

**Задача А. Жестокая задача**

Имя входного файла: `cruel.in`  
 Имя выходного файла: `cruel.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

Штирлиц и Мюллер стреляют по очереди. В очереди  $n$  человек, стоящих друг за другом. Каждым выстрелом убивается один из стоящих. Кроме того, если у кого-то из стоящих в очереди убиты все его соседи, то этот человек в ужасе убегает. Проигрывает тот, кто не может ходить. Первым стреляет Штирлиц. Требуется определить, кто выиграет при оптимальной игре обеих сторон, и если победителем будет Штирлиц, то найти все возможные первые ходы, ведущие к его победе.

**Формат входных данных**

Входной файл содержит единственное число  $n$  ( $2 \leq n \leq 5\,000$ ) — количество человек в очереди.

**Формат выходных данных**

Если выигрывает Мюллер, выходной файл должен состоять из единственного слова `Mueller`. Иначе в первой строке необходимо вывести слово `Schtirlitz`, а в последующих строках — номера людей в очереди, которых мог бы первым ходом убить Штирлиц для достижения своей победы. Номера необходимо выводить в порядке возрастания.

**Примеры**

| <code>cruel.in</code> | <code>cruel.out</code>    |
|-----------------------|---------------------------|
| 3                     | Schtirlitz<br>2           |
| 4                     | Mueller                   |
| 5                     | Schtirlitz<br>1<br>3<br>5 |

**Задача В. Дровосек**

Имя входного файла: `woodcut.in`  
 Имя выходного файла: `woodcut.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 256 мегабайт

Двое играют в следующую игру: имеется дерево с отмеченной вершиной (корнем). Игроки ходят по очереди. За ход игрок разрубает ветку (стирает ребро), причем из двух полувывшихся компонент связности остается только та, которая содержит корень — остальная отваливается и больше в игре не участвует. Проигрывает тот, кто не может сделать ход.

Определите, может ли выиграть первый игрок, и если да, то укажите любой из его выигрышных ходов.

**Формат входных данных**

В первой строке входного файла находится 2 числа  $N$  и  $R$  — количество вершин дерева и номер корня ( $2 \leq N \leq 1\,000\,000$ ,  $1 \leq R \leq N$ ). Далее следует  $N - 1$  строк, в каждой из которых находятся два числа — номера вершин, которые соединяет очередное ребро.

**Формат выходных данных**

Выведите в выходной файл одно число: 1 или 2 — номер игрока, который выигрывает при правильной игре. Если выигрывает первый игрок, то выведите также любой его выигрышный ход, т.е. порядковый номер ребра во входном файле, которое ему достаточно разрубить первым ходом (число от 1 до  $N - 1$ ).

**Примеры**

| <code>woodcut.in</code> | <code>woodcut.out</code> |
|-------------------------|--------------------------|
| 5 5                     | 1                        |
| 2 3                     | 1                        |
| 1 3                     |                          |
| 2 5                     |                          |
| 4 5                     |                          |

**Задача С. Шашки**

Имя входного файла: `checkers.in`  
 Имя выходного файла: `checkers.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

Как известно, в ЛКШ каждую смену проходит турнир по шашкам. В ЛКШ.Зима организаторы тоже хотели провести такой турнир, но проводить его по стандартным правилам было бы банально. Культурги полезли в Википедию и выяснили, что шашки имеют много разновидностей. Одна из них, «Новогодняя», абсолютна не похожа на привычные нам шашки. Игра происходит на прямоугольной доске  $n \times m$ . В каждой из  $n$  строк расположено ровно 2 шашки (одна белая, другая черная, при этом белая всегда стоит левее черной). За один ход игрок может взять любую шашку своего цвета и передвинуть ее на любую клетку этой же строки, находящуюся ближе к шашке соперника, но не перепрыгивая через нее. Проигрывает тот, кто не может сделать ход. Найдите, кто выиграет при правильной игре.

**Формат входных данных**

Во входном файле записаны через пробел два числа —  $n$  и  $m$  ( $1 \leq n \leq 10^5$ ,  $2 \leq m \leq 10^9$ ). Далее в  $n$  строках записаны по два целых числа —  $l$  и  $r$  ( $1 \leq l < r \leq m$ ) — расположение белой и черной шашки в  $i$ -ой строке соответственно.

**Формат выходных данных**

В выходной файл выведите «First», если у первого игрока есть выигрышная стратегия и «Second» в противном случае.

### Примеры

| checkers.in        | checkers.out |
|--------------------|--------------|
| 1 10<br>1 5        | First        |
| 2 10<br>1 5<br>1 5 | Second       |

### Задача D. Терминатор

Имя входного файла: `terminator.in`  
Имя выходного файла: `terminator.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Два игрока играют в настольную игру. Игровое поле представляет собой квадратный лабиринт,  $8 \times 8$  клеток. В некоторых клетках располагаются стенки. Один игрок управляет фишкой-терминатором, а второй — фишкой-беглецом. Игроки ходят по очереди, ходы пропускать нельзя (гарантируется, что ход всегда возможен). За один ход игрок может переместить свою фишку в любую из свободных клеток, расположенных рядом с исходной по горизонтали, вертикали или по диагонали (то есть ходом короля). Терминатор, кроме того, может стрелять в беглеца ракетами. Выстрел идет по прямой в любом направлении по горизонтали, вертикали или диагонали. Если беглец оказывается на линии выстрела терминатора и не прикрыт стенками, то терминатор незамедлительно делает выстрел (вне зависимости от того, чей ход), и беглец проигрывает. Начальное положение фишек задано. Первый ход делает беглец. Он выигрывает, если сделает ход с восьмой строки за пределы игрового поля, так как остальные границы поля окружены стенками.

Вопрос задачи: может ли беглец выиграть при оптимальной игре обеих сторон?

### Формат входных данных

Во входном файле задано игровое поле. Свободная клетка обозначена цифрой 0, а клетка со стенкой — цифрой 1. Клетка, в которой находится беглец, обозначена цифрой 2, а клетка с терминатором — цифрой 3.

### Формат выходных данных

В выходной файл выведите число 1, если беглец выигрывает, и  $-1$  — в противном случае.

### Примеры

| terminator.in  | terminator.out |
|--|----------------|
| 01000000<br>10100000<br>31100000<br>00020000<br>00000000<br>00000000<br>00000000<br>00000000 | -1             |