

## Задача А. Ретроанализ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.3 секунд
Ограничение по памяти:	256 мегабайт

Дан ориентированный весёлый граф из  $n$  вершин и  $m$  ребер. Двое 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.

### Пример

стандартный ввод	стандартный вывод
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 В. Огромный ним

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Петя и Вася играют в ним, но не простой, а просто огромный. У них есть очень много кучек камней. Кучки разделены на  $n$  групп. Группа  $i$  состоит из кучек размеров от  $l_i$  до  $r_i$  включительно. Помогите ребятам понять, кто выиграет при оптимальной игре.

### Формат входных данных

Первая строка содержит число  $n$  ( $1 \leq n \leq 10^5$ ).

Следующие  $n$  строк содержат пары чисел  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq 10^{18}$ ).

### Формат выходных данных

Если первый игрок проигрывает, выведите `Lose`, если выигрывает — выведите в первой строке `Win`, а во второй строке — любой выигрышный ход для первого игрока. Ход задается размером кучки до хода и после него.

### Примеры

стандартный ввод	стандартный вывод
1	Win
1 10	8 3
2	Lose
2 5	
2 5	

## Задача С. Дровосек

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.25 секунд  
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 находятся 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 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  $N - 1$ ).

### Пример

стандартный ввод	стандартный вывод
5 5	1
2 3	1
1 3	
2 5	
4 5	

## Задача D. Операции с простыми возвращаются

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.3 секунд  
Ограничение по памяти: 256 мегабайт

Дэвид Блейн и Вася сидят на уроке математики и скучают. Через некоторое время Дэвид Блейн предлагает поиграть в следующую игру:

- он записывает на листочке некоторое большое целое число и ходит первым;
- игроки ходят по очереди;
- на каждом ходу игрок обязан разделить текущее число на его простой делитель  $p$ . На этом же ходу игрок может умножить (а может и нет) результат деления на простое число  $q$  ( $1 < q < p$ );
- проигрывает тот, кто не может сделать ход.

Мягко говоря, Вася не очень доверяет Дэвиду Блейну и боится, что тот выписывает только проигрышные для Васи начальные числа. Помогите ему определить, так это или нет.

### Формат входных данных

Первая строка входного файла содержит единственное число  $n$  ( $1 \leq n \leq 10^{12}$ ).

### Формат выходных данных

В выходной файл выведите `Vasya`, если у Васи есть выигрышная стратегия, независимо от ходов Дэвида Блейна. Иначе выведите `David`.

### Пример

стандартный ввод	стандартный вывод
16	Vasya

## Задача Е. Вариация нима

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.25 секунд  
Ограничение по памяти: 256 мегабайт

На столе лежат  $n$  кучек камней:  $a_1$  камней в первой кучке,  $a_2$  камней во второй,  $\dots$ ,  $a_n$  в  $n$ -ой. Двое играют в игру, делая ходы по очереди. За один ход игрок может либо взять произвольное ненулевое количество камней (возможно, все) из одной любой кучки, либо произвольным образом разделить любую существующую кучку, в которой не меньше двух камней, на две непустые кучки. Проигрывает тот, кто не может сделать ход. Кто выигрывает при правильной игре?

### Формат входных данных

В первой строке задано целое число  $t$  — количество тестов ( $1 \leq t \leq 100$ ). Следующие  $t$  строк содержат сами тесты. Каждая из них начинается с целого числа  $n$  — количества кучек ( $1 \leq n \leq 100$ ). Далее следует  $n$  целых чисел  $a_1, a_2, \dots, a_n$  через пробел — количество камней в кучках ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите  $t$  строк; в  $i$ -ой строке выведите “FIRST”, если в  $i$ -ом тесте при правильной игре выигрывает первый игрок, и “SECOND”, если второй.

