[1. Лабораторные работы по технологии программирования 2](#_Toc81028728)

[1.1. Лабораторная работа № 1 Списки. Стеки. Очереди. (30) 3](#_Toc81028729)

[1.2. Лабораторная работа №2 Разреженные матрицы (31) 6](#_Toc81028730)

[1.3. Лабораторная работа №3 Рекурсия. Бинарные деревья. (28) 8](#_Toc81028731)

[1.4. Лабораторная работа №4 Рекурсия. N-древья. (22) 10](#_Toc81028732)

[1.5. Лабораторная работа №5 Методы поиска (хэширование, B-дерево и т.д.) 12](#_Toc81028733)

[1.6. Лабораторная работа №6 Методы сортировки(20) 13](#_Toc81028734)

[1.7. Дополнительная лабораторная работа №1 Рекурсия и головоломки. (49) 16](#_Toc81028735)

[1.8. Дополнительная Лабораторная работа №2 Рекурсивные графические алгоритмы (Фракталы) 46](#_Toc81028736)

## Лабораторная работа № 1 Списки. Стеки. Очереди. (30)

**Задача 1.** Вводится последовательность, состоящая из N пар символов (ai,bi). Каждая пара определяет порядок предшествования символов, например, пара (b,с) означает, что символ "b" предшествует символу "с". Из порядка (b,с) и (с,a) следует порядок (b,a). Необходимо определить, является ли введенная последовательность:

а) полной, т.е. все использованные для формирования пар символы (выбросив повторяющиеся) можно выстроить в цепочку (A1,A2,...,As) в порядке предшествования;

б) противоречивой, т.е. для некоторых символов x,y можно получить одновременно порядок как (x,y) так и (y,x);

--------------------------------------------------------------------------------

**Задача 2.** Вокруг считающего стоит N человек, из которых выделен первый, а остальные занумерованы по часовой стрелке числами от 2 до N. Считающий, начиная с кого-то, ведет счет до M. Человек, на котором остановился счет, выходит из круга. Счет продолжается со следующего человека и так до тех пор, пока не останется один человек.

Определить

a) номер оставшегося человека, если известно M и то, что счет начинался с первого человека;

b) номер человека c которого начинался счет, если известно M и номер оставшегося человека L.

--------------------------------------------------------------------------------

**Задача 3.** Дана конечная последовательность, состоящая из левых и правых скобок pазличных заданных типов( «(» «{» «[»«)» «}» «]»). Определить, можно ли добавить в нее цифры и знаки арифметических действий так, чтобы получилось правильное арифметическое выражение.

--------------------------------------------------------------------------------

**Задача 4**. В списке А размера N необходимо каждый элемент заменить на ближайший следующий за ним элемент, который больше его. Если такого элемента нет, то заменить его на ноль.

ПРИМЕР А=1 3 2 5 3 4

ОТВЕТ А=3 5 5 0 4 0

--------------------------------------------------------------------------------

**Задача 5.** Одинокий король долго ходил по бесконечной шахматной доске. Известна последовательность из N его ходов (вверх, вниз, влево, вправо, вверх-влево и т.п.).Написать программу, определяющую побывал ли король дважды на одном и том же поле за минимально возможное при заданном N число вычислений.

--------------------------------------------------------------------------------

**Задача 6** По кругу расположено N монет гербами вверх и M монет гербами вниз. Обходя круг по ходу часовой стрелки, переворачивает каждую S -тую монету. В первый раз счет начинается с герба. В каком порядке надо расставить монеты, чтобы после K ходов стало L монет, лежащих гербами вверх.

--------------------------------------------------------------------------------

**Задача 7.**

N серых и M белых мышей сидят по кругу. Кошка ходит по кругу по часовой стрелке и съедает каждую S -тую мышку. В первый раз счет начинается с серой мышки. Составить алгоритм определяющий порядок в котором сидели мышки, если через некоторое время осталось K серых и L белых мышей.

--------------------------------------------------------------------------------

**Задача 8.**

Имеется n черных и белых карточек, сложенных в стопку. Карточки раскладываются на стол в одну линию следующим образом: первая кладется на стол, вторая под низ стопки, третья- на стол, четвертая - под низ стопки и т.д., пока все карточки не будут выложены на стол. Каким должно быть исходное расположение карточек в стопке, чтобы разложенные на столе карточки чередовались по цвету: белая, черная, белая, черная и т.д.

--------------------------------------------------------------------------------

**Задача 9.**

1. Дана величина *a* строкового типа из четного количества символов. Получить и напечатать величину *b*, состоящую из символов первой половины величины *a*, записанных в обратном порядке, после которых идут символы второй половины величины *a*, также записанные в обратном порядке. Например, при *а* = “привет” *b* должно быть равно “ипртев”.

--------------------------------------------------------------------------------

**Задача 10.**

Написать процедуру, которая меняла бы в односвязном списке крайние элементы.

--------------------------------------------------------------------------------

**Задача11.**

Даны четыре стека. Два первых стека пустые, а в двух других находятся натуральные числа. В первый пустой стек поместить четные числа в порядке не убывания, взятые из непустых стеков, во второй пустой стек поместить нечетные числа в порядке не возрастания, так же взятые их непустых стеков. Предполагается, что множества четных чисел и нечетных чисел непустые.

--------------------------------------------------------------------------------

**Задача 12.**

Используя список, удалить из текста программы комментарии и поместить их в отдельный текстовый файл.

--------------------------------------------------------------------------------

**Задача 13.**

Дан список вещественных чисел. Все его элементы, не равные нулю, переписать, сохраняя порядок, в начало списка, а нулевые элементы в конец списка. Новый список заводить нельзя.

--------------------------------------------------------------------------------

**Задача 14.**

Используя очередь, отредактировать текст, оставляя один пробел в каждой серии пробелов.

--------------------------------------------------------------------------------

**Задача 15.**

Используя список, удалить их текста заданный набор букв.

-----------------------------------------------------------------------------------------------------------------

**Задача 16.**

Даны две непустые очереди, которые содержат одинаковое количество элементов. Объединить очереди в одну, в которой элементы исходных очередей чередуются.

-----------------------------------------------------------------------------------------------------------------

**Задача 17.**

Даны две непустые очереди. Элементы каждой из очередей упорядочены по возрастанию. Объединить очереди в одну с сохранением упорядоченности элементов.

-----------------------------------------------------------------------------------------------------------------

**Задача 18.**

Пусть имеется набор действительных чисел и некоторое число C. Используя очередь, напечатать сначала все элементы, меньшие числа C, а затем большие числа C (если в наборе встретиться С, то его выводить С раз).

-----------------------------------------------------------------------------------------------------------------

**Задача 19.**

Написать функцию, которая по произвольному указателю на один из элементов двусвязного списка подсчитывает количество элементов в этом списке

-----------------------------------------------------------------------------------------------------------------

**Задача 20.**

Имеется список действительных чисел. Продублировать в нем все положительные числа

-----------------------------------------------------------------------------------------------------------------

**Задача 21.**

Определить, расположены ли элементы в двусвязном списке симметричным образом

-----------------------------------------------------------------------------------------------------------------

**Задача 22.**

Осуществить циклический сдвиг элементов двусвязного списка на k позиций вправо

-----------------------------------------------------------------------------------------------------------------

**Задача 23.**

Осуществить циклический сдвиг элементов односвязного списка на k позиций влево (не по направлению движения)

-----------------------------------------------------------------------------------------------------------------

**Задача 24.**. Объединить n списков в один список

-----------------------------------------------------------------------------------------------------------------

**Задача 25.** Поменять местами два элемента в списке

-----------------------------------------------------------------------------------------------------------------

**Задача 26**. Добавить новый элемент в список перед заданным элементом

-----------------------------------------------------------------------------------------------------------------

**Задача 27**. Удалить из списка неупорядоченные подсписки

-----------------------------------------------------------------------------------------------------------------

**Задача 28**. Сделать определённый элемент в списке первым

-----------------------------------------------------------------------------------------------------------------

**Задача 29.** Реализовать удаление элементов из очереди с приоритетом. Очередь с приоритетом – первым включается – с высшим приоритетом исключается

-----------------------------------------------------------------------------------------------------------------

**Задача 30**. Реализовать функции работы со стеком (удаление, вставка).

--------------------------------------------------------------------------------------------------------------

## Лабораторная работа №2 Разреженные матрицы (31)

**CS - координатный формат**

**CRS - Разреженный строчный формат**

**CCS - Разреженный столбцовый формат**

**SSS - Разреженный структурно-симметричный** **формат**

**Ленточный строчный формат**

**Ленточный столбцовый формат**

**Диагональный формат**

**Задача 1.** Даны две разреженные матрицы **CS**. Перемножить их и результат занести в разреженную матрицу **CRS**.

**Задача 2.** Даны две разреженные матрицы **(Ленточный строчный формат)**. Сложить их и результат занести в разреженную матрицу **CCS**.

**Задача 3.** Дана разреженная матрица **CRS**. Найти её определитель.

**Задача 4.** Дана разреженная матрица (**CS**). Осуществить циклический сдвиг в матрице. Сдвинуть всю матрицу. В первой строке первый элемент переносится на второе место, второй элемент на третье и т.д. Последний элемент в первой строке становиться первым элементом во второй строке. Последний элемент в последней строке переноситься на первую строку на первое место.

**Задача 5.** Дана разреженная матрица (**CS**) и число b. Матрица просматривается слева на право, и сверху вниз. На места ненулевых элементов матрицы вначале поместить все её ненулевые элементы большие b, а затем ненулевые элементы меньшие b. Элементы не сортировать.

