Задача A. Конфетки

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

После разгромной победы Директора ЛКШ руководитель физической школы сдался и отпустил Деда Мороза к нам. Но ради интереса предложил сразиться ЛФШатам и ЛКШатам в еще одной непростой игре.

В каждой игре участвует один из вас и один ЛФШонок, а ходите вы по очереди. В кучку перед вами кладется N вкусных конфеток. На каждом ходе игрок может съесть от 1 до K конфеток (больше нельзя — много сладкого вредно даже в Новый Год), но при этом не больше, чем взял его противник на предыдущем ходе (не будем жадничать, мы же добрые). Второго ограничения нет лишь для первого хода каждой игры. Проигрывает тот, кому не осталось конфеток.

У нас возникли подозрения, что директор ЛФШ специально подобрал такие N и K, чтобы ЛКШата никогда не смогли выиграть. Мы надеемся, что это не так, и очень просим вас проверить это.

Формат входных данных

Во входном файле записаны через пробел два целых числа — N ($1 \leqslant N \leqslant 500$) и K ($1 \leqslant K \leqslant 100$).

Формат выходных данных

В выходной файл выведите минимальное число конфет, которое должен съесть ЛКШонок первым ходом, чтобы выиграть при оптимальной игре ЛФШонка, либо 0, если даже самый умный из нас не сможет одолеть идеального играющего противника.

Примеры

sweets.in	sweets.out
7 3	1

Замечание

Если в свой ход ЛФШонок берёт 2 или 3 конфетки, то следующим ходом вы можете закончить игру, если же он возьмёт одну конфетку, то далее каждый будет съедать по одной конфетке и последняя достанется вам.

Задача B. Ретроанализ для маленьких

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

Дан ориентированный весёлый граф из n вершин и m ребер. Оля и Коля играют в игру. Изначально фишка стоит в вершине i. За ход можно передвинуть фишку по любому из исходящих ребер. Тот, кто не может сделать ход, проигрывает. Ваша задача — для каждой вершины i определить, кто выиграет при оптимальной игре обоих.

Формат входных данных

Входные данные состоят из одного или нескольких тестов. Каждый тест содержит описание весёлого ориентированного графа. Граф описывается так: на первой два целых числа n ($1 \leqslant n \leqslant 300\,000$) и m ($1 \leqslant m \leqslant 300\,000$). Следующие m строк содержат ребра графа, каждое описывается парой целых чисел от 1 до n. Пара a b обозначает, что ребро ведет из вершины a в вершину b. В графе могут быть петли, могут быть кратные ребра. Сумма n по всем тестам не превосходит $300\,000$, сумма m по всем тестам также не превосходит $300\,000$.

Формат выходных данных

Для каждого теста выведите для каждой вершины `FIRST`, `SECOND` или `DRAW` в зависимости от того, кто выиграет при оптимальной игре из этой вершины. Ответы к тестам разделяйте пустой строкой.

Примеры

retro.in	retro.out
5 5 1 2 2 3 3 1 1 4 4 5 2 1 1 2 4 4 1 2 2 3 3 1 1 4	DRAW DRAW DRAW FIRST SECOND FIRST SECOND FIRST FIRST SECOND SECOND

Задача C. 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 ($2 \leqslant n \leqslant 5\,000$) — 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

cruel.in	cruel.out
3	Schtirlitz 2
4	Mueller
5	Schtirlitz 1 3 5

Задача D. 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 получившихся 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 ($2 \leqslant N \leqslant 100\,000, 1 \leqslant R \leqslant N$). 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

`woodcut.in`	`woodcut.out`
5 5 2 3 1 3 2 5 4 5	1 1

Задача E. 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 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 10100000 31100000 00020000 00000000 00000000 00000000 00000000	-1