

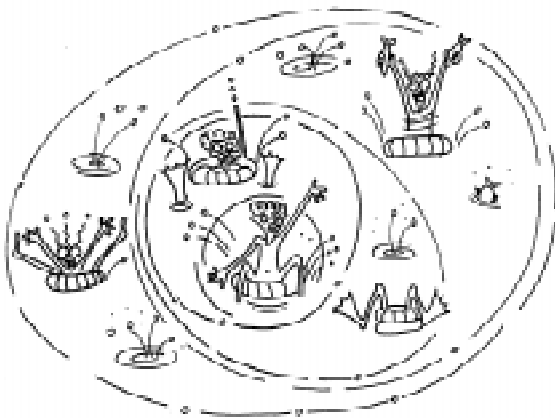
ПРИМЕРЫ И ЗАДАЧИ НА ТЕМУ «РЕКУРСИВНЫЕ АЛГОРИТМЫ И ИХ ПОСТРОЕНИЕ»

Помните детское стихотворение-считалку про 10 негрят? Оно может служить прологом к любой работе на тему «Рекурсия».

10 негрят пошли купаться в море,
10 негрят резвились на просторе,
Один из них пропал – и вот вам результат:

9 негрят пошли купаться в море,
9 негрят резвились на просторе,
Один из них пропал – и вот вам результат:
... ..

1 (из) негрят пошли(ел) купаться в море,
1 (из) негрят резвились(ся)на просторе,
Один из них пропал – и вот вам результат:
Нет больше негрят!



Первые три строчки этого стихотворения повторяются 10 раз с небольшим изменением - число негрят уменьшается с каждым разом на единицу. И только, когда число негрят уменьшилось до нуля, стихотворение заканчивается единственной строчкой «Нет больше негрят!». Попробуем написать

рекурсивную процедуру, печатающую это стихотворение.

```
Procedure Negr(k: integer);  
{k-число негрят, параметр процедуры}  
Begin  
If k=0 {проверка, что число  
негрят равно нулю}  
then writeln('Нет больше негрят!')  
{выход из рекурсии}  
else begin  
writeln(k, ' негрят пошли  
купаться в море,');  
writeln(k, ' негрят  
резвились на просторе,');  
writeln('Один и них пропал  
- и вот вам результат:');  
Negr(k-1);  
{Вызов процедуры с уменьшенным  
на 1 параметром}  
end  
End;
```

Вызов этой процедуры в основной программе будет выглядеть так: Negr(10).

Интересным является применение рекурсии при создании рисунков.

Пример 1.

Рассмотрим простейший рисунок из окружностей разных радиусов (рисунок 1).

Если взглянуть на него внимательно, то можно заметить, что рисунок начинается с центральной окружности самого

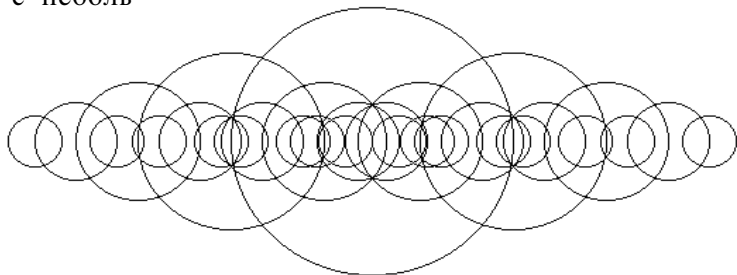


Рисунок 1.

большого радиуса. Затем осуществляется переход на концы горизонтального диаметра окружности, которые должны играть роль центров двух окружностей меньшего радиуса (примерно в полтора раза). Этот же процесс повторяется и с этими двумя окружностями, и с полученными четырьмя новыми, и так далее до тех пор, пока уменьшающийся радиус окружности не станет меньше первоначального в 1,5·1,5·1,5·1,5 раз (если посчитать, то должно выполняться четыре вложенных вызова рекурсивной процедуры).

Рекурсивная процедура, выполняющая такой рисунок, должна иметь в качестве передаваемых параметров координаты центра окружности и величину радиуса.

