

Задача А. 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 — определить, 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 следующей строке вводятся N_2 и M_2 — количество вершин и рёбер во втором графе соответственно. Далее в следующих M_2 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

smith.in	smith.out
3 2	first
1 2	second
2 3	
2 1	
1 2	
2	
1 1	
3 2	

Задача В. 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 ($1 \leq n \leq 300\,000$) и m ($1 \leq m \leq 300\,000$). 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	DRAW
1 2	DRAW
2 3	DRAW
3 1	FIRST
1 4	SECOND
4 5	FIRST
2 1	SECOND
1 2	FIRST
4 4	FIRST
1 2	SECOND
2 3	SECOND
3 1	
1 4	

Задача С. Странная игра

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

Двое играют в простую игру на доске $n \times n$. У первого игрока есть одна белая фишка, а у второго — одна чёрная. Игроки ходят по очереди, первым ходит первый игрок (белые).

Первый игрок имеет право двигать свою фишку на одну клетку в одном из четырёх основных направлений (влево, вправо, вверх, вниз). Второй игрок при своем ходе также выбирает одно из этих четырёх направлений, но может передвинуть свою фишку как на одну клетку в этом направлении, так и на две. Выигрывает тот, кто первым съедает фишку соперника.

Определите победителя и число ходов, требуемое для победы, при оптимальной игре сторон.

Формат входных данных

Во входном файле даны пять чисел — n ($2 \leq n \leq 20$), а также координаты белой и чёрной фишек. Фишки стоят в разных клетках доски.

Формат выходных данных

Выведите `WHITE` x , если выигрывают белые, `BLACK` x , если выигрывают чёрные, `DRAW`, если игра закончится вничью. Здесь x — число ходов обеих сторон (полуходов) до момента окончания игры.

Примеры

<code>strange-game.in</code>	<code>strange-game.out</code>
2 1 1 2 2	BLACK 2
2 2 2 1 2	WHITE 1
3 1 1 3 3	BLACK 6

Задача D. 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 Е. 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 ($2 \leq N \leq 100\,000, 1 \leq R \leq N$). 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

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

Задача F. Шоколадка-революция (задача 3)

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

Аким получил золотую медаль на IOI, за что от правительства ему положена благодарность в виде шоколадки. На данный момент имеется N видов шоколадок, которые производит государственная шоколадная фабрика. У каждой шоколадки есть стоимость. Кроме того, у каждого вида, кроме шоколадки «Юлька», есть вид-прародитель. Известно, что прародителем может быть только шоколадка, которая была выпущена хронологически раньше. Исторически самый первый вид — «Юлька». Чиновник и Аким выбирают шоколадку Акиму в подарок. При этом цель первого — сэкономить, второго — получить настолько дорогую шоколадку, насколько это возможно. Начинается выбор с «Юльки». Аким за ход либо берет текущую шоколадку, либо меняет свой выбор на шоколадку-потомка (если это возможно). Чиновнику же кажется, что позже выпущенные виды хуже и дешевле, поэтому он меняет выбор на шоколадку-потомка (если это возможно), либо покупает Акиму текущую шоколадку (если потомков нет). Первым ходит Аким. Определите, на шоколадку какой стоимости может претендовать Аким.

Формат входных данных

Виды шоколадок занумерованы в некотором произвольном порядке, «Юлька» имеет номер 1. В первой строке входного файла находится одно число N — количество видов шоколадок ($0 < N \leq 200\,000$). Далее следуют N строк, в каждой из которых заданы два числа P_i и C_i — номер сорта-прародителя i -ого сорта и стоимость i -ого сорта ($0 \leq P_i \leq N$, $0 < C_i \leq 1\,000\,000\,000$; для «Юльки», у которой нет прародителя, указано $P_1 = 0$);

Формат выходных данных

Выведите в выходной файл одно число — стоимость шоколадки, которую получит Аким при правильных действиях как чиновника, так и своих.

Примеры

chocorev.in	chocorev.out
8 0 4 5 3 1 2 5 1 1 5 4 8 3 6 3 7	6

Задача G. Игры на графе

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

Противник начал e2-e4. Я
проанализировал его архитектуру и сдался

Из мемуаров 20-го чемпиона мира Фрица
Рыбкина

Прибыв на место, Ааз тут же потребовал организовать совещание букмекеров, на котором он изложит свой план.

— Главная задача, — начал Ааз своё выступление перед букмекерами, — научиться использовать достижения прогресса. Мы планируем запуск множества новых видов соревнований, что — вполне возможно — приведёт к тому, что появятся какие-то игры по правилам, придуманным не нами. А значит, необходимо уметь быстро выяснять, насколько эти правила могут быть нам полезны.

— А можно ли хотя бы в общем пояснить, как это будет делаться? — последовал вопрос из зала.

— Вот пример задачи, решив которую, мы сможем разобраться с целым классом игр. Дан ориентированный граф некоторой игры для двух игроков и начальная позиция в ней. Напомним, что в игре на графе игрок имеет право походить из позиции в любую позицию, в которую есть рёбро из текущей. Игроки ходят по очереди; проигрывает тот, кто не может сделать ход. Требуется проверить, верно ли, что при любой игре сторон всегда выигрывает первый игрок.

Формат входных данных

Во входном файле содержится описание одного или нескольких тестов. В первой строке каждого теста заданы число вершин V и число рёбер E ($1 \leq V \leq 100\,000$, $1 \leq E \leq 100\,000$), а также номер начальной позиции a ($1 \leq a \leq V$). Далее следуют E строк — описания рёбер в формате $u_i v_i$ ($1 \leq u_i, v_i \leq V$), что означает наличие ребра, направленного из вершины u_i в вершину v_i . Файл завершается тремя нулями. Сумма всех E по всем тестам не превосходит 100 000, количество тестов в файле не превосходит 1000.

Формат выходных данных

Следуйте формату примера максимально точно — проверка производится автоматически.

Примеры

gg.in	
3 2 1	
1 2	
1 3	
1 1 1	
1 1	
0 0 0	
gg.out	
First player always wins in game 1.	
Players can avoid first player winning in game 2.	