

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 входного файла

Входной файл содержит единственное число n ($2 \leq n \leq 5000$) — количество человек в очереди.

Формат выходного файла

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. 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 входного файла

В первой строке входного файла находится 2 числа N и R — количество вершин дерева и номер корня ($2 \leq N \leq 1000000$, $1 \leq R \leq N$). Далее следует $N - 1$ строк, в каждой из которых находятся два числа — номера вершин, которые соединяет очередное ребро.

Формат выходного файла

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. 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, 8×8 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 входного файла

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 выходного файла

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

Примеры

terminator.in	terminator.out
01000000	-1
10100000	
31100000	
00020000	
00000000	
00000000	
00000000	
00000000	
00000000	

Задача D. Флатландское фехтование

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Король Флатландии проводит рыцарский турнир! Победителю достанется полцарства и расположение принцессы, о красоте и мудрости которой ходят легенды. Последней проверкой смелости и силы претендентов будет турнир по фехтованию. Турнир проходит по следующим правилам: участники сражаются один на один, победитель (а точнее, выживший) выходит в следующий круг соревнований.

Перед битвой оба участника становятся в указанные им точки оси Ox с целыми координатами. Затем они совершают ходы по очереди. Первым, естественно, ходит первый участник. На своем ходу первый участник из точки x может переместиться в любую целочисленную точку из отрезка $[x+a; x+b]$, а второй — в любую целочисленную точку из отрезка $[x-b; x-a]$, то есть варианты ходов для игроков симметричны (обратите внимание, что числа a и b не обязаны быть положительными; также, если $a \leq 0 \leq b$, то остаться на месте — корректный ход.) В любой момент участники могут располагаться произвольным образом друг относительно друга, то есть «перепрыгивать» через противника в любую сторону позволяет. Побеждает тот участник, который на своем ходу переместится в точку, где находится его противник.

Разумеется, принцесса уже выбрала себе жениха, и теперь она хочет сделать так, чтобы её избранник выиграл турнир. Он уже вышел в финал турнира, и ему осталась последняя битва. Принцесса просит управляющего турнира расположить участников финала так, чтобы при оптимальных действиях обоих участников выиграл тот, который интересен принцессе. Однако начальное расположение участников уже оглашено, и можно повлиять лишь на то, кто из соперников будет первым участником, а кто — вторым. Но как понять, какой из участников может обеспечить себе победу, ведь принцесса не сильна в военных делах... Поэтому она просит Вас определить, как закончится битва при оптималь-

ных действиях обоих соперников, а также, в случае, если первый участник выигрывает, выигрышный ход для него.

Формат входного файла

В первой и единственной строке входных данных записано четыре целых числа, разделенных пробелами — x_1, x_2, a и b ($x_1 \neq x_2, a \leq b, -10^9 \leq x_1, x_2, a, b \leq 10^9$) — координаты точек, в которых начинают первый и второй участник, а также два числа, определяющие ходы игроков, соответственно.

Формат выходного файла

В первой строке выведите результат битвы в виде: «FIRST» (без кавычек), если при оптимальных действиях обоих игроков выигрывает первый участник, «SECOND» (без кавычек), если выигрывает второй участник, или «DRAW» (без кавычек), если никто не может обеспечить себе победу.

В случае, если выигрывает первый участник, в следующей строке выведите одно целое число x — координату точки, в которую должен пойти первый игрок, чтобы выиграть. Указанный ход должен быть корректным, то есть должно выполняться условие $x_1 + a \leq x \leq x_1 + b$. Если выигрышных ходов несколько, выведите любой. Если первый участник не может обеспечить себе победу, ничего дополнительно выводить не нужно.

Примеры

stdin	stdout
0 2 0 4	FIRST 2
0 2 1 1	SECOND
0 2 0 1	DRAW

Примечание

В первом примере первый участник может выиграть в один ход.

Во втором примере первый участник обязан пойти в точку 1, куда сразу же идет второй участник и выигрывает.

В третьем примере никому из участников не выгодно изменять свое положение, поэтому никто не выигрывает.