Программа, при помощи которой будет нарисован этот рисунок, может быть написана так:

```

Program RIS1;
Uses Graph;
Var x,y,r,gd,gm:integer;
Procedure Ris(x,y,r:integer);
begin
if r>=10
  then begin
    Circle(x,y,r);
    Ris(x-r,y,r*2 div 3);
    Ris(x+r,y,r*2 div 3)
  end
end;
BEGIN
gd:=detect;
gm:=1;
InitGraph(gd,gm,'');
  Ris(320,240,100);
Readln;
CloseGraph;
END.

```

Пример 2.

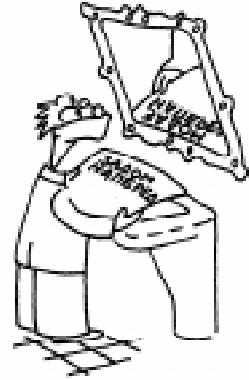
На входном потоке находится слово (последовательность литер), заканчивающаяся пробелом. Написать программу, которая позволит напечатать это слово «задом наперед», то есть выписывая буквы слова с конца.

В этом примере мы опишем процедуру, у которой не будет параметров, так как при каждом ее вызове будет вводиться одна литера, которая будет сравниваться с пробелом. Сама же программа будет состоять только из вызова этой процедуры.

```

Program REVERSE;
Procedure REV;
Var c:char;
begin read(c);
if c<>' '
then begin REV; write(c)
  end
end;
BEGIN REV
END.

```



Теперь переходим к более сложным задачам.

Пример 3.

Фрактали называются множества, части которых являются повторением образов самих множеств. Изображения фракталов вызывают обычно у всех большой интерес.

Рассмотрим процесс сгибания бумажной полоски: если взять полоску бумаги, согнуть ее пополам k раз и развернуть полоску так, чтобы углы на сгибах стали равны 90° , то, посмотрев на торец полоски, можно увидеть ломаную, которая называется «драконовой» ломаной k -го порядка, где k – количество сгибов. Схема этого процесса изображена на рисунке 2.

Опишем способ создания драконовой ломаной.

На ломаной нулевого порядка, как на гипотенузе, строим прямой угол, на по-

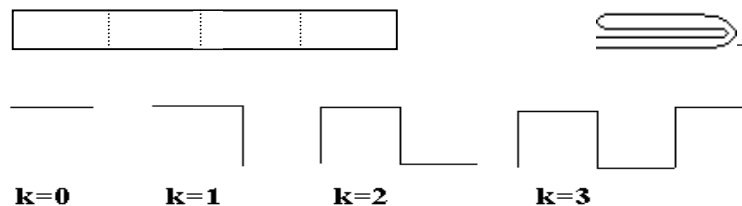
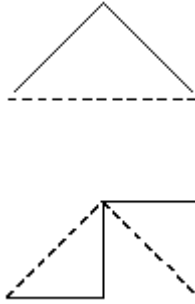


Рисунок 2.

лученных сторонах прямого угла строим тоже прямые углы, но один развернут в правую сторону, а другой - в левую. Данный процесс повторяется k раз.

Напишем соответствующую программу, которая по заданному числу k рисует «драконову» ломаную k -го порядка.



Если с помощью этой программы построить ломаную дракона 14-го порядка, то мы получим образ множества, называемого фракталом Хартера-Хейтуэя, рисунок которого приведен на рисунке 3.

```

Program DRACON;
Uses Graph;
Var Gd,Gm,k:integer;
Procedure st (x1, y1, x2, y2,
k:integer);
var xn,yn:integer;
begin
If k > 0 then
begin
xn:=(x1+x2)div 2 + (y2-y1)div 2;
yn:=(y1+y2)div 2 - (x2-x1)div 2;
st(x1,y1,xn,yn,k-1);
st(x2,y2,xn,yn,k-1);
end
else Line(x1,y1,x2,y2);
end;
BEGIN
Gd:=Detect; Gm:=1;
InitGraph(Gd,Gm,'');
writeln ('Введите номер уровня ');
readln(k);
st(200, 300, 500, 300, k);
readln;
CloseGraph;
END.

