

## Задача А. Ферзи

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

На доске  $n \times n$  расставьте, пожалуйста,  $n$  обычных шахматных ферзей так, чтобы они друг друга не били.

### Формат входных данных

В единственной строке входных данных содержится одно целое число  $n$  — размер доски ( $4 \leq n \leq 200$ ).

### Формат выходных данных

Для каждой горизонтали исходной доски выведите номер вертикали, на которой стоит ферзь в этой горизонтали, вертикали нумеруются слева направо, начиная с единицы.

### Пример

стандартный ввод	стандартный вывод
4	3 1 4 2

## Задача В. Хорошие раскраски

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Назовем раскраску клеток таблицы  $n \times m$  хорошей, если никакие четыре клетки, центры которых образуют вершины прямоугольника со сторонами, параллельными осям координат, не покрашены в один цвет. Иначе говоря, для раскраски не должно быть четверки целых чисел  $x_1, x_2, y_1, y_2$ , что  $1 \leq x_1 < x_2 \leq n, 1 \leq y_1 < y_2 \leq m$ , и клетки  $(x_1, y_1), (x_2, y_1), (x_1, y_2)$  и  $(x_2, y_2)$  покрашены в одинаковый цвет.

Требуется написать программу, которая по заданным целым числам  $n, m$  и  $c$  находит любую хорошую раскраску таблицы  $n \times m$  в  $c$  цветов.

### Формат входных данных

В первой строке записаны три целых числа  $n, m, c$  ( $2 \leq n, m \leq 10, 2 \leq c \leq 3$ ).

Гарантируется, что для заданных во входных данных значений существует хотя бы одна хорошая раскраска.

### Формат выходных данных

Выведите  $n$  строк по  $m$  чисел в каждой.

В качестве  $j$ -го числа  $i$ -й строки выведите  $a_{i,j}$  — цвет клетки  $(i, j)$  ( $1 \leq a_{i,j} \leq c$ ).

Если есть несколько хороших раскрасок, можно вывести любую из них.

### Пример

стандартный ввод	стандартный вывод
2 2 2	1 2 2 2

### Замечание

Примените какой-нибудь неточный алгоритм нахождения минимума функции, например алгоритм отжига.