### Пример

стандартный ввод	стандартный вывод
3	FIRST
1 1	SECOND
2 1 1	FIRST
3 1 2 3	

## Задача F. Бинарная игра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Искандер и Оля любят придумывать ребусы. Но больше, чем придумывать ребусы, они любят придумывать какие-нибудь игры на строках. Вот и сейчас им в голову пришла забавная игра со следующими правилами:

- Выбирается какой-то набор *запрещённых* двоичных (состоящих из нулей и единиц) строк  $f_1, f_2, \dots, f_n$ .
- Выбирается некоторая стартовая бинарная строка  $s$ , такая что ни одна из запрещённых строк не входит в неё как подстрока.
- Игроки по очереди дописывают в конец строки  $s$  по одному символу «0» или «1». Оля ходит первой.
- Проигрывает тот, после чьего хода хотя бы одна из запрещённых строк  $f_1, f_2, \dots, f_n$  входит в  $s$  как подстрока.
- В случае если при оптимальной игре обоих игроков игра может продолжаться сколь угодно долго, то объявляется ничья.

Вы обожаете портить другим людям их любимые развлечения, поэтому решили написать программу, которая будет определять исход игры по заданному набору запрещённых строк и стартовой строке  $s$ .

### Формат входных данных

В первой строке входных данных записаны два целых числа  $n$  и  $m$  ( $0 \leq n \leq 100\,000$ ,  $0 \leq m \leq 1\,000\,000$ ) — количество запрещённых строк и изначальная длина строки  $s$ .

В каждой из последующих  $n$  строк содержится одна запрещённая строка. Гарантируется, что все эти строки непусты, состоят из символов «0» и «1» и никакая из них не является подстрокой строки  $s$ . Дополнительно гарантируется, что **суммарная длина** всех запрещённых строк не превосходит  $1\,000\,000$ .

В последней строке входных данных записана стартовая строка  $s$  длины  $m$ , состоящая только из символов «0» и «1». Обратите внимание, строка  $s$  может быть пустой, в этом случае соответствующая строка входных данных отсутствует (в том числе символ перевода строки). Длина  $s$  не превосходит  $1\,000\,000$ .

### Формат выходных данных

В зависимости от результата игры при оптимальной игре обоих игроков выведите:

- «Olya» (без кавычек), если Оля может победить вне зависимости от того как будет играть Искандер. Напомним, что Оля ходит первой.
- «Iskander» (без кавычек), если Искандер может победить не зависимо от ходов Оли.
- «Friendship» (без кавычек), если при оптимальной игре обоих игроков игра будет продолжаться бесконечно долго.

## Примеры

стандартный ввод	стандартный вывод
1 0 1	Friendship
3 1 000 001 011 0	Olya
2 3 1001 000 100	Iskander

## Замечание

Если вы не слушали вводную лекцию про анализ игр на ациклических и циклических графах, рекомендую первую главу данной статьи:

<https://ejudge.lksh.ru/archive/2014/07/A/games.pdf>

## Задача G. Сумма игр

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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.

### Пример

стандартный ввод	стандартный вывод
3 2	first
1 2	second
2 3	
2 1	
1 2	
2	
1 1	
3 2	



## Задача Н. Вас снова замяукали!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Два котёнка попали в запутанный лабиринт со множеством комнат и переходов между ними. Котят долго по нему плутали, обошли все комнаты по много раз, нашли выход (да даже и не один, а несколько), в общем, изучили там всё, что смогли. Теперь этот лабиринт котята используют в своих играх.

Чаще всего котята играют в следующую игру: начиная в какой-то комнате лабиринта, котята поочередно выбирают, в какую из комнат им перейти. Котята изначально находятся в одной комнате и ходят вместе. Как только котёнок, который должен выбрать следующую комнату, не может этого сделать, он признаётся проигравшим. Обычно в таких играх выигрывающий игрок стремится выиграть как можно быстрее, а проигрывающий стремится как можно дольше оттянуть свое поражение. Но у котят свои представления о победе и поражении. Если котёнок знает, что, начиная из текущей комнаты, он выиграет (вне зависимости от действий другого котёнка), то он стремится играть как можно дольше, чтобы продлить себе удовольствие от выигрыша (естественно, при этом выигрывающий котёнок должен гарантировать себе, что будет постоянно уверен в выигрыше). Котёнок, который знает, что проиграет (при условии, конечно, что другой котёнок будет действовать оптимально), старается проиграть как можно быстрее, чтобы начать новую игру, в которой и взять реванш.

Если котята будут ходить бесконечно долго, но никто из них не сможет выиграть, то котята считают игру завершившейся вничью и замяукивают Вас.