**Задача 6.** Даны две разреженные матрицы **CS**. Перемножить их и результат занести в разреженную матрицу **CCS**.

**Задача 7.** Даны две разреженные матрицы **CS**. Сложить их и результат занести в разреженную матрицу **CRS**.

**Задача 8.** Дана разреженная матрица (**CCS**). Найти сумму её элементов aij , у которых сумма (i+j) является чётной.

**Задача 9.** Дана разреженная матрицы (**CCS**). Переставить строки в матрице по возрастанию сумм элементов в этих строках.

**Задача 10.** Дана разреженная матрицa (CCS). Зеркальное отображение относительно диагонали, проходящей с левого нижнего угла к правому верхнему углу.

**Задача 11.** Дана разреженная матрица (**CCS**) и число b. Матрица просматривается слева на право, и сверху вниз. На места ненулевых элементов матрицы вначале поместить все её ненулевые элементы большие b, а затем ненулевые элементы меньшие b. Элементы не сортировать.

**Задача 12.** Дана разреженная матрица **SSS**. Найти её определитель.

**Задача 13.** Дана разреженная матрицы (CS). Осуществить циклический сдвиг в матрице каждого столбца на n разрядов.

**Задача 14.** Дана разреженная матрицы (**CCS**). Переставить столбцы в матрице по возрастанию сумм элементов в этих столбцах.

**Задача 15.** Дана разреженная матрица (**CRS**). Осуществить циклический сдвиг в матрице. Сдвинуть всю матрицу. В первой строке первый элемент переносится на второе место, второй элемент на третье и т.д. Последний элемент в первой строке становиться первым элементом во второй строке. Последний элемент в последней строке переноситься на первую строку на первое место.

**Задача 16.** Даны две разреженные матрицы **(Диагональный формат)**. Сложить их и результат занести в разреженную матрицу **CCS**.

**Задача 17.** Даны две разреженные матрицы **(Ленточный строчный формат)**. Перемножить их и результат занести в разреженную матрицу **CCS**.

**Задача 18.** Дана разреженная матрицы (**CCS**). Осуществить циклический сдвиг в матрице каждого столбца на n разрядов.

**Задача 19.** Дана разреженная матрица (**SSS**) и число b. Матрица просматривается слева на право, и сверху вниз. На места ненулевых элементов матрицы вначале поместить все её ненулевые элементы большие b, а затем ненулевые элементы меньшие b. Элементы не сортировать.

**Задача 20.** Дана разреженная матрица (**Диагональный формат**). Циклически сдвинуть все диагонали a,b,c, и т.д..



**Задача 21.** Дана разреженная матрицa (**Ленточный столбцовый формат**). Зеркальное отображение относительно диагонали, проходящей с левого нижнего угла к правому верхнему углу.

**Задача 22.** Дана разреженная квадратная матрица (**CS**). Повернуть матрицу на 90 градусов.

**Задача 23.** Дана разреженная матрица (**Ленточный столбцовый формат**). Найти её определитель.

**Задача 24.** Дана разреженная матрица (**Диагональный формат**). Найти сумму её элементов aij , у которых сумма (i+j) является чётной.

**Задача 25.** Дана разреженная матрицы (**CRS**). Осуществить циклический сдвиг в матрице каждого столбца на n разрядов.

**Задача 26.** Дана разреженная квадратная матрица (**CRS**). Повернуть матрицу на 180 градусов.

**Задача 27.** Даны две разреженные матрицы **SSS**. Перемножить их и результат занести в разреженную матрицу **CRS**.

**Задача 28.** Дана разреженная квадратная матрица (**CCS**). Повернуть матрицу на 270 градусов.

**Задача 29.** Дана разреженная матрица (**CCS**). Осуществить циклический сдвиг в матрице. Сдвинуть всю матрицу. В первой строке первый элемент переносится на второе место, второй элемент на третье и т.д. Последний элемент в первой строке становиться первым элементом во второй строке. Последний элемент в последней строке переноситься на первую строку на первое место.

**Задача 30.** Дана разреженная матрица (**SSS**). Найти сумму её элементов.

**Задача 31.** Дана разреженная матрица (**SSS**). Найти количество её различных элементов и вывести их на экран.

## Лабораторная работа №3 Рекурсия. Бинарные деревья. (28)

**Задача 1.** Дано бинарное дерево. Найти поддеревья с максимальным и минимальным соотношением (высота / число листьев).

**Задача 2.** Дано бинарное дерево. Найти самый длинный от корня путь, проходящий только по вершинам с нечётными номерами.

**Задача 3.** Дано бинарное дерево. Заданы веса вершин. Найти поддеревья с min и max отношением (сумма весов / число вершин). Вывести эти поддеревья.

**Задача 4.** Дано бинарное дерево. Найти в дереве длиннейший путь (пути), вдоль которого номера вершин упорядочены по возрастанию.

**Задача 5.** Дано бинарное дерево. Каждую вершину с чётным номером поменять местами с сыном, имеющим чётный номер.

**Задача 6.** Дано бинарное дерево. Найти все вершины с одинаковыми номерами.

**Задача 7.** Дано бинарное дерево. Найти все вершины равноудалённые от корня и от ближайшего своего листа.

**Задача 8.** Дано бинарное дерево. Найти ветви с мах числом ветвлений.

**Задача 9.** Дано бинарное дерево. Выполнить вращение для самого разбалансированого поддерева.

**Задача 10.** Дано бинарное дерево. Определить какие поддеревья являются пирамидами.

**Задача 11.** Дано бинарное дерево. Найти в дереве самые длинные пути без ветвлений.

**Задача 12.** Дано бинарное дерево. Найти в дереве самое высокое (низкое) поддерево, имеющее заданное число листьев.

**Задача 13.** Дано бинарное дерево. Найти в дереве самое широкое (узкое) поддерево имеющее заданную высоту.

**Задача 14.** Дано бинарное дерево. Найти поддерево не включающее ни одну из заданных вершин.

**Задача 15.** Дано бинарное дерево. Найти все поддеревья, структура которых совпадает с заданной.

**Задача 16.** Дано бинарное дерево. Найти все поддеревья, листья которых находятся в заданном диапазоне высот от корня поддерева.

**Задача 17.** Дано бинарное дерево. Найти все поддеревья, вершины которых имеют номера: нечётные, если вершина находится на нечётном расстоянии от корня, и чётные в противном случае.

**Задача 18.** Дано бинарное дерево. Найти самый длинный в дереве путь, проходящий только по вершинам с нечётными номерами.

**Задача 19.** Дано бинарное дерево. Найти все поддеревья, структура которых совпадает с заданной.

**Задача 20.** Дано бинарное дерево. Вывести сумму значений всех листьев данного дерева

**Задача 21.** Дано бинарное дерево. Вывести количество листьев дерева, являющихся правыми дочерними вершинами.

**Задача 22.** Дано бинарное дерево. Для каждого из уровней данного дерева, начиная с нулевого, вывести количество вершин, находящихся на этом уровне.

**Задача 23.** Дано бинарное дерево. Вывести максимальное из значений его внутренних вершин (т. е. вершин, не являющихся листьями)

**Задача 24.** Дано бинарное дерево. Дано бинарное дерево. Для каждой вершины дерева, имеющей две дочерние вершины, поменять местами значения дочерних вершин (т. е. значения их полей Data).

**Задача 25.** Дано бинарное дерево. Ко всем вершинам дерева, которые являются листьями, добавить по одной дочерней вершине-листу; при этом к исходной вершине с нечетным значением добавляется левая дочерняя вершина, а к вершине с четным значением — правая. Значение каждой добавленной вершины положить равным значению ее родительской вершины.

**Задача 26.** Дано бинарное дерево. Кроме того все вершины имеют цвет. Всего 3 цвета (белый, черный и желтый). Все вершины с нечетными номерами перекрасить желтый 🡪черный🡪белый🡪желтый.

**Задача 27.** Дано N-дерево. Каждая вершина имеет некоторое значение. В каждую вершину вставить сумму значений всех его предков.

**Задача 28.** Дано бинарное дерево. Кроме того все вершины имеют цвет. Всего 3 цвета (белый, черный и желтый). Из множества листьев имеющих желтый цвет удалить самые далекие от корня, если таковых нет, то из множества листьев имеющих черных или белый цвет удалить самые близкие от корня.

## Лабораторная работа №4 Рекурсия. N-древья. (22)

**Задача 1.** Дано N-дерево. Удалить самые далекие от корня листья в дереве.

**Задача 2.** Дано N-дерево. Удалить самые близкие от корня листья в дереве.

**Задача 3.** Дано N-дерево. Найти поддеревья с максимальным и минимальным соотношением (высота / число листьев).

**Задача 4.** Дано N-дерево. Найти самый длинный от корня путь, проходящий только по вершинам с нечётными номерами.

**Задача 5.** Дано N-дерево. Заданы веса вершин. Найти поддеревья с min и max отношением (сумма весов / число вершин). Вывести эти поддеревья.

**Задача 6.** Дано N-дерево. Найти в дереве длиннейший путь (пути), вдоль которого номера вершин упорядочены по возрастанию.