```

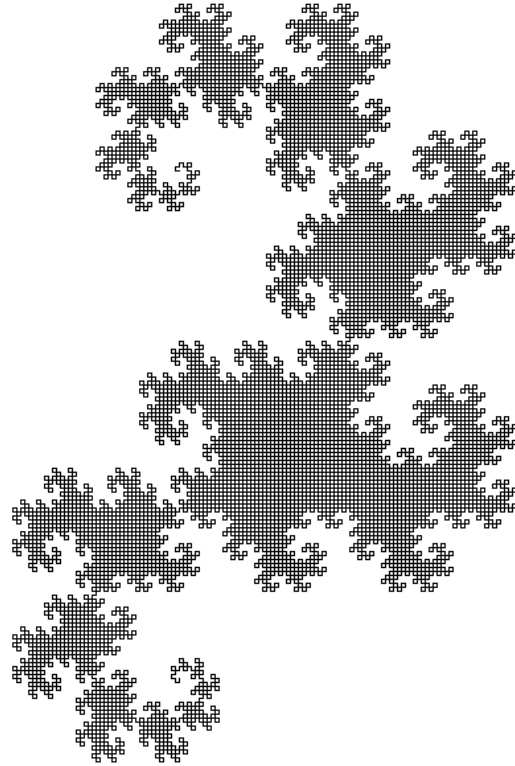


Рисунок 3.

ЗАДАЧИ.

Задача 1.

Возвести заданное вещественное число X в заданную натуральную степень N . В этой задаче совсем легко найти формулу: $X^N = X \cdot X^{N-1}$, но такой алгоритм потребует $N-1$ повторения процесса. Гораздо более эффективным будет следующий алгоритм:

$$X^N = \begin{cases} X, & \text{если } N = 1 \\ (X^{N/2})^2, & \text{если } N - \text{четно} \\ (X^{(N-1)/2})^2, & \text{если } N - \text{нечетно} \end{cases}$$

Уровень 1. Описать по шагам алгоритм «быстрого» возведения в степень. Продемонстрировать его работу при $X = 2$, $N = 11$.

Уровень 2. Описать рекурсивную функцию «быстрого» возведения в степень. Написать программу, которая возводит заданное вещественное число в натуральную степень.

Формат ввода:

Число X

Число N

Формат вывода:
Значение полученной степени

Пример.

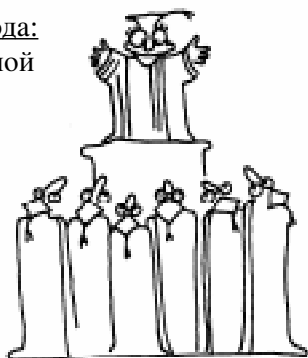
Ввод:

2

4

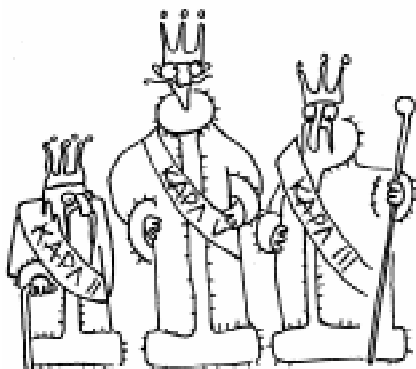
Вывод:

16



Задача 2.

Последовательность чисел 0, 1, 1, 2, 3, 5, ..., где каждое число, начиная с третьего, есть сумма двух предыдущих чисел последовательности, называется последовательностью чисел Фибоначчи.



При заданном N требуется найти первые N чисел Фибоначчи.

Уровень 1. Описать алгоритм, который для заданного N получит число Фибоначчи с номером N . Вычислить тринадцатое число Фибоначчи.

Уровень 2. Описать рекурсивную функцию, которая выдает значение k -ого числа Фибоначчи.

Примечание: Обратить особое внимание на неэффективность использования формулы $F(k)=F(k-1)+F(k-2)$, поскольку в вызове $F(k-1)$ содержится вызов $F(k-2)$ и т.д., что порождает многочисленные повторные вычисления одного и того же числа.

Формат ввода:

Число N

Формат вывода:

Значение N -ого числа Фибоначчи

Пример.

Ввод:

9

Вывод:

21

Задача 3.