Вас попросили для каждой комнаты в лабиринте узнать, выиграет или проиграет котёнок, начинающий ходить из данной комнаты. Если котёнок, начинающий из этой комнаты, выигрывает, требуется узнать максимальное количество ходов, которое он сможет играть, если же проигрывает — минимальное количество, которое ему придётся играть.

### Формат входных данных

В первой строке ввода находятся два числа  $n$  и  $m$  — число комнат и переходов между комнатами в лабиринте ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 100\,000$ ). Далее следует  $m$  строк с описаниями переходов. Описание перехода состоит из двух чисел  $a$  и  $b$ , означающих, что котёнок, начинающий игру в комнате с номером  $a$ , может выбрать комнату  $b$  в качестве следующей.

### Формат выходных данных

Выведите  $n$  строк — для каждой комнаты результат игры для котёнка, который начнет игру из этой комнаты. Если игра закончится вничью, выведите «DRAW». Если начинающий котёнок выиграет, выведите «WIN  $K$ », где  $K$  — количество ходов, которые сможет играть выигрывающий котёнок. Если котёнок сможет играть сколь угодно долго, сохраняя возможность в любой момент выиграть, выведите «WIN INF». Если котёнок, начинающий из этой комнаты, проиграет, выведите «LOSE  $K$ », где  $K$  — количество ходов, которые придется играть проигрывающему котёнку. Если же котёнку придется играть сколь угодно долго, при том, что его соперник сможет в любой момент выиграть, выведите «LOSE INF».

## Примеры

стандартный ввод	стандартный вывод
4 4 1 2 1 3 2 4 3 4	LOSE 2 WIN 1 WIN 1 LOSE 0
6 6 1 2 2 3 3 4 4 1 4 5 5 6	DRAW DRAW DRAW DRAW WIN 1 LOSE 0
6 6 1 2 2 3 3 4 4 1 2 6 4 5	LOSE INF WIN INF LOSE INF WIN INF LOSE 0 LOSE 0

## Задача I. Игрушечный поезд

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Арезу и Борзу — близнецы. На день рождения им подарили восхитительную игрушечную железную дорогу, из которой они построили железнодорожную сеть, состоящую из  $n$  станций и  $m$  односторонних путей. Станции пронумерованы от 0 до  $n - 1$ . Каждый путь выходит из одной станции и приходит либо к той же станции, либо к некоторой другой. С каждой станции выходит хотя бы один путь.

Некоторые станции являются *зарядными*. Когда поезд прибывает на зарядную станцию, он полностью заряжается. Энергии полностью заряженного поезда хватает, чтобы проехать  $n$  путей без дополнительной зарядки. Иными словами, поезд полностью разряжается в момент перехода на  $(n+1)$ -й по счёту путь с момента последней зарядки.

На каждой станции имеется переключатель, который может указывать на любой из путей, выходящий от станции. Поезд может покинуть станцию только по пути, на который указывает переключатель.

Близнецы хотят сыграть в игру, используя свой поезд. Они поделили между собой все станции: каждая станция принадлежит либо Арезу, либо Борзу. У них есть один поезд. В начале игры полностью заряженный поезд находится на станции  $s$ . Игра начинается с того, что владелец станции  $s$  устанавливает переключатель на один из исходящих путей. После этого дети включают поезд и он начинает двигаться вдоль путей.

Когда поезд первый раз оказывается на некоторой станции, владелец станции устанавливает переключатель, расположенный на этой станции. Установленный переключатель остаётся в том же положении до конца игры. Таким образом, если поезд оказывается на станции, где он уже был, то он отправляется с неё по тому же пути, что и раньше.

Так как станций конечное количество, рано или поздно поезд попадёт в цикл. Цикл это такая последовательность различных станций  $c[0], c[1], \dots, c[k-1]$ , что со станции  $c[i]$  (для  $0 \leq i \leq k-2$ ) поезд выезжает по пути до станции  $c[i+1]$ , а со станции  $c[k-1]$  поезд выезжает по пути до станции  $c[0]$ . Обратите внимание, что цикл может состоять из единственной станции (то есть  $k = 1$ ), если поезд со станции  $c[0]$  выезжает по пути, ведущему обратно на станцию  $c[0]$ .