**Задача 7.** Дано N-дерево. Найти все вершины с одинаковыми номерами.

**Задача 8.** Дано N-дерево. Найти все вершины равноудалённые от корня и от ближайшего своего листа.

**Задача 9.** Дано N-дерево. Найти ветви с мах числом ветвлений.

**Задача 10.** Дано N-дерево. Найти в дереве самые длинные пути без ветвлений.

**Задача 11.** Дано N-дерево. Найти в дереве самое высокое (низкое) поддерево, имеющее заданное число листьев.

**Задача 12.** Дано N-дерево. Найти в дереве самое широкое (узкое) поддерево имеющее заданную высоту.

**Задача 13.** Дано N-дерево. Найти поддерево не включающее ни одну из заданных вершин.

**Задача 14.** Дано N-дерево. Найти все поддеревья, структура которых совпадает с заданной.

**Задача 15.** Дано N-дерево. Найти все поддеревья, листья которых находятся в заданном диапазоне высот от корня поддерева.

**Задача 16.** Дано N-дерево. Найти все поддеревья, вершины которых имеют номера: нечётные, если вершина находится на нечётном расстоянии от корня, и чётные в противном случае.

**Задача 17.** Дано N-дерево. Найти самый длинный в дереве путь, проходящий только по вершинам с нечётными номерами.

**Задача 18.** Дано N-дерево. Вывести сумму значений всех листьев данного дерева

**Задача 19.** Дано N-дерево. Для каждого из уровней данного дерева, начиная с нулевого, вывести количество вершин, находящихся на этом уровне.

**Задача 20.** Дано N-дерево. Вывести максимальное из значений его внутренних вершин (т. е. вершин, не являющихся листьями)

**Задача 21.** Дано N-дерево. Кроме того все вершины имеют цвет. Всего 3 цвета (белый, черный и желтый). Все вершины с нечетными номерами перекрасить желтый 🡪черный🡪белый🡪желтый.

**Задача 22.** Дано N-дерево. Каждая вершина имеет некоторое значение. В каждую вершину вставить сумму значений всех его предков.

## Лабораторная работа №5 Методы поиска (хэширование, B-дерево и т.д.)

Реализовать алгоритмы поиска, вставки, удаления элементов таблицы и распечатки таблицы и метод ре-хэширования таблицы при увеличении размера данных.

|  |  |
| --- | --- |
| **№ Варианта** | **Ключ** |
| 1 | СНИЛС |
| 2 | Номер телефона |
| 3 | Адрес Компании(Страна, Город, Улица, Дом). |
| 4 | Фамилия Имя Отчество |

|  |  |
| --- | --- |
| **№ Варианта** | **Хэш-функция** |
| 1 | функция деления |
| 2 | функция умножения |
| 3 | функция «середины квадрата» |
| 4 | функция свертки (слияния) |
| 5 | функции работы со строковыми ключами |

|  |  |
| --- | --- |
| **№ Варианта** | **Метод разрешения коллизий** |
| 1 | Метод цепочек |
| 2 | Квадратичное опробование |
| 3 | Использование бинарных деревьев |

|  |  |  |  |
| --- | --- | --- | --- |
| **№ Варианта** | **Ключ** | **Хэш-функция** | **Метод разрешения коллизий** |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 1 |
| 5 | 1 | 5 | 2 |
| 6 | 2 | 1 | 3 |
| 7 | 3 | 2 | 1 |
| 8 | 4 | 3 | 2 |
| 9 | 1 | 4 | 3 |
| 10 | 2 | 5 | 1 |
| 11 | 3 | 1 | 2 |
| 12 | 4 | 2 | 3 |
| 13 | 1 | 3 | 1 |
| 14 | 2 | 4 | 2 |
| 15 | 3 | 5 | 3 |
| 16 | 4 | 1 | 1 |
| 17 | 1 | 2 | 2 |
| 18 | 2 | 3 | 3 |
| 19 | 3 | 4 | 1 |
| 20 | 4 | 5 | 2 |
| 21 | 1 | 1 | 3 |
| 22 | 2 | 2 | 1 |
| 23 | 3 | 3 | 2 |
| 24 | 4 | 4 | 3 |
| 25 | 1 | 5 | 1 |
| 26 | 2 | 1 | 2 |
| 27 | 3 | 2 | 3 |
| 28 | 4 | 3 | 1 |
| 29 | 1 | 4 | 2 |
| 30 | 2 | 5 | 3 |

## Лабораторная работа №6 Методы сортировки(20)

**Задача 1.** (Лексикографическая сортировка)

Имеется N камней веса А1,А2,...,АN.

Необходимо разбить их на две кучи таким образом, чтобы веса куч отличались не более чем в 2 раза. Если этого сделать нельзя, то указать это.

**Задача 2.** Условие задачи 1, только веса куч отличаются не более, чем в 1,5 раза.

**Задача 3.** Даны две целочисленных таблицы А[1:10] и В[1:15]. Разработать алгоритм и написать программу, которая проверяет, являются ли эти таблицы похожими. Две таблицы называются похожими, если совпадают множества чисел, встречающихся в этих таблицах.

**Задача 4.** Задается словарь. Найти в нем все анаграммы (слова, составленные из одних и тех же букв).

**Задача 5.** (Лексикографическая сортировка) Имеется 2\*N чисел. Известно, что их можно разбить на пары таким образом, что произведения чисел в парах равны. Сделать разбиение, если числа

а) натуральные;

б) целые.

**Задача 6.** Имеются числа А1,А2,...,АN и B1,B2,...,BN. Составить из них N пар (Аi, Bj) таким образом, чтобы сумма произведений пар была максимальна (минимальна). Каждое Ai и Bj в парах встречаются ровно по одному разу.

**Задача 7.** В музее регистрируется в течение дня время прихода и ухода каждого посетителя. Таким образом за день получены N пар значений, где первое значение в паре показывает время прихода посетителя и второе значения - время его ухода. Найти промежуток времени, в течение которого в музее одновременно находилось максимальное число посетителей.

**Задача 8.** Пусть A - множество из N натуральных чисел. Ваша программа должна определить, существует ли по крайней мере одно подмножество B множества A, имеющие следующее свойство (\*) для любых X,Y,Z из B, X<>Y<>Z<>X,

X+Y+Z <= {t: t из B\{X,Y,Z}},

тут B\{X,Y,Z} означает «множество B без элементов X,Y и Z».

В случае положительного ответа программа должна найти подмножество B, удовлетворяющее условию (\*) и состоящее из максимально возможного числа элементов.

**Задача 9.** Даны К целых чисел А(1),...,А(К). Вычислить

а) наибольшее,

b) наименьшее,

c) наиболее близкое к нулю.

**Задача 10.** Есть два отсортированных в порядке неубывания массива A[1,N] и B[1,M]. Получить отсортированный по неубыванию массив C[1,N+M], состоящий из элементов массивов A и B ("слить" вместе массивы A и B).

**Задача 11.** Дан массив X[1..N]. Необходимо циклически сдвинуть его на k элементов вправо (т.е. элемент X[i] после сдвига должен стоять на месте X[i+k]; тут мы считаем что за X[N] следует X[1]). Разрешается использовать только несколько дополнительных слов памяти (Дополнительного массива заводить нельзя!).

**Задача 12.** Некоторое число содержится в каждом из трех целочисленных неубывающих массивов x[1] <= ... <= x[p], y[1] <= ... <= y[q], z[1] <= ... <= z[r]. Найти одно из таких чисел. Число действий должно быть порядка p + q + r.

**Задача 13.** Пусть слово - это последовательность от 1 до 8 символов, не включающая пробелов.

Вводится n слов A1,...,An. Можно ли их переупорядочить так, чтобы получилась "цепочка", т.е. для каждого слова Aj его первая буква должна совпадать с последней буквой предыдущего слова, а последняя буква в Aj - с первой буквой последующего слова; соответственно последняя буква последнего слова должна совпадать с первой буквой первого слова. В цепочку входят все n слов без повторений.

Дать ответ в виде "Можно"\"Нельзя".

Если такое упорядочение возможно, то вывести какую-нибудь цепочку слов. Слова при выводе разделяются пробелами.

**Задача 14.** Реализовать «быструю» сортировку

**Задача 15.** Реализовать внешнюю сортировку

**Задача 16.** Реализовать лексикографическую сортировку

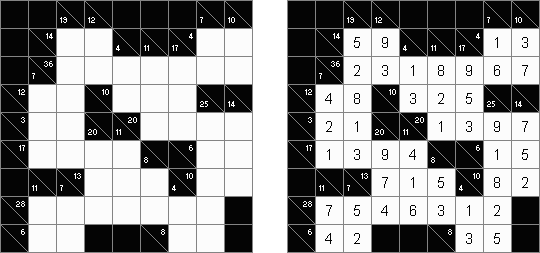
**Задача 17.** Реализовать пирамидальную сортировку

**Задача 18.** (пирамидальная сортировка) Отсортировать последовать так, чтобы в вначале шли отсортированные чётные элементы, а затем отсортированные нечётные.

