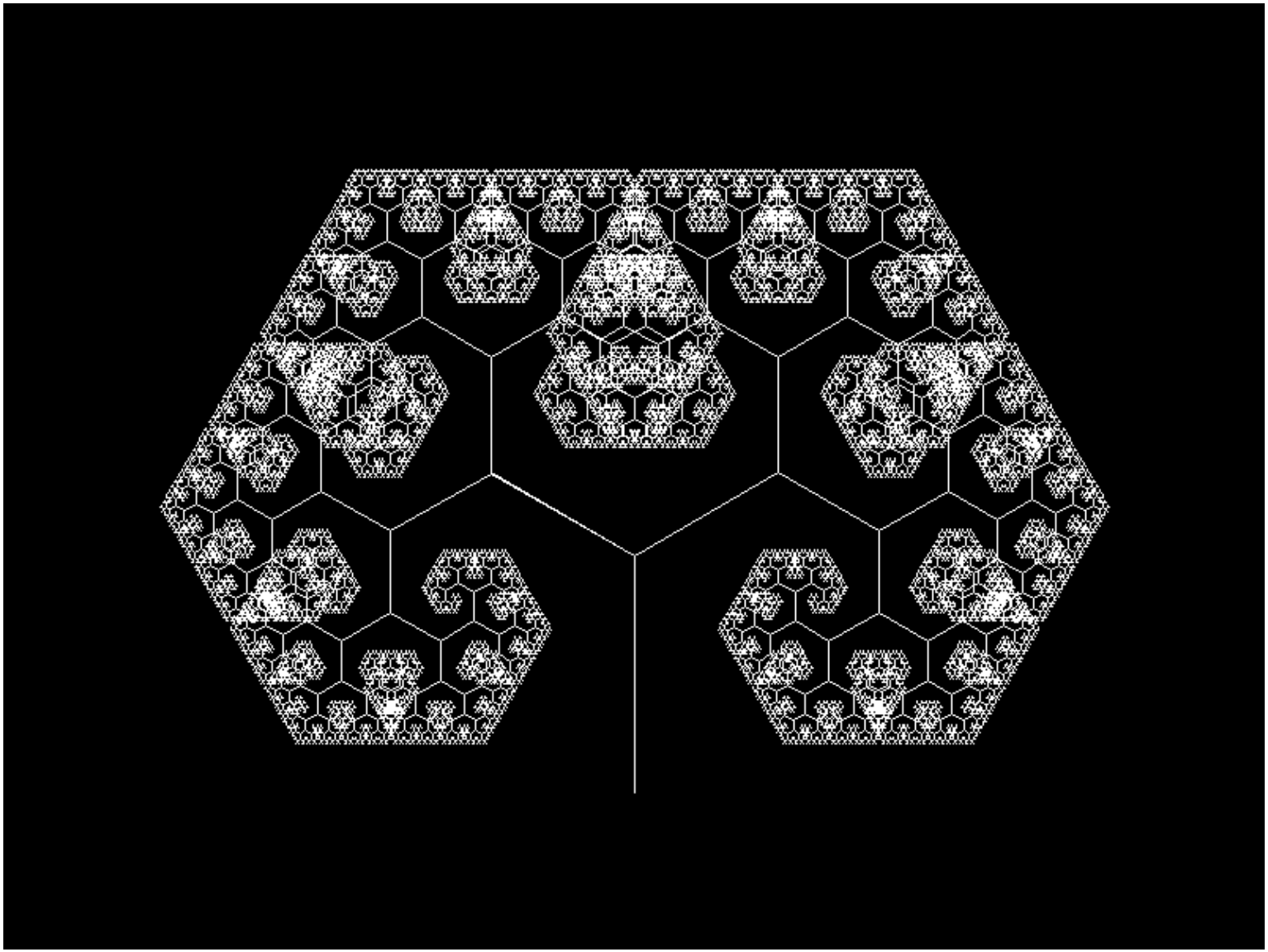






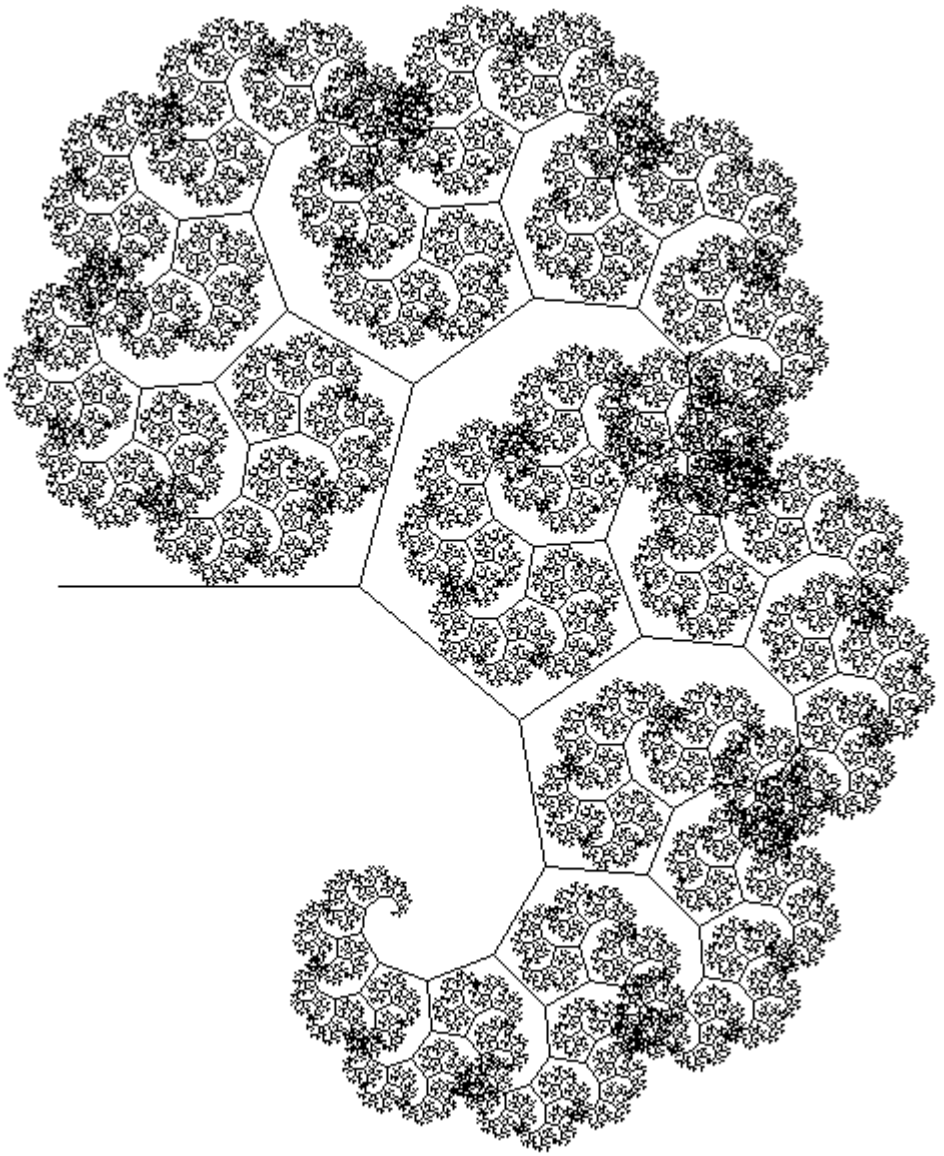
В kTurtle наберите программу, разберите, как она работает и попробуйте изменить параметры. Посмотрите, что получилось. Результат сохраните под именем logo22.logo

```
сброс
НРХ 800, 600
ИДИ 400, 500
НЦХ 0, 0, 0
НЦП 255, 255, 255

выучи дерево ст, дл
[
если ст>0 [
ВП дл
ЛВ 60
дерево ст-1, 0.7*дл
ПР 120
дерево ст-1, 0.7*дл
ЛВ 60
Лв 180
ВП дл
ЛВ 180
]
]

дерево 15,150
спрячь
```