Во входном потоке находится последовательность литер, которые могут быть только символами «0» или «1». Пробел - признак конца последовательности. Любая другая литера - ошибка. Нужно найти десятичное число, равное двоичному числу, заданному этой последовательностью.

Уровень 1. Построить по шагам алгоритм перевода двоичного числа в десятичное число. Найти десятичное представление для последовательностей: 100, 110001, 10000011.

Уровень 2. По заданной последовательности найти десятичное число, описав соответствующую рекурсивную функцию.

Формат ввода:

Последовательность нулей и единиц

Формат вывода:

Значение десятичного числа



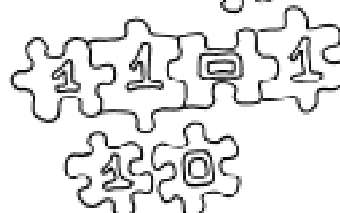
Пример.

Ввод:

10111

Вывод:

23



Задача 4.

Уровень 2. Во входном потоке находится последовательность литер, которые могут быть только цифрами. Пробел - признак конца последовательности. Любая другая литера - ошибка. «Собрать» десятичное число из цифр этой последовательности, описав рекурсивную функцию.

Формат ввода:

Последовательность литер, являющихся цифрами

Формат вывода:

Значение полученного числа

Пример.

Ввод:

4521

Вывод:

4521

В примере 1 мы подробно разобрали рекурсивный алгоритм создания рисунка из окружностей. Теперь можно рассмотреть более сложные, но в то же время и более красивые рисунки, например, «веточки» (рисунок 4 и рисунок 5) и «снежинки» (рисунок 6 и рисунок 7).

При создании таких рисунков «из реальной жизни» можно проявить настоящее творчество, например, нарисовать: – снежинки, различные по форме или падающие по всему экрану; – веточки разного цвета, разной пушистости и разного размера (в длину ветки, угол наклона и цвет вносятся случайная составляющая, рисунок 5); – а если на последнем уровне вложенности веточки рисовать желтые кружочки, то из голой зимней ветки можно получить пушистую весеннюю мимозу.

Задача 5.

Уровень 1. Описать по шагам, как нарисовать веточку, изображенную на рисунке 4.

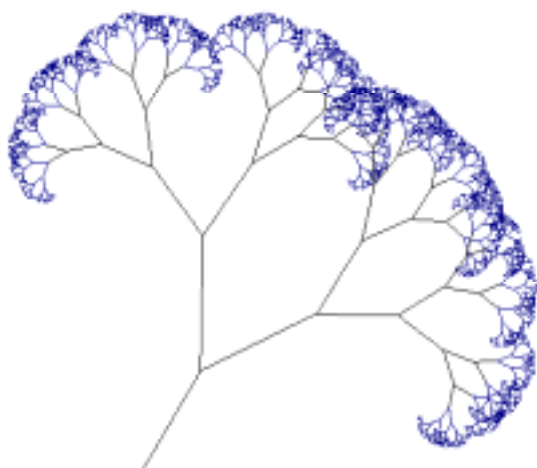


Рисунок 4.

Уровень 2. Написать программу, рисующую на экране дисплея веточку, подобную изображенной на рисунке 4.

Формат ввода:

Входных данных нет

Формат вывода:

Рисунок веточки

Задача 6.

Уровень 1. Описать по шагам, как нарисовать веточку, изображенную на рисунке 5.

Уровень 2. Написать программу с использованием случайных составляющих параметров, рисующую веточку, подобную изображенной на рисунке 5.

Формат ввода:

Входных данных нет

Формат вывода:

Рисунок веточки

Задача 7.

Уровень 1. Описать по шагам, как нарисовать снежинку, показанную на рисунке 6.

Уровень 2. Написать программу, рисующую снежинку, изображенную на рисунке 6.