Арезу выигрывает игру, если поезд продолжает движение неограниченно долгое время, а Борзу выигрывает, если поезд рано или поздно разрядится. Иными словами, если среди станций  $c[0], c[1], \dots, c[k-1]$  есть хотя бы одна зарядная станция, то поезд сможет постоянно заряжаться и неограниченно долго перемещаться вдоль данного цикла, и Арезу выигрывает. В противном случае поезд разрядится (возможно, после некоторого количества полных перемещений вдоль цикла), и выигрывает Борзу.

Задано описание железнодорожной сети. Арезу и Борзу собираются провести  $n$  игр. В  $s$ -й игре для  $0 \leq s \leq n-1$  поезд исходно находится на станции  $s$ . Для каждой из игр требуется определить, есть ли у Арезу стратегия, гарантирующая ей выигрыш вне зависимости от ходов Борзу.

### Формат входных данных

В первой строке входных данных даны 2 числа  $n$  и  $m$  ( $1 \leq n \leq 5000$ ) — число станций и число путей.

В следующей строке даны  $n$  чисел  $a_0, a_1, \dots, a_{n-1}$ . Если  $i$ -я станция принадлежит Арезу, то  $a_i = 1$ . В противном случае станция принадлежит Борзу и  $a_i = 0$ .

В следующей строке даны  $n$  чисел  $r_0, r_1, \dots, r_{n-1}$ . Если  $i$ -я станция является зарядной, то  $r_i = 1$ . В противном случае станция принадлежит Борзу и  $r_i = 0$ .

В следующих  $m$  строках задано описание путей, в каждой строке по два числа  $u_i$  и  $v_i$  ( $0 \leq u_i, v_i \leq n$ ), и эти числа означают, что есть путь из станции  $u_i$  в станцию  $v_i$ .

### Формат выходных данных

В единственной строке выведите  $n$  чисел,  $w_0, w_1, \dots, w_n$ . Для каждого  $i$  значение  $w_i$  должно

равняться 1, если Арезу выигрывает в игре, начинающейся со станции  $i$ , вне зависимости от ходов Борзу. В противном случае значение  $w_i$  должно быть установлено в 0.

### Примеры

стандартный ввод	стандартный вывод
2 4 0 1 1 0 0 0 0 1 1 0 1 1	1 1
4 5 0 1 0 1 1 0 0 0 0 0 1 0 1 2 2 3 3 2	1 1 0 0

## Задача J. Игра на дереве

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Алиса и Боб играют в следующую игру:

- Боб рисует подвешенное дерево на  $n$  вершинах, занумерованных от 1 до  $n$ , с корнем в вершине 1.
- Алиса отвечает на  $q$  запросов. Для каждого запроса она берёт копию дерева и красит  $k$  его вершин в чёрный цвет, оставляя другие  $n - k$  вершин белыми.
- Цель игры — ответить на каждый запрос нахождения значений  $c_0, c_1, \dots, c_k$ , где  $c_i$  — это количество поддеревьев, содержащих ровно  $i$  чёрных вершин (включая вершину корня поддерева).

Помогите Алисе отвечать на такие запросы.

### Формат входных данных

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 3 \cdot 10^5$ ).

Каждая из следующих  $n - 1$  строк содержат два целых числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ), описывающие ребро дерева.

Следующая строка содержит целое число  $q$  ( $1 \leq q \leq 3 \cdot 10^5$ ).

Далее следует описание запросов. Каждый запрос описывается двумя строками. Первая строка содержит целое число  $k$  ( $1 \leq k \leq n$ ). Следующая строка содержит  $k$  различных номеров вершин, которые красятся в чёрный цвет.

Сумма  $k$  по всем запросам не превышает  $3 \cdot 10^5$ .

### Формат выходных данных

Для каждого запроса выведите  $k + 1$  число  $c_0, c_1, \dots, c_k$ .

## Примеры

стандартный ввод	стандартный вывод
7 1 2 1 3 1 4 2 5 3 7 3 6 2 3 5 7 2 3 7 6 5	2 3 1 1 1 4 1 1
7 2 1 1 3 4 1 7 4 3 5 3 6 3 3 5 2 1 4 7 6 2 4 2 3 1	3 3 0 1 1 4 1 0 1 5 1 1