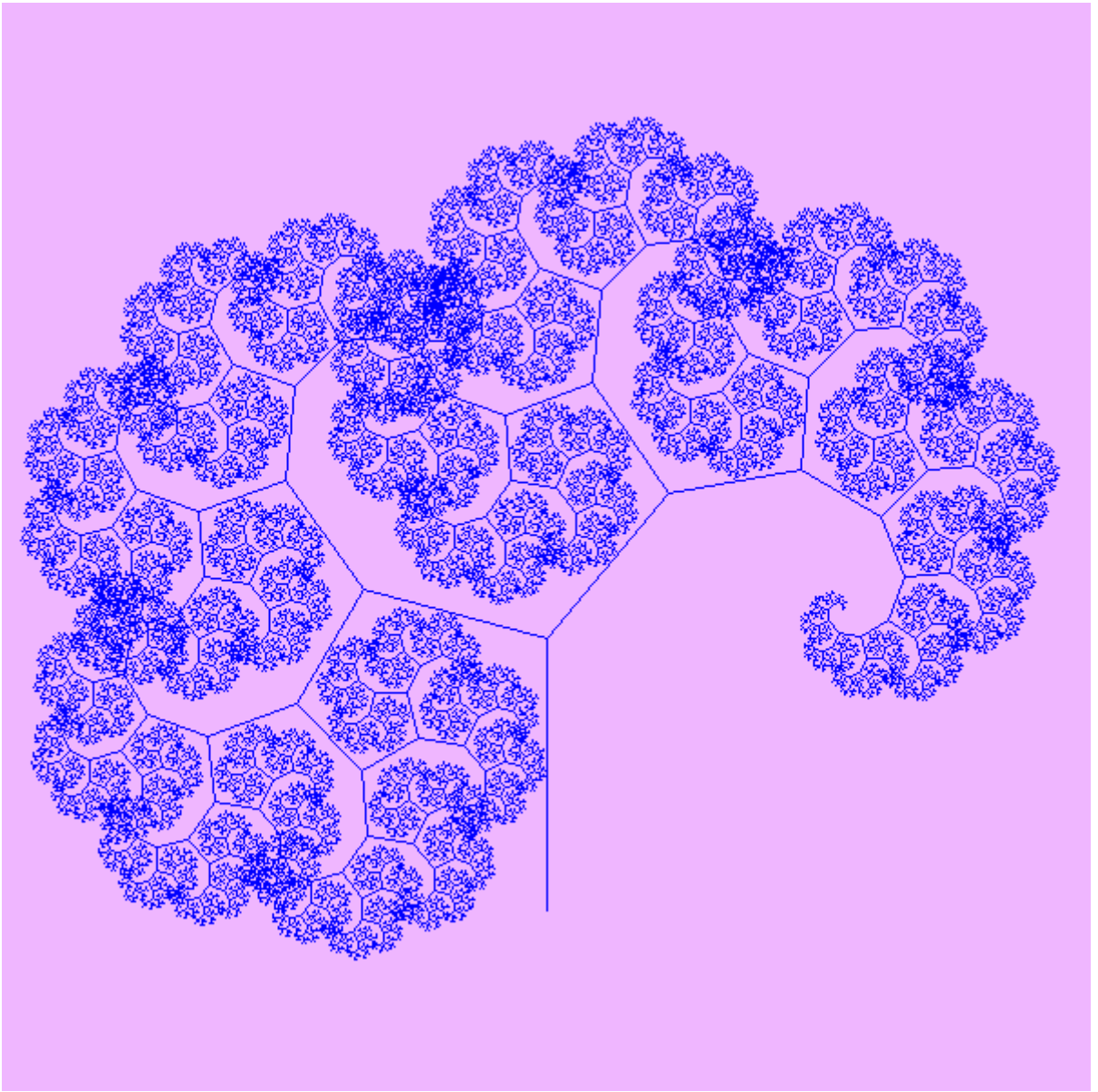


В kTurtle наберите программу, разберите, как она работает и попробуйте изменить параметры. Посмотрите, что получилось. Результат сохраните под именем logo23.logo

```
сброс
НРХ 600, 600
ИДИ 100, 300
НЦХ 0,0,0
НЦП 255,0,0
спрячь

ПР 90

выучи дерево2 ст, дл
[
если ст>0 [
  ВП дл
  ЛВ 75
  дерево2 ст-1, 0.7*дл
  ПР 115
  дерево2 ст-1, 0.7*дл
  ЛВ 40
  НД дл
        ]
]

дерево2 15, 150
```



Переверните дерево на 90 градусов, для этого исправьте предыдущую программу и сохраните под именем logo24.logo



В kTurtle наберите программу, разберите, как она работает и попробуйте изменить параметры. Посмотрите, что получилось. Результат сохраните под именем logo25.logo

