

\* Необходимо решить одну задачу на выбор.

## Игра «Жизнь»

Необходимо написать программу, которая будет рассчитывать состояние системы на определенный заданный ход.

Место действия этой игры — размеченная на клетки поверхность (квадратное поле  $n \times n$ ,  $n$  не более 20). Каждая клетка на этой поверхности может находиться в двух состояниях: быть «живой» (заполненной) или быть «мёртвой» (пустой). Клетка имеет всегда восемь соседей, окружающих её (т.е. крайние клетки с одной стороны «соприкасаются» с клетками с другого края). 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»)

Входные данные необходимо считывать из файла «input.txt». Формат начальных данных:

- в первой строке будет число  $m$  ( $1 < m < 20$ ) номер хода, поколение, состояние которого необходимо рассчитать.
- в последующих  $n$ -строках будет записано начальное состояние системы (положение первого поколения) —  $n$ -цифр через пробел, в формате 1 — в этой клетке на старте есть жизнь, 0 — жизни нет.

Результат работы программы — итоговое поле  $n \times n$ , состояние системы после  $m$  ходов, необходимо вывести в файл «output.txt»

Формат выходных данных соответствует формату входных: 1 – в клетке есть жизнь, 0 – в клетке жизни нет. Клетки разделяются пробелами.

Пример формата входных данных:

```
6
1 0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 1 0
0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 0 1 0
1 0 0 0 0 0 0 0 0
```

Пример выходного файла с решением:

```
0 0 0 0 1 0 1 1 1
0 0 1 1 0 0 0 0 0
0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 0 1 1
1 0 0 0 0 0 0 0 1
0 0 0 0 0 1 0 0 1
1 0 0 0 0 0 0 0 1
1 0 0 1 0 0 0 1 1
1 0 0 1 1 1 1 1 0
```

# Решение «Судоку»

Необходимо написать программу, которая найдет решение для переданного поля головоломки «Судоку».

Чтение головоломки осуществляется из файла «input.txt»

Запись результата осуществляется в файл «output.txt»

Если решения не существует – вывести на экран слово «Impossible».

Пример формата входных данных (пустые клетки обозначаются нулем):

0	3	5	7	0	0	0	6	0
0	0	0	0	2	0	0	0	0
0	0	1	4	0	0	0	7	9
0	0	0	0	9	0	0	8	2
0	0	6	0	0	0	0	0	4
0	1	0	5	0	7	0	0	0
6	0	0	0	0	0	0	0	5
7	0	3	0	0	5	8	0	0
0	0	0	0	0	2	0	0	0

Пример выходного файла с решением:

4	3	5	7	8	9	2	6	1
9	6	7	1	2	3	4	5	8
2	8	1	4	5	6	3	7	9
3	7	4	6	9	1	5	8	2
5	9	6	2	3	8	7	1	4
8	1	2	5	4	7	6	9	3
6	2	9	8	7	4	1	3	5
7	4	3	9	1	5	8	2	6
1	5	8	3	6	2	9	4	7