

Задача А. Одна кучка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Я наверное снова закинул много слишком сложных задач, но все они от души.

Два игрока играют в игру. На столе лежит кучка из N камней. Двое ходят по очереди. За ход можно взять a_1, a_2, \dots, a_k камней. Проигрывает тот, кто не может сделать ход. Определите победителя!

Формат входных данных

В первой строке записано число k . Во второй строке k чисел — a_1, a_2, \dots, a_k . В третьей строке идет число m — количество различных N , для каждого из которых требуется определить победителя. В четвертой строке m чисел — N_1, N_2, \dots, N_m .

Ограничения: $1 \leq k \leq 20, m \leq 10^4, 1 \leq N_i, a_i \leq 10^6$.

Формат выходных данных

Выведите m строк, в каждой ответ на вопрос “кто выигрывает” — **First** или **Second**.

Пример

стандартный ввод	стандартный вывод
3	First
1 2 3	First
8	First
1 2 3 4 5 6 7 8	Second
	First
	First
	First
	Second

Задача В. Ретроанализ для маленьких

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

Пример

стандартный ввод	стандартный вывод
5 5	DRAW
1 2	DRAW
2 3	DRAW
3 1	FIRST
1 4	SECOND
4 5	
2 1	FIRST
1 2	SECOND
4 4	
1 2	FIRST
2 3	FIRST
3 1	SECOND
1 4	SECOND

Задача С. Вариация нима

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
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 D. Комфортабельная рассадка

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

В парке стоит скамейка на n мест. Изначально она пуста. Две команды играют в игру, по очереди *комфортабельно* сажая на скамейку по одному человеку. Человек садится *комфортабельно*, если на соседних местах ни справа, ни слева от него непосредственно никого нет. Проигрывает та команда, которая не может *комфортабельно* посадить очередного человека на своём ходу.

Дано n — количество мест на скамейке. Какая команда выигрывает при правильной игре?

Формат входных данных

В первой строке входного файла задано целое число n ($1 \leq n \leq 100$).

Формат выходных данных

В первой строке выходного файла выведите “FIRST”, если при правильной игре выигрывает первая команда, и “SECOND”, если вторая.

Примеры

стандартный ввод	стандартный вывод
2	FIRST
4	SECOND

Задача Е