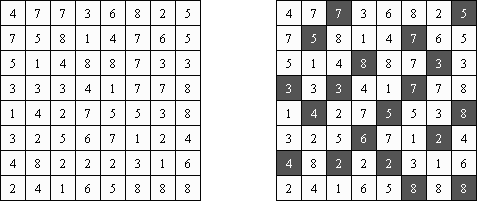
## Дополнительная лабораторная работа №1 Рекурсия и головоломки. (49)

1. **Какуро**
2. **Хитори**
3. **Китайская стена**
4. **Филиппинский кроссворд**
5. **Японская мозаика**
6. **Морской бой**
7. **Мосты**
8. **Ожерелье**
9. **Фонари**
10. **Филломино**
11. **Неравенство**
12. **Куромасу**
13. **Нурикабе**
14. **Лагерь**
15. **Кен-кен**
16. **Хидато**
17. **Нумератор**
18. **Квадраты**
19. **Галактики**
20. **Небоскрёбы**
21. **Волновой эффект**
22. **Гранд тур**
23. **Первые встречные**
24. **Облако**
25. **Бумеранг**
26. **Сапёр**
27. **Сапёр + Морской бой**
28. **Комнаты**
29. **Числобус**
30. **Сто**
31. **Стрелки**
32. **Матракс**
33. **Стрейтс**
34. **Линейщик**
35. **Двоичный код**
36. **Стены**
37. **Домино-пасьянс**
38. **Лоскутки**
39. **Особняки**
40. **Рекуто**
41. **Соседи**
42. **Роза ветров**
43. **Шака-шака**
44. **Какурасу**
45. **Мочикоро**
46. **Двери**
47. **Маяки**
48. **Тапа**
49. **Четвёртый лишний**

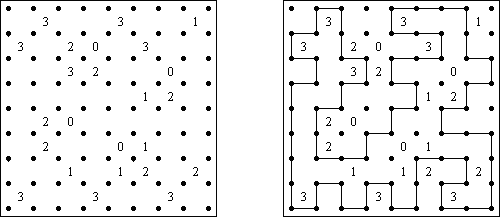
**Какуро** (*"Kakuro"*) - это числовая головоломка, математический эквивалент кроссворда. Необходимо вставить в клетки цифры от 1 до 9, причем некоторые клетки неактивны (такие клетки помечены черным цветом). В клетках с заданными числовыми значениями правое верхнее значение означает сумму цифр в ряду, а значение снизу слева равно сумме цифр столбца ниже клетки. Например, число 6 можно представить как сумму 1 и 5, 2 и 4; одинаковые цифры (3 и 3) использовать запрещено.



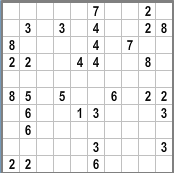
**Хитори** (*"Hitori"*) - это логическая головоломка. Необходимо вычеркивать повторяющиеся числа, чтобы ни одна из них не встречалась в любой строке или столбце чаще одного раза. Зачеркнутые клетки могут касаться друг друга только углами, но никак не сторонами. Незачеркнутые клетки должны составлять непрерывное "белое" пространство, то есть ни одна из них не должна быть изолирована от других таких же.

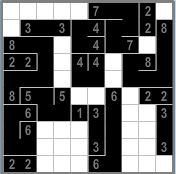


**Китайская стена** (*"Slitherlink"*, *"Fences"*, *"Loop the Loop"*, *"Dotty Dilemma"*, *"Sli-Lin"*; еще одно название - **Скользящие линии**) - это логическая головоломка, напоминающая лабиринт. Необходимо соединить точки вертикальными и горизонтальными прямыми отрезками так, чтобы получилась единая замкнутая цепь, не пересекающая саму себя. Каждая цифра указывает, сколько отрезков должно расположиться вокруг нее по периметру. Если в ячейке нет цифры, то вокруг нее может быть любое количество линий.

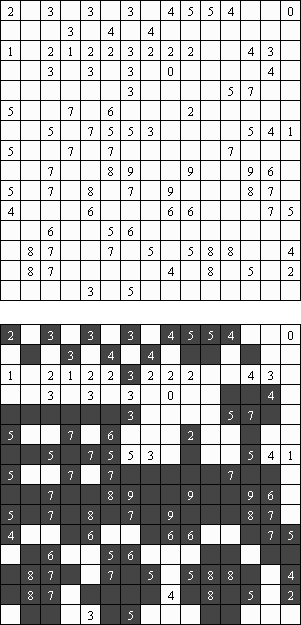


**Филиппинский кроссворд** (*"Link-a-Pix", "Paint by Pairs"*) - это головоломка с числами. Все числа, расположенные в сетке, кроме единицы, имеют свою пару. Необходимо найти каждую пару чисел и соединить их линиями. Количество клеток в ней должно равняться числам на ее концах. Линии, соединяющие пары, могут преломляться и идти в горизонтальном или вертикальном направлениях (но не по диагоналям). Линии не могут пересекаться друг с другом или проходить через одни и те же клетки.





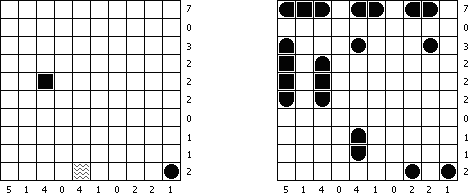
**Японская мозаика** (*"Fill-a-Pix"*, *"Mosaik"*, *"Nurie Puzzle"*, *"Nampre Puzzle"*) - это логическая головоломка, в которой с помощью цифр зашифрована картинка. Каждое из чисел означает, сколько примыкающих клеток (считая ту, что с цифрой) должно быть закрашено. Например, если в клетке стоит ноль, то ни эта клетка, ни соседствующие с ней закрашены не будут.



**Морской бой** (*"Battleships"*, *"Solitaire Battleships"*, *"Battleship Solitaire"*) - это головоломка, правила которой похожи на известную игру. Необходимо расположить "корабли" так, чтобы они не соприкасались даже углами. Цифры сбоку и снизу означают, сколько всего фрагментов кораблей попало в эту строку или столбец.

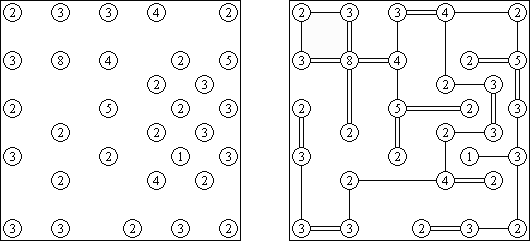


Стандартные размеры поля - **10 x 10**. Имеются один корабль, занимающий 4 клетки, два корабля, занимающие 3 клетки, три корабля из 2 клеток и четыре корабль размером в 1 клетку. Иногда некоторые из фрагментов кораблей уже открыты.

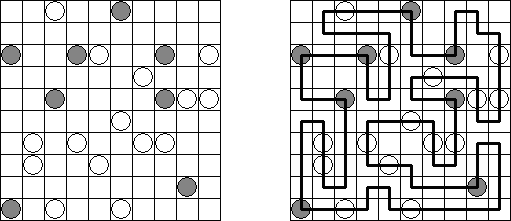


**Мосты** (*"Hashiwokakero"*, *"Hashi"*, *"Bridges"*, *"Chopsticks"*, *"Ai-Ki-Ai"*) - головоломка, в которой необходимо соединить кружки с цифрами ("острова") прямыми линиями ("мостами"). Цифра на острове показывает, сколько мостов должно быть к нему проложено. Изолированных островов быть не должно; линии должны быть проведены таким образом, чтобы с любого острова по построенным мостам можно было попасть на любой остров.

Между двумя островами разрешается строить не больше двух мостов. Линии могут проходить по горизонтали и вертикали, но не по диагонали. Они не должны преломляться, пересекаться или проходить сквозь острова.

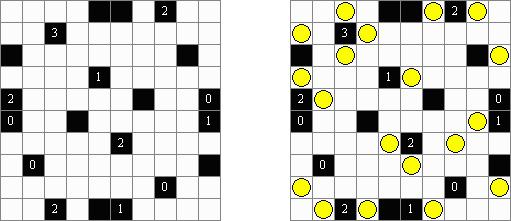


**Ожерелье** (*"Masyu"*, *"Shiroshinju Kuroshinju"*, *"White pearls and black pearls"*) - это логическая головоломка, в которой необходимо соединить белые и черные круги вертикальными и горизонтальными прямыми отрезками так, чтобы получилась единая замкнутая линия, не пересекающая саму себя. Через белые круги линия проходит прямо, но должна повернуть в предыдущей или следующей клетке (или в обеих этих клетках). Когда линия пересекает черный круг, она должна повернуть на 90 градусов, при этом в предыдущей и следующей клетке повороты запрещены.

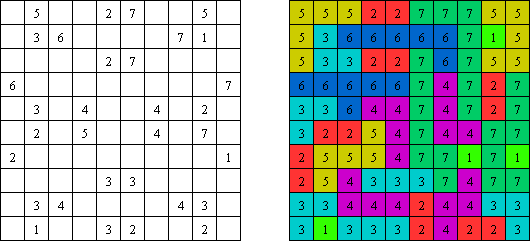


**Фонари** (*"Light Up"*, *"Akari"*, *"Bijutsukan"*) - это логическая головоломка. Игровое поле состоит из белых и черных клеток; в некоторых черных клетках расположены числа. Необходимо разместить "светильники" в белых клетках таким образом, чтобы все игровое поле было освещено, но фонари не "светили" бы друг на друга.

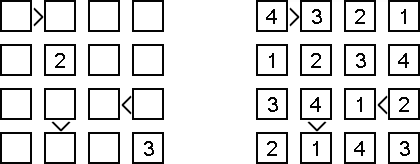
Свет фонаря распространяется по горизонтали и по вертикали, но может быть заблокирован черной клеткой. В черной клетке может находиться число от 0 до 4, указывая, сколько фонарей должно быть размещено рядом с ней (не учитываются фонари, помещенные по диагонали от этой черной клетки). Если клетка не содержит числа, около нее может быть размещено любое количество фонарей.