Формат ввода:

Входных данных нет

Формат вывода:

Рисунок снежинки

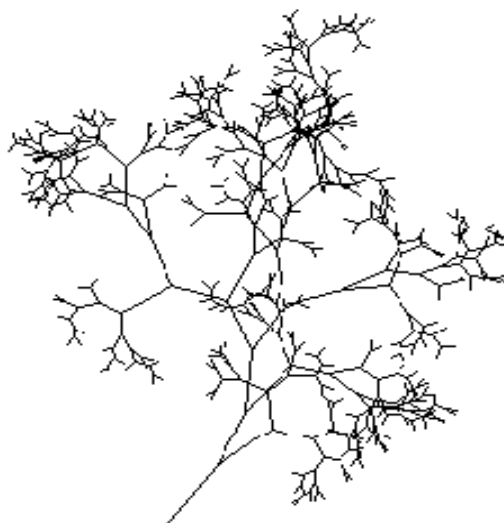


Рисунок 5.

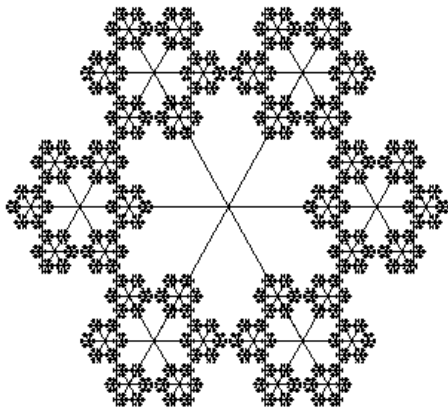


Рисунок 6.



Рисунок 7.

Задача 8.

Уровень 1. Описать по шагам, как нарисовать снежинку, показанную на рисунке 7.

Уровень 2. Написать программу, рисующую снежинку, изображенную на рисунке 7.

Формат ввода:

Входных данных нет

Формат вывода:

Рисунок снежинки

Указание. Из рисунка 8 видно, что третья ветвь составляет по длине примерно половину первой ветви. Уменьшается также толщина ветвей-сосудов.

Уровень 2. Написать программу, рисующую модель легкого, приведенную на рисунке 8.

Формат ввода:

Входных данных нет

Формат вывода:

Графическое изображение легкого

Задача 9.

Для следующей задачи рассмотрим пример из области биологии. С помощью рекурсии можно построить модель Бенуа Мандельброта человеческого легкого, а именно, иерархию трахей и бронхов.

Задача 10.

На рисунке 9 представлена салфетка Серпинского. Как она образуется? Рисуеться треугольник и в нем средние линии. В образованных при углах исходного треугольника новых треугольниках

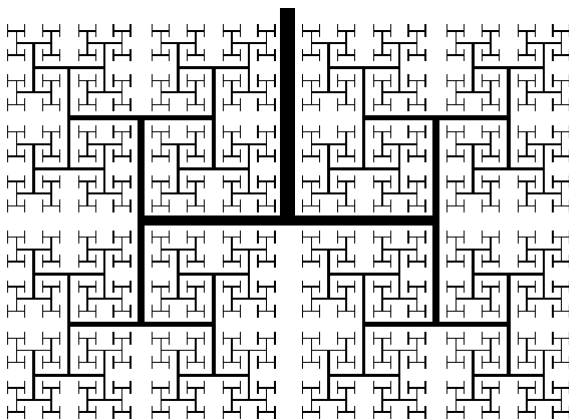


Рисунок 8.

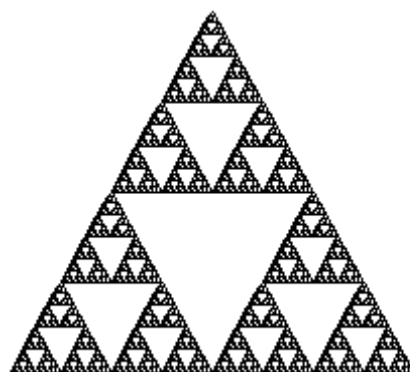


Рисунок 9.

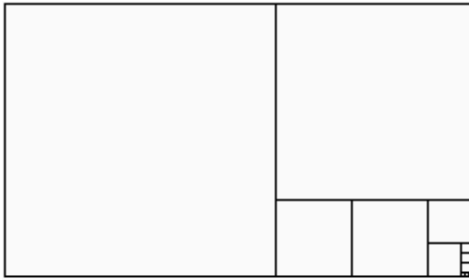


Рисунок 10.

