

## ЗАДАНИЕ 1

### Повелитель вселенной

Реализовать игру "Жизнь" на C++.

#### Правила.

- Поле действия для игры — условная «вселенная» — размеченная на клетки плоскость (ограниченная в размерах).
- Каждая клетка на этой поверхности может находиться в двух состояниях: быть «живой» (заполненной) или быть «мёртвой» (пустой). Клетка имеет восемь соседей, окружающих её.
- Распределение живых клеток в начале игры называется **первым поколением**. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:
  - в пустой (мёртвой) клетке, рядом с которой ровно три живые клетки, зарождается жизнь;
  - если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае, если соседей меньше двух или больше трёх, клетка умирает («от одиночества» или «от перенаселённости»);
  - за пределами игровой решётки жизни нет.
- Игра прекращается, если:
  - на поле не останется ни одной «живой» клетки;
  - при очередном шаге ни одна из клеток не меняет своего состояния (складывается стабильная конфигурация; предыдущее правило, вырожденное до одного шага назад);
  - конфигурация на очередном шаге в точности (без сдвигов и поворотов) повторит себя же на одном из более ранних шагов (складывается периодическая конфигурация).

#### Требования к программе:

- предусмотреть как минимум 2 способа задания первого поколения: случайная конфигурация на исходном поле и чтение конфигурации из файла;
- попыбывать придумать алгоритм смены поколений не основанный на полном переборе и просмотре всех клеток поля;
- как реализовать выявление периодических конфигураций (идея, алгоритм, реализация).

## **ЗАДАНИЕ 2**

### **Математика никак не отпускает**

Реализовать вычисление произвольных арифметических (для начала) выражений. Пример выражения:

$$2.56 + 8/9.5 - (345 - 3 * 2) * 0.75$$

#### **Требования к программе:**

- иметь ввиду несколько источников ввода выражений, а не только консольный ввод.

### ЗАДАНИЕ 3

#### Стилист нужен даже исходному коду

Реализовать программу, которая форматирует файл с исходным кодом на C++ по заданным правилам оформления.

**Требования к программе:**

- составить набор общих *правил*, как то, глубина отступа, перенос фигурных скобок, отступы между последовательными функциями / методами;
- придумать формат конфигурации для задания правил.

## ЗАДАНИЕ 4

### Запутывать людей - это нехорошо. Но интересно

Реализовать программу, которая строит лабиринт на заданной прямоугольной плоскости заданного размера.

#### Требования к программе:

- лабиринт должен иметь единственное решение;
- все ячейки должны быть соединены с путём, который даёт основное решение;
- вход и выход из лабиринта располагаются на противоположных сторонах прямоугольника;
- предусмотреть как случайный выбор входа и выхода, так и заданный вручную (с проверкой предыдущего условия);
- предусмотреть вывод полученного лабиринта в различные выходные устройства (консоль или файл).

#### Рассуждение:

Один из возможных подходов к решению таков. Выбираем вход; затем, начав от него, добавляем по одной ячейке к главному пути-решению, пока он не достигнет выходной стороны. После этого удаляем некоторые внутренние стенки так, чтобы все клетки оказались соединенными с главным путем. Чтобы главный путь не получился прямым коридором, следует при его построении предусмотреть случайные повороты. Программа должна также следить за тем, чтобы при построении главного пути или при открытии боковых ячеек не нарушалась единственность решения.

## ЗАДАНИЕ 5

### ЭВМ тоже хочет поиграть

Разработать программу, которая отгадывает заданную комбинацию цифр с помощью стратегии, отличной от полного перебора всех комбинаций. Задача основана на игре «Великий комбинатор» (англ. «Mastermind»), также известная как «Быки и коровы».

#### Что за игра:

Один из игроков, «загадывающий», записывает секретную комбинацию из любых четырех цифр от 1 до 6 (повторения допускаются), называемую кодом. Второй игрок, «отгадывающий», пытается раскрыть код, высказывая разумные предположения, называемые пробами. Каждая проба, как и код, представляет собой произвольную комбинацию из четырех цифр в диапазоне от 1 до 6. Отгадывающий игрок сообщает пробу загадывающему, и тот должен ответить, сколько цифр в пробе совпадает с цифрами кода как по положению, так и по величине и сколько из остальных цифр пробы входят в код, но стоят на другом месте. Так, на пробу 1424 при коде 4221 будет получен ответ: «Одна цифра совпадает и стоит на том же месте, и ещё две совпадают, но стоят на других местах». Тур игры продолжается до тех пор, пока отгадывающий не назовет пробу, в точности совпадающую с кодом, т. е. пока не отгадает код.

**Что нужно сделать:** Продумать и написать алгоритм при котором компьютер выступает в роли «отгадывающего».

#### Возможные подход к решению:

Стратегия для реализации алгоритма, предложенная Бобом Кули (Bob Cooley) из Lawrence Livermore Laboratory. Центральное место в стратегии занимает идея **пространства решений**. Начальное пространство решений  $P_0$  состоит из всех возможных кодов (и имеет, следовательно, 64 элементов); после  $i$ -й пробы  $G_i$  пространство  $P_i$  состоит из всех тех членов пространства  $P_{i-1}$ , которые не опровергаются ответом  $R_i$ . Иными словами, пространство  $P_i$  — это множество всех комбинаций, которые все еще могут быть кодом; задача отгадывающего — свести пространство к одному элементу.

Сама стратегия заключается в следующем. Пробой  $G_1$  пусть будет любая случайно выбранная комбинация с одной повторяющейся цифрой, например 4311, 6552 или 1335. Выполните эту пробу и постройте пространство  $P_1$  на основе ответа  $R_1$ . Новая проба  $G_{i+1}$  ищется по пространству  $P_i$ ,  $i \geq 1$ , путем поочередного сравнения всех комбинаций из  $P_i$  с пробой  $G_i$ . В качестве следующей пробы выбирается наименее похожая на  $G_i$  комбинация  $S$ . Мерой сходства служит число точных совпадений, а в случае равенства — число цифр, совпадающих по значению, но расположенных по-другому. Так, среди трех комбинаций 2641, 2356 и 1345 наиболее похожей на 2345 будет 1345, а 2641 — наименее похожей. Если имеется несколько наименее похожих комбинаций, то можно выбрать любую кандидатуру случайным образом. Тур прекращается, когда будет получен ответ «четыре точных попадания», и, разумеется, в случае пространства из одного элемента в качестве следующей пробы всегда надо брать этот элемент. Как показывают эксперименты, размеры пространства решений сокращаются после каждой пробы примерно в 4 раза и никогда не требуется более шести проб.

#### Над чем поразмыслить в случае успеха:

- Возможно ли обобщить алгоритм на произвольное количество загадываемых цифр (а не четыре, как приведено в описании)?