**Филломино** (*"Fillomino"*) представляет собой прямоугольную сетку произвольного размера; в некоторых клетках находятся числа. Необходимо разбить игровое поле на блоки; блок должен содержать столько клеток, сколько обозначено числом в клетках блока. Блоки, имеющие одинаковый размер, не должны соприкасаться по горизонтали или по вертикали. Клетки, которые изначально не содержали чисел, также могут быть объединены в блоки, необходимые для решения головоломки.

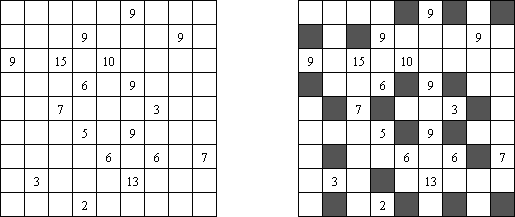


**Неравенство** (*"Futoshiki"*, *"Hutoshiki"*, *"Unequal"*) - это логическая головоломка с числами. Игровое поле представляет собой квадратную сетку; в некоторых клетках могут стоять цифры; между клетками могут присутствовать знаки "<" и ">", которые показывают соотношения, установленные между соседними цифрами. Необходимо заполнить свободные клетки цифрами так, чтобы в каждой строке и в каждом столбце каждая цифра встречалась бы только один раз.



**Куромасу** (*"Kuromasu"*, *"Kurodoko"*, *"Where is black cells?"*) представляет собой прямоугольную сетку, в некоторых клетках которой могут стоять числа. Необходимо закрасить черным цветом клетки, соблюдая следующие условия:

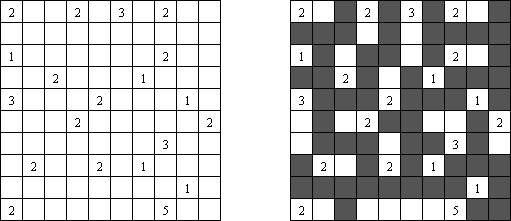
* каждое число означает, сколько белых клеток "видно" из данной клетки по вертикали и по горизонтали; "взгляд" простирается до тех пор, пока не достигнет края поля или на пути не встретится клетка черного цвета;
* клетки с числами всегда остаются белыми;
* черные клетки не могут соприкасаться по вертикали или по горизонтали;
* все белые клетки составляют единое белое поле, то есть соприкасаются по вертикали или по горизонтали.



**Нурикабе** (*"Nurikabe"*, *"Cell Structure"*, *"Islands in the Stream"*) - логическая головоломка с числами. В японской мифологии "нурикабе" - это чудовище в виде большой невидимой стены, загораживающей проход.

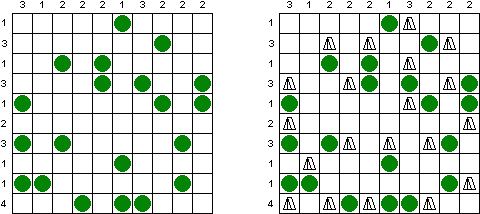
Необходимо восстановить карту, на которой изображено расположение островов, соблюдая следующие правила:

* информация о каждом острове представлена в виде числа, показывающего количество клеток, которые занимает этот остров;
* любые два острова могут соприкасаться только углами;
* все острова содержат в своем описании только одно число;
* между островами протекает река;
* все клетки реки должны быть соединены между собой;
* на карте не должно присутствовать ни одного квадрата размерами **2 x 2**, все клетки которого содержат реку.

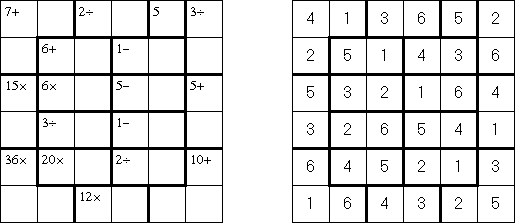


**Лагерь** (*"Tents"*, *"Tents and Trees"*) представляет собой прямоугольную сетку, некоторые клетки которой содержат "деревья". Необходимо разместить рядом с деревьями "палатки", соблюдая следующие правила:

* Число палаток равняется числу деревьев.
* Каждая палатка располагается рядом со "своим" деревом по горизонтали или вертикали, но не по диагонали. Если это условие выполнено, расположение по отношению к "чужим" деревьям значения не имеет.
* Две палатки не могут располагаться в соседних клетках, в том числе и по диагонали.
* Числа сбоку и сверху означают, сколько палаток находится в этой строке или столбце.

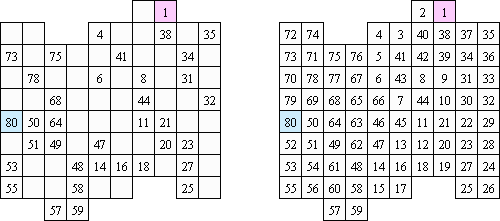


**Кен-кен** (*"KenKen"*, *"KENKEN"*, *"KenDoku"*, *"CalcuDoku"*, *"Square Wisdom"*) - это математическая и логическая головоломка. Необходимо заполнить сетку цифрами так, чтобы в каждой строке и в каждом столбце они не повторялись. Число в углу каждого выделенного блока является результатом арифметической операции над цифрами в этом блоке. В отличие от судоку-убийцы (сум-до-ку), цифры внутри блока могут повторяться.

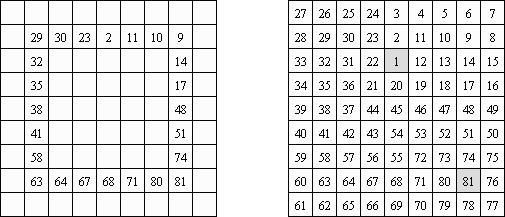


**Хидато** (*"Hidato"*; в переводе с идиш: *"моя головоломка"*; еще одно название - *"Hidoku"*) была изобретена израильским математиком Гиором Бенедеком (Gyora Benedek). Головоломка представляет собой поле произвольной формы (чаще всего прямоугольное или квадратное), состоящее из клеток. Необходимо заполнить все клетки последовательными числами, которые соединены горизонтально, вертикально или по диагонали. В каждой головоломке уже присутствуют наименьшее и наибольшее числа. Также на поле могут стоять и другие числа, чтобы облегчить игроку процесс разгадывания и обеспечить единственность решения задачи.

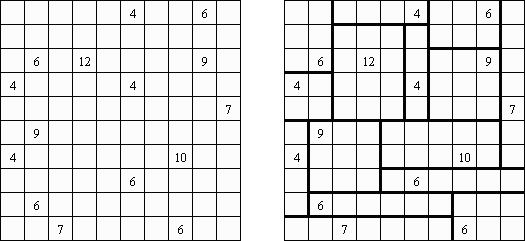
В России головоломку еще называют *"Путь короля"*, так как движение от одного числа к другому напоминает перемещение шахматного короля по доске.



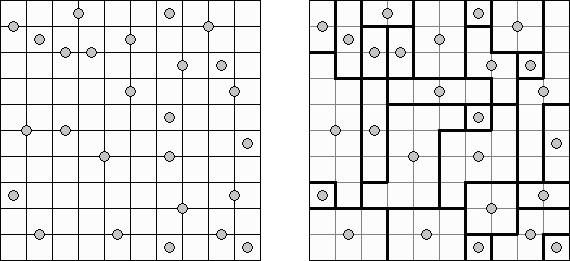
**Нумератор** (*"Numbrix"*) - вид логической головоломки. Представляет собой прямоугольную сетку, в некоторых клетках которой стоят числа. Требуется заполнить пустые клетки таким образом, чтобы все числа были соединены последовательно, по горизонтали или вертикали. Перемещение по диагонали не допускается.



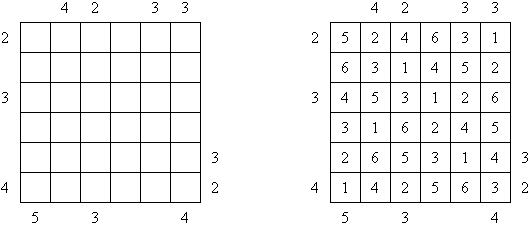
**Квадраты** (*"Shikaku"*, *"Divide by Squares"*, *"Divide by Box"*, *"Number Area"*) - это логическая головоломка. Она представляет собой прямоугольную сетку без стандартного размера. Некоторые клетки сетки содержат числа. Необходимо прочертить линии, разделяющие сетку на прямоугольные и квадратные регионы таким образом, чтобы каждый регион содержал только одно число, равное площади этого региона.



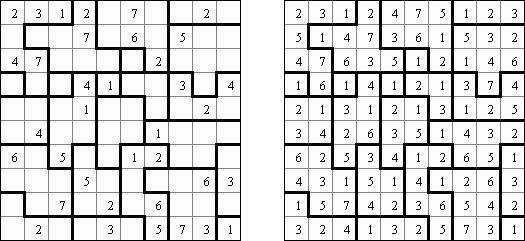
**Галактики** (*"Galaxies"*, *"Tentai Show"*) - это задача, соединяющая в себе логику и геометрию. Головоломка представляет собой прямоугольную сетку с точками. Необходимо разделить сетку на регионы; каждый регион должен содержать только одну точку. Точка является центром симметрии региона.