опять рисуются средние линии и т.д. до заданного порядка вложенности. Интересно, что полученная фигура допускает другое (не рекурсивное) построение с помощью моделирования методом Монте-Карло (но об этом в другой раз).

Уровень 2. Написать программу, рисующую салфетку Серпинского, приведенную на рисунке 9.

Формат ввода:

Входных данных нет

Формат вывода:

Изображение салфетки Серпинского

Задача 11.

Уровень 2. Рассмотрим задачу о разрезании прямоугольника с натуральными длинами сторон a и b на квадраты максимальной площади (задача на использование алгоритма Евклида). Предлагается не просто решить эту задачу, но и сделать наглядное графическое сопровождение с помощью рекурсии (рисунок 10).

Указание. Параметрами рекурсивной процедуры являются координаты

вершин диагонали прямоугольника (x_1, y_1) и (x_2, y_2) . Как только стороны прямоугольника становятся равными, то есть $(x_2 - x_1 = y_2 - y_1)$, осуществляется выход из процедуры, если же нет, то вычисляется длина меньшей стороны k (это и будет сторона квадрата), и вдоль большей стороны отрезается (рисует) квадрат. Затем происходит обращение к рекурсивной процедуре с измененными параметрами, либо, вместо x_1 , берется $x_1 + k$ (если сторона по оси x больше), либо вместо y_1 берется $y_1 + k$ (если сторона по оси y больше).

Формат ввода:

Два целых числа через пробел

Формат вывода:

Изображение схемы деления

Далее приводится несколько задач повышенной сложности для программирования.

Задача 12.

Во входном потоке последовательность литер, заканчивающаяся пробелом. При помощи рекурсивной логической функции определить, является ли эта последовательность

правильным идентификатором. (Правильным идентификатором считается любая конечная последовательность букв или цифр, начинающаяся с буквы).

Формат ввода:

Последовательность букв и/или цифр

Формат вывода:

«Последовательность является правильным идентификатором» или «Последовательность не является правильным идентификатором»

Пример.

Ввод:

Max12

Вывод:

Последовательность является правильным идентификатором



Задача 13.

Во входном потоке последовательность литер, признаком конца которой является знак равенства. В самой последовательности могут встречаться только цифры и знаки «+». При этом вся последовательность представляет собой формулу сложения однозначных целых чисел. Описать рекурсивную функцию, которая найдет значение этой формулы или сообщит об ошибке.

Формат ввода:

Чередующаяся последовательность цифр и знаков «+», заканчивающаяся знаком «=»

Формат вывода:

Значение суммы

Пример.

Ввод:

1+3+6+2 =

Вывод:

12

Задача 14.

Во входном потоке последовательность литер, признаком конца которой является знак «=». В самой последовательности могут встречаться только цифры и знаки «+» или «-». При этом вся последовательность представляет собой формулу арифметической суммы однозначных целых чисел. Описать рекурсивную функцию, которая найдет значение этой формулы или сообщит об ошибке.

Формат ввода:

Чередующаяся последовательность цифр и знаков «+» и «-», заканчивающаяся знаком «=»

Формат вывода:

Значение суммы

Пример.

Ввод:

2+5-7+1+9 =

Вывод:

10

Задача 15.

Во входном потоке последовательность литер, признаком конца которой является «=». В самой последовательности могут встречаться только цифры и знаки «+» или «-». При этом вся последовательность представляет собой формулу арифметической суммы произвольных целых чисел. Описать рекурсивную процедуру или функцию, которая найдет значение этой формулы или сообщит об ошибке.

Формат ввода:

Последовательность цифр и знаков «+» или «-», заканчивающаяся знаком «=»

Формат вывода:

Значение суммы

Пример.

Ввод:

45+6-35+4-17 =

Вывод:

3

На дискетке, приложенной к журналу, вы найдете файл Nano1.exe, который содержит демонстрационно-игровую программу для решения задачи «Ханойские башни».

*Павлова Марианна Владимировна,
старший преподаватель кафедры
информатики СПбГУ.*

*Паньгина Нина Николаевна,
преподаватель ОИ и ВТ лицея № 8
г. Сосновый Бор.*

НАШИ АВТОРЫ