

## 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 2
4	Mueller
5	Schtirlitz 1 3 5

## Задача В. 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

<code>terminator.in</code>	<code>terminator.out</code>
01000000 10100000 31100000 00020000 00000000 00000000 00000000 00000000	-1

## Задача С. Игры с Укконеном

Имя входного файла: `ukkonen.in`  
Имя выходного файла: `ukkonen.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 512 мегабайт

На этот раз коварный Ишма Дровкин решил сорвать лекцию параллели А про алгоритм Укконена, позволяющий построить сжатое суффиксное дерево за линейное время, заявившись на неё с правилами новой игры. К счастью, эту лекцию вел сам завуч ЛКШ Станкей Андревич, который рассудил, что новая игра может помочь школьникам в изучении непростого алгоритма. Правила игры заключались в следующем.

В самом начале два участвующих в игре школьника загадывают некоторую строку  $S$ , состоящую из строчных букв латинского алфавита. После этого они по очереди делают ходы, каждый из которых как-то меняет вторую строку  $T$ , которая изначально пуста. При этом первый школьник своим ходом должен дописать ровно один символ в конец строки  $T$ , а второй — в ее начало. Главное правило игры заключается в том, что после каждого хода строка  $T$  должна оставаться подстрокой строки  $S$ . Соответственно, школьник, который не сможет сделать ход, соблюдающий это правило, объявляется проигравшим.

Пока школьники развлекались, играя в эту игру, Станкей Андревич заметил, что в ситуации, когда оба игрока играют оптимально, победителя всегда можно определить по строке  $S$ , загаданной школьниками в самом начале. Чтобы убить оставшееся от лекции время, он решил написать программу, умеющую решать эту задачу. А поскольку завуч ЛКШ никогда и ничего не должен делать сам, вам необходимо сделать это за него.

### Формат входных данных

Первая строка входного файла содержит одно натуральное число  $N$  — количество строк, которые хочет изучить Станкей Андревич. Следующие  $N$  строк содержат изучаемые строки.

Все изучаемые строки состоят только из строчных букв английского алфавита, а их суммарная длина не превышает  $2 \cdot 10^3$ .

### Формат выходных данных

В выходной файл выведите слово «First», если победу при этой строке  $S$  одержит школьник, делающий первый ход в игре, и «Second» — в противном случае.

### Примеры

<code>ukkonen.in</code>	<code>ukkonen.out</code>
2 aaabaaa aaaba	First Second

## Задача D. Битва за кольцо

Имя входного файла: `rings.in`  
Имя выходного файла: `rings.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Саруман Белый и Гэндальф Серый решили сыграть в игру. Победителю достается Кольцо Всевластия. Перед игроками лежат кольца, соединенные в  $K$  цепочек. Для каждого кольца известно содержание золота в нем в процентах — целое число от 1 до 100. Ходят по очереди. За ход разрешается выбрать одну из цепочек и какое-то кольцо из этой цепочки и дематериализовать все кольца из данной цепочки с процентным содержанием золота не больше, чем у выбранного. При этом, понятно, цепочка может распасться на несколько. Игра продолжается на оставшихся цепочках. Тот, кто дематериализовал последнее кольцо, выиграл. Первым ходит Гэндальф. Определите, может ли Гэндальф выиграть и, если может, какой первый ход он должен для этого сделать.

### Формат входных данных

В первой строке дано целое число  $K$  ( $1 \leq K \leq 50$ ). В следующих  $K$  строках приведены описания цепочек в следующем формате: сперва дана длина цепочки — целое число от 1 до 100, затем — процентные содержания золота в кольцах цепочки. Числа в строке разделены пробелом.

### Формат выходных данных

Выведите "S", если Кольцо Всевластия достанется Саруману. В противном случае выведите в первой строке "G", а во второй пару чисел, описывающих выигрышный первый ход Гэндальфа — номер цепочки и номер кольца в ней. Цепочки и кольца внутри цепочек нумеруются с 1. Если существует несколько выигрышных первых ходов, выведите ход с наименьшим номером цепочки, если и таких несколько — с наименьшим номером кольца.

### Примеры

<code>rings.in</code>	<code>rings.out</code>
2 3 1 2 1 1 1	G 1 1
2 3 2 1 2 1 1	S

## Задача Е. Сумма игр

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

Пусть дан ориентированный граф. Стандартная игра на графе заключается в следующем: изначально на одной из вершин графа (называемой начальной позицией) стоит фишка. Двое игроков по очереди двигают её по рёбрам. Проигрывает тот, кто не может сделать ход.

В теории игр часто рассматриваются более сложные игры. Например, прямая сумма двух игр на графах. Прямая сумма игр — это следующая игра: изначально на каждом графе в начальной позиции стоит по фишке. За ход игрок выбирает любую фишку и двигает по ребру соответствующего графа. Проигрывает тот, кто не может сделать ход.

Ваша задача — опеределить, кто выиграет при правильной игре.

### Формат входных данных

На первой строке будут даны числа  $N_1$  и  $M_1$  — количество вершин и рёбер в первом графе ( $1 \leq N_1, M_1 \leq 10\,000$ ). На следующих  $M_1$  строках содержится по два числа  $x$  и  $y$  ( $1 \leq x, y \leq N_1$ ).

В следующих  $M_2 + 1$  строках задан второй граф в том же формате.

Заканчивается входной файл списком пар начальных вершин, для которых нужно решить задачу. На первой строке задано число  $T$  ( $1 \leq T \leq 100\,000$ ) — количество пар начальных вершин. В следующих  $T$  строках указаны пары вершин  $v_1$  и  $v_2$  ( $1 \leq v_1 \leq N_1, 1 \leq v_2 \leq N_2$ ).

### Формат выходных данных

На каждую из  $T$  пар начальных вершин выведите строку “**first**”, если при правильной игре выиграет первый, “**second**”, если второй, или “**draw**”, если будет ничья.

### Примеры

<code>smith.in</code>	<code>smith.out</code>
3 2	<b>first</b>
1 2	<b>second</b>
2 3	
2 1	
1 2	
2	
1 1	
3 2	