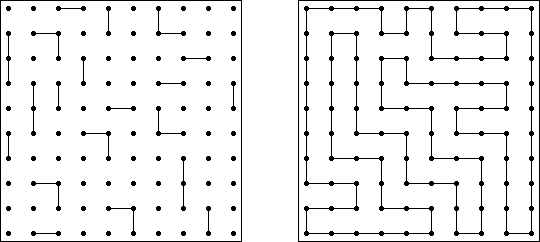
**Небоскрёбы** (*"Skyscrapers"*) - вид логической головоломки. В ней требуется заполнить числами квадрат так, чтобы в каждой строке и в каждом столбце каждое число использовалось лишь единожды. Заполнение нужно делать с помощью ключевых чисел, считая, что в каждой клетке внутри квадрата стоит дом с количеством этажей, равном стоящему в клетке числу, а числа вне поля означают, сколько домов видно в соответствующей строке или столбце, если смотреть от этого числа.



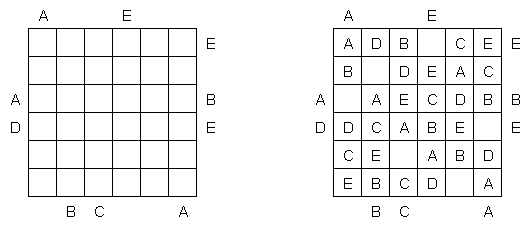
**Волновой эффект** (*"Ripple Effect"*, *"Hakyuu"*, *"Seismic"*) - вид логической головоломки. Задание состоит из прямоугольной сетки, разделенной на блоки ("комнаты"); в некоторых клетках могут стоять числа. Необходимо заполнить все клетки числами так, чтобы каждый блок содержал числа от 1 до количества клеток в блоке. Если ряд клеток или столбец содержат два одинаковых числа, количество клеток между ними больше или равно этим числам. Например, если столбец содержит две клетки с числом 5, между ними должно быть по крайней мере пять клеток с другими числами.



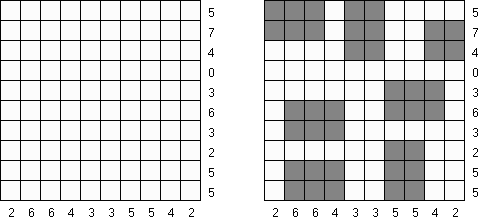
**Гранд тур** (*"Grand Tour"*; еще одно название - **Петля с фрагментами**) - логическая головоломка, в которой требуется провести не касающуюся и не пересекающую себя замкнутую линию через все точки прямоугольной или квадратной сетки. Для того, чтобы обеспечить единственность решения головоломки, несколько точек уже соединены отрезками.



**Первые встречные** (*"Easy as ABC"*, *"ABC End View"*, *"Last Man Standing"*; еще одно название - **Букварики**) - логическая головоломка с буквами. Необходимо заполнить квадратную сетку латинскими буквами (например, от A до E), так чтобы каждый символ встречался в каждой строке и в каждом столбце ровно один раз. Некоторые клетки сетки могут быть пустыми. Буква, стоящая на границе сетки, показывает - какая буква встретится первой в данной строке (столбце).

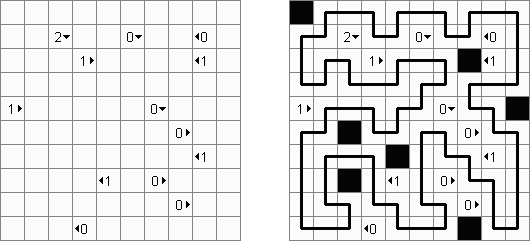


**Облака** (*"Clouds"*, *"Radar"*; еще одно название - **Синоптик**) - это разновидность головоломки "Морской бой". В сетке произвольной формы необходимо расставить прямоугольники ("облака") так, чтобы они не касались друг друга даже углами. Числа по краям сетки означают количество клеток, занятых облаками. Количество и размер облаков не заданы; известно лишь, что стороны прямоугольников имеют длину не меньше двух.

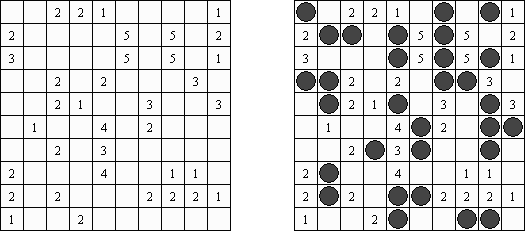


**Бумеранг** (*"Yajilin"*, *"Arrow Ring"*, *"Straight and Arrow"*) - логическая головоломка квадратной или прямоугольной формы. Задание содержит клетки со стрелками и числами; эта информация необходима для расстановки клеток черного цвета на поле. Цель - нарисовать замкнутую, не пересекающую себя петлю, которая не должна проходить через клетки со стрелками или черные клетки.

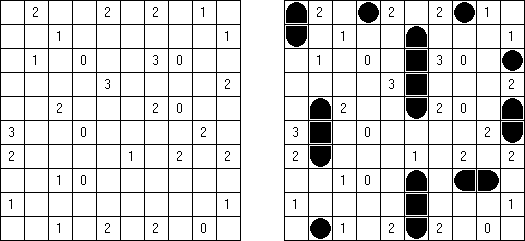
* Числа в клетках указывают, сколько именно черных клеток находится в том ряду или том столбце и в том направлении, в котором указывает стрелка. Сами клетки со стрелками не считаются за черные клетки.
* Если клетка не содержит стрелку и не является черной, то она должна содержать сегмент петли.
* Числа в клетках указывают точное число черных клеток. То есть, если ячейка сетки содержит число "3" и стрелку, указывающую влево, это означает, что в ряду слева от такой ячейки располагаются три черных клетки.
* Черные клетки не должны соприкасаться по горизонтали или по вертикали.
* На поле могут быть черные клетки, на которые не указывает ни одна стрелка.



**Сапёр** (*"Minesweeper"*) известен всем пользователям операционной системы Microsoft Windows. Необходимо расставить "мины" в свободных клетках, используя ключевые числа. Каждое ключевое число показывает, сколько соседних с ним клеток занято минами. Мины расположены по одной в каждой клетке, а в клетке с числом мин не бывает.

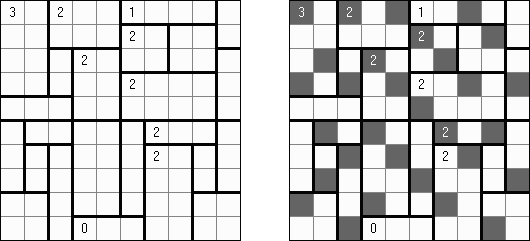


**Сапёр + Морской бой** (*"Minesweeper Battleships"*) объединяет в себе черты двух популярных головоломок. Необходимо расставить на поле корабли таким образом, чтобы они не касались даже углами. В некоторых клетках сетки стоят цифры. Эти цифры показывают, сколько фрагментов кораблей находится вокруг данной клетки (включая диагонали).

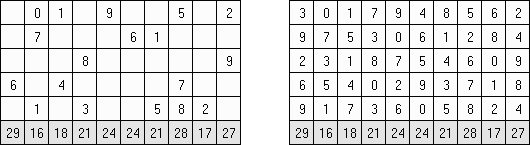


**Комнаты** (*"Heyawake"*) - логическая головоломка. Она представляет собой прямоугольную сетку, разделенную на блоки ("комнаты"). Некоторые блоки могут содержать числа; число показывает, сколько именно черных клеток должен содержать блок. Если число в блоке отсутствует, такой блок может содержать любое количество черных клеток.

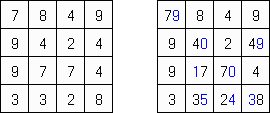
* Черные клетки не должны соприкасаться по горизонтали или по вертикали (только по диагонали).
* Все белые клетки соединены друг с другом по горизонтали или по вертикали.
* Непрерывная линия из белых клеток не должна пересекать более двух "комнат".



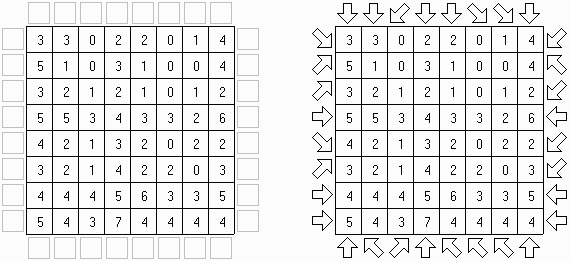
**Числобус** (*"Tenner Grid"*, *"From 1 to 10"*, *"Zehnergitter"*) представляет собой прямоугольную сетку шириной десять клеток. Необходимо заполнить сетку таким образом, чтобы каждый ряд содержал цифры от 0 до 9. В столбцах числа могут повторяться. Число внизу сетки означает сумму всех цифр в столбце. Числа, находящиеся в смежных клетках (даже если клетки соприкасаются лишь по диагонали), должны быть разными.



**Сто** (*"Hundred"*) состоит из квадратной сетки, в каждой клетке которой проставлены цифры. Требуется поставить дополнительные цифры таким образом, чтобы сумма чисел в каждом ряду и каждом столбце была равна 100.

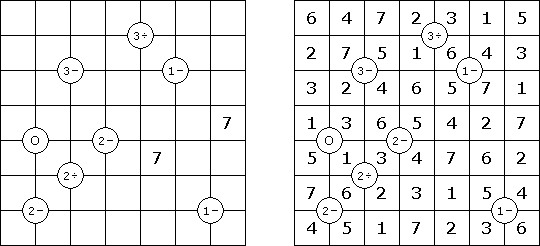


**Стрелки** (*"Arrows"*) - это разновидность логической головоломки. Она состоит из сетки прямоугольной или квадратной формы, заполненной числами. Цель головоломки - расставить стрелки за пределами сетки. Каждая стрелка указывает хотя бы на одну клетку с числом. Число означает количество стрелок, указывающих на данную клетку.