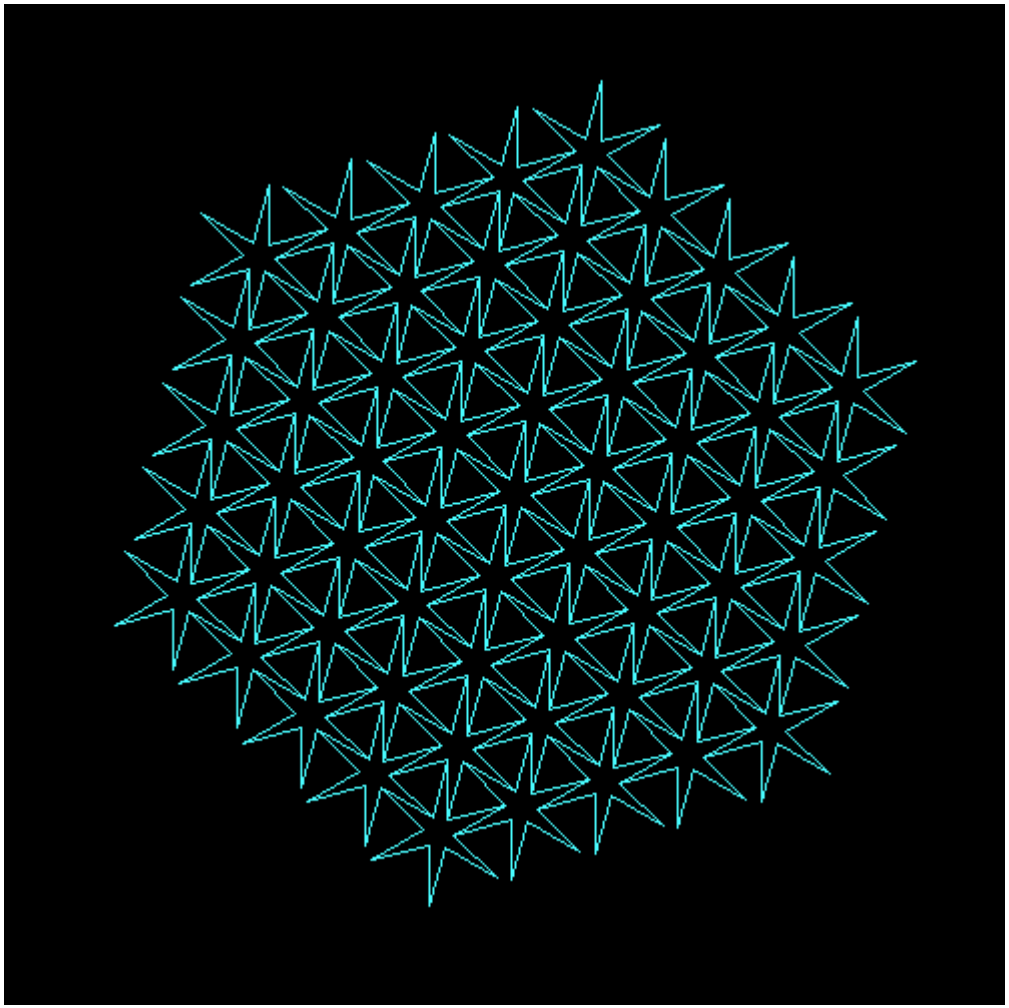
```
сброс
НРХ 600, 600
ИДИ 330, 550
НЦХ 211, 234, 255
НЦП 60, 127, 71
спрячь

выучи дерево4 дл [
если дл>5 [
  ВП дл/3
  ЛВ 30
  дерево4 дл*2/3
  ПР 30
  ВП дл/6
  ПР 25
  дерево4 дл/2
  ЛВ 25
  ВП дл/3
  ПР 25
  дерево4 дл/2
  ЛВ 25
  ВП дл/6
  НД дл
  ] иначе [
  ВП дл
  НД дл
 ]
]

дерево4 350
```

4. Построение звёзд.

С помощью рекурсии можно нарисовать и другие красивые изображения.

В kTurtle наберите программу, разберите, как она работает и попробуйте изменить параметры. Посмотрите, что получилось. Результат сохраните под именем logo26.logo

```
сброс
НРХ 500, 500
ИДИ 250, 280
НЦХ 0, 0, 0
НЦП 72, 252, 255
спрячь

выучи звезда ур
[
  повтори 6 [
  если ур>=1 [
     ВП 30
     ПР 180
     звезда ур-1
     ПР 180
     ЛВ 180-(360/6+15)
     ВП 30
     ПР 180-15
   ]
 ]
]

звезда 5
```



Меняя параметры можно получить совершенно различные узоры.
Исправьте предыдущую программу для того, чтобы получить следующие узоры и сохраните под именами logo27.logo и logo28.logo