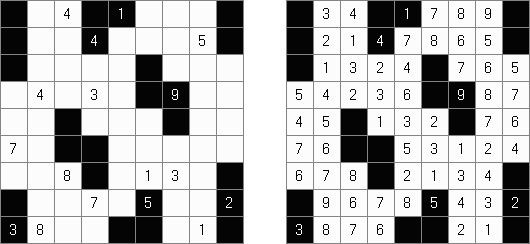


**Матракс** (*"Mathrax"*) - разновидность логической головоломки. Необходимо заполнить числами квадрат так, чтобы в каждой строке и в каждом столбце каждое число использовалось лишь единожды. Некоторые из чисел уже могут присутствовать в сетке. На пересечениях линий сетки могут располагаться кружки с дополнительными условиями:

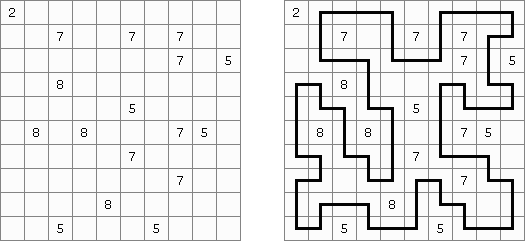
* Число и знак арифметического действия (сложение, вычитание, умножение, деление) - число в кружке означает результат выполнения арифметического действия над парами чисел в клетках, примыкающих к кружку и соприкасающихся друг с другом по диагонали.
* Латинская буква "**E**" - в клетках, примыкающих к кружку, все четыре числа являются чётными ("E" - от английского слова "even", чётный).
* Латинская буква "**O**" - в клетках вокруг кружка все четыре числа являются нечётными ("O" - от английского слова "odd", нечётный).



**Стрейтс** (*"Str8ts"*, *"Straights"*; от термина в карточной игре покер, обозначающего пять карт по порядку) - логическая головоломка, придуманная Джеффом Виддеричем (Jeff Widderich) из Канады. Сетка квадратной формы содержит черные и белые клетки. Необходимо расставить числа в белых клетках таким образом, чтобы в промежутках между черными клетками образовывались наборы последовательных чисел, но не обязательно в порядке возрастания или убывания (например: 2-1-3-4). В каждой строке и в каждом столбце числа, стоящие в белых и черных клетках, не должны повторяться. Числа в черных клетках не входят в наборы последовательных чисел.

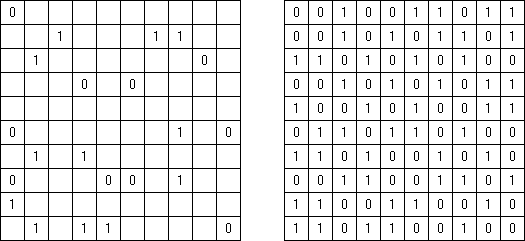


**Линейщик** (*"Linesweeper"*) - логическая головоломка, правила которой похожи на правила головоломки "Сапёр". Задание представляет собой квадратную или прямоугольную сетку; в некоторых клетках сетки стоят числа (возможны значения от 0 до 8). Необходимо провести замкнутую линию через клетки; линия не должна пересекать саму себя или проходить через клетки с числами. Цифра в клетке показывает, через какое количество клеток, расположенных вокруг клетки с числом, проходит линия.

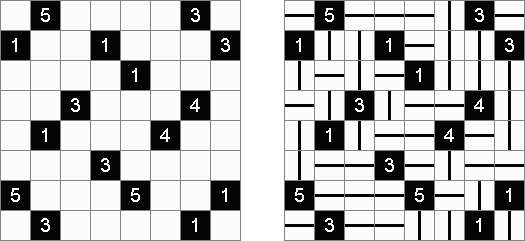


**Двоичный код** (*"Binairo"*, *"Binary Puzzle"*, *"Takuzu"*, *"Tohu wa Vohu"*) - логическая головоломка с использованием цифр "0" и "1". Прямоугольную или квадратную сетку необходимо заполнить цифрами в соответствии со следующими правилами:

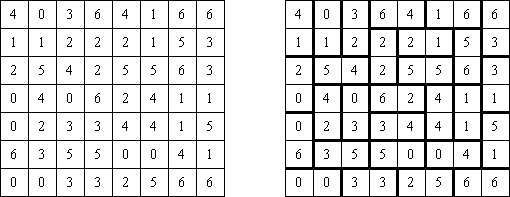
* Каждая строка и каждый столбец содержат столько же цифр "1", как и цифр "0" (или на одну больше для сеток с нечётными размерами).
* Одна и та же цифра может стоять лишь в двух ячейках подряд.
* Каждая строка должна быть уникальной, и каждый столбец должен быть уникальным.



**Стены** (*"Walls"*) - логическая головоломка; её изобрёл Наоки Инаба (Naoki Inaba) из Японии. Необходимо расставить в белых клетках горизонтальные и вертикальные линии так, чтобы суммарная длина всех "лучей", исходящих из клетки чёрного цвета, совпадала с числом, стоящим в этой клетке.

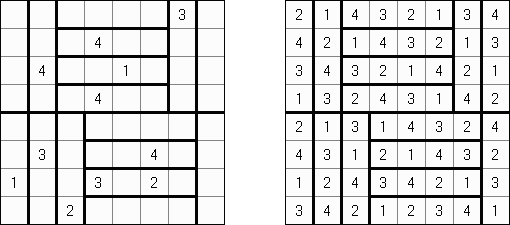


**Домино-пасьянс** (*"Dominosa"*, *"Dominosa Omnibus"*, *"Solitaire Dominoes"*, *"Domino Hunt"*) - логическая головоломка, в которой используются костяшки домино. На поле показаны только числа; необходимо восстановить границы между костяшками домино.

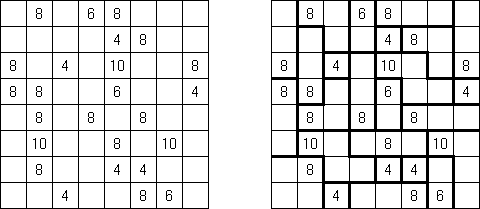


**Лоскутки** (*"Patchwork"*, *"Tatami"*; еще одно название - **Магические полоски**) - головоломка, которая представляет собой квадратную сетку, поделенную на регионы одинакового размера ("комнаты"). Необходимо заполнить каждую комнату цифрами от 1 до числа, равного количеству клеток в регионе. Каждый ряд и каждая колонка должны содержать одинаковые количества каждого вида цифр. Соседние клетки, соприкасающиеся по горизонтали или вертикали, не должны содержать одинаковые цифры.

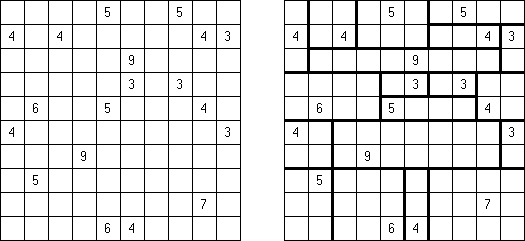
В некоторых заданиях этой разновидности головоломок используются буквы вместо цифр. **Cross+A** может решать головоломки как с цифрами, так и с буквами в задании.



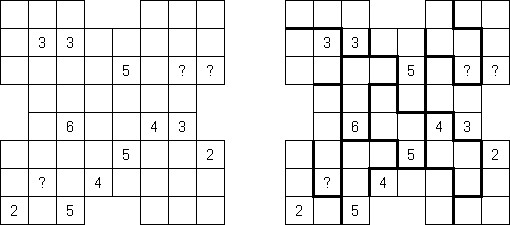
**Особняки** (*"Knossos"*, *"Кносс"* - древний город и дворец на острове Крит, а также местонахождение лабиринта, в котором был заточён Минотавр) представляет собой прямоугольную или квадратную сетку, в некоторых ячейках которой стоят числа. Необходимо разделить сетку на регионы ("комнаты") таким образом, чтобы в каждом регионе находилось по одному числу. Это число должно равняться длине периметра региона.



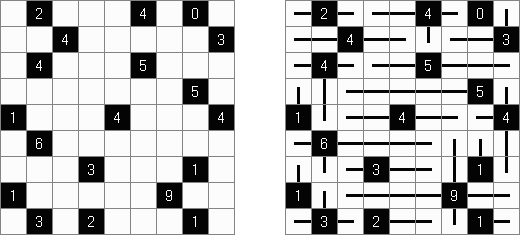
**Рекуто** (*"Rekuto"*) - логическая головоломка квадратной или прямоугольной формы. В некоторых клетках сетки стоят числа. Необходимо прочертить линии, разделяющие сетку на прямоугольные и квадратные регионы таким образом, чтобы каждый регион содержал только одно число, равное сумме ширины и высоты региона.



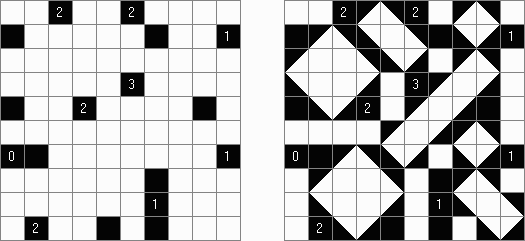
**Соседи** (*"Neighbours"*) - это логическая головоломка, представляющая собой сетку произвольной формы. В некоторых клетках сетки проставлены числа или знаки вопроса. Необходимо поделить сетку на регионы с равным количеством клеток. Каждый регион должен содержать только одну клетку с числом (или со знаком вопроса); это число показывает, сколько именно "соседей" должно быть у данного региона. Два региона считаются соседними, если у них есть общая граница. Если регион содержит клетку со знаком вопроса, у такого региона может быть любое количество "соседей".



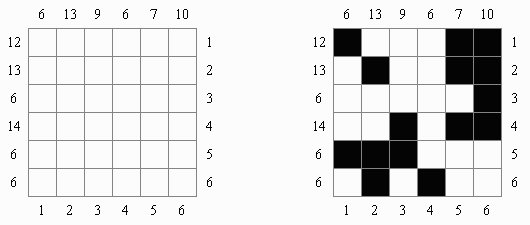
**Роза ветров** (*"Four Winds"*) - логическая головоломка, которая представляет собой квадратную или прямоугольную сетку с черными и белыми клетками. Необходимо нарисовать прямые линии, исходящие из черных клеток и проходящие через все белые клетки. Число в черной клетке показывает, сколько белых клеток занимают линии, соединенные с этой черной клеткой. Линии не должны пересекаться.



**Шака-шака** (*"Shakashaka"*, *"Proof of Quilt"*) - логическая головоломка квадратной или прямоугольной формы. Задание содержит белые и черные клетки; в некоторых черных клетках могут стоять цифры (от 0 до 4). Необходимо расставить в белых клетках черные треугольники таким образом, чтобы сформировать прямоугольные и квадратные области белого цвета. Некоторые белые клетки могут оставаться пустыми. Области могут быть ориентированы горизонтально, вертикально или по диагонали. Соседние области белого цвета не должны иметь общих сторон. Цифра в черной клетке показывает, сколько треугольников соприкасаются с этой клеткой.

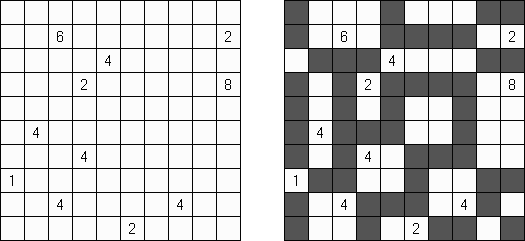


**Какурасу** (*"Kakurasu"*) - головоломка квадратной или прямоугольной формы, в которой необходимо расставить черные и белые клетки. Числа слева и сверху от сетки обозначают суммы "весов" черных клеток в рядах и столбцах. Числа справа и снизу от сетки определяют "вес" черных клеток в соответствующих рядах и столбцах ("вес" черного квадрата в первом ряду или первом столбце равен 1, во втором ряду или втором столбце - 2 и т.д.).

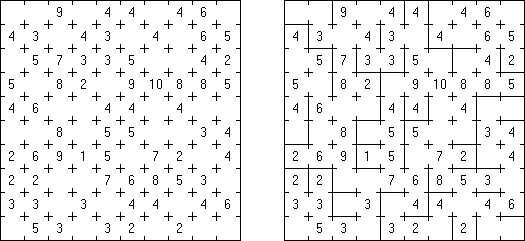


**Мочикоро** (*"Mochikoro"*, *"Mochinuri"*) представляет собой прямоугольную или квадратную сетку, в некоторых ячейках которой стоят числа. Необходимо разместить белые и черные клетки таким образом, чтобы образовались прямоугольные или квадратные «острова» из белых клеток.

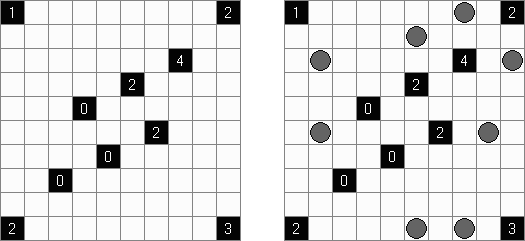
* Один белый «остров» может содержать не более одной клетки с числом; число означает количество белых клеток, принадлежащих этой области.
* Некоторые «острова» не содержат клеток с числами.
* Регионы белого цвета не могут соприкасаться сторонами, но все белые «острова» касаются друг с друга по диагонали.
* В сетке не должно быть ни одного квадрата **2 x 2**, состоящего только лишь из черных клеток.



**Двери** (*"Seethrough"*, *"Doors"*, *"Open Office"*) - логическая головоломка квадратной или прямоугольной формы, где каждая ячейка обозначает "комнату". Требуется закрыть или открыть "двери" между комнатами. Открытые двери позволяют смотреть из комнаты в комнату или сквозь несколько комнат. Число обозначает, сколько комнат видно из данной комнаты, за исключением самой этой комнаты. Не должно быть изолированных друг от друга частей сетки, то есть из любой комнаты можно попасть в любую комнату.

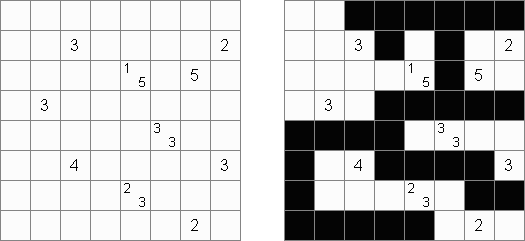


**Маяки** (*"Lighthouses"*) - логическая головоломка, содержащая черные клетки с числами ("маяки"). Число означает количество кораблей, которые освещены данным маяком. Маяк освещает корабль, если они находятся в одном ряду или колонке, даже если между маяком и кораблем расположены другие корабли или маяки. Каждый корабль освещен по крайней мере одним маяком. Корабли располагаются таким образом, что не соприкасаются с маяками или друг с другом, в том числе и по диагонали.

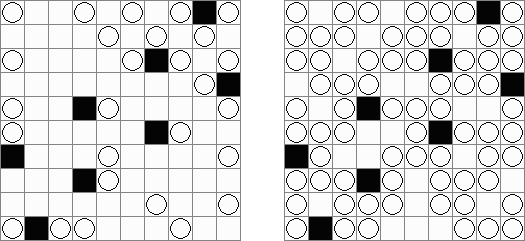


**Тапа** (*"Tapa"*) была изобретена Серканом Юрекли (Serkan Yürekli) из Турции. Головоломка представляет собой квадратную или прямоугольную сетку, состоящую из белых клеток. Необходимо закрасить некоторые клетки черным цветом таким образом, чтобы выполнялись следующие правила:

* Все черные клетки должны соприкасаться друг с другом по вертикали или по горизонтали.
* Не должно быть группы клеток размером **2 x 2**, состоящей из одних лишь черных клеток.
* Некоторые из клеток содержат числа или знаки вопроса; такие клетки всегда остаются белыми.
* Каждое число обозначает непрерывную группу черных клеток вокруг белой клетки с этим числом; две такие группы черных клеток обязательно должны быть разделены хотя бы одной белой клеткой.
* Знак вопроса обозначает любое число больше нуля.
* Позиции чисел или порядок их размещения внутри клетки не имеют значения.



**Четвёртый лишний** (*"Forbidden Four"*) - логическая головоломка, изобретенная Наоки Инаба (Naoki Inaba) из Японии. Головоломка представляет собой прямоугольную или квадратную сетку, в некоторых ячейках которой стоят круги. Необходимо расставить в пустых клетках круги таким образом, чтобы все они соприкасались друг с другом по горизонтали или по вертикали. Не должно быть четырех подряд кругов по горизонтали или по вертикали.



## Дополнительная Лабораторная работа №2 Рекурсивные графические алгоритмы (Фракталы)

Пользователь сам задает глубину рекурсии

|  |  |  |
| --- | --- | --- |
| **Задача 1.** Рекурсивное рисование папоротника | **Задача 2.** Рекурсивное рисование ёлки | **Задача 3.** Рекурсивное рисование фигуры. |
| http://www.chair36.msiu.ru/uploads/jewexntvaiwe/lecture_notes/fern.gif | http://www.chair36.msiu.ru/uploads/jewexntvaiwe/lecture_notes/tree_new.png |  |
| **Задача 4.** Рекурсивное рисование фигуры. | **Задача 5.** Рекурсивное рисование фигуры. | **Задача 6.** Рекурсивное рисование фигуры. |
|  |  |  |
| **Задача 7.** Рекурсивное рисование фигуры. Треугольника Серпинского. | **Задача 8.** Рекурсивное рисование фигуры. | **Задача 9.** Рекурсивное рисование фигуры. |
|  |  |  |
|  |  |  |

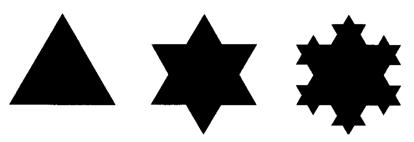
|  |  |  |
| --- | --- | --- |
| **Задача 10.** Рекурсивное рисование фигуры. | **Задача 11.** Рекурсивное рисование фигуры (размер расстояния между параллельными линиями одинаковый) | **Задача 12.** Рекурсивное рисование фигуры. (см. **задание 11**, но размер расстояния между параллельными линиями уменьшающийся). |
|  |  |  |
| **Задача 13.** Рекурсивное рисование фигуры. | | **Задача 14.** Рекурсивное рисование фигуры. |
|  | |  |
| **Задача 15.** Рекурсивное рисование фигуры. |  |  |
|  |  |  |
|  |  |  |

**Задача 4.** Нарисовать фигуру.

**Задача 5.** Нарисовать фигуру.

**Задача 6.** Рекурсивное рисование фигуры кривую Гильберта.

**Задача 7.** Рекурсивное рисование фигуры кривой (снежинки) Коха.



**Задача 8.** Нарисовать фигуру.

 рисования **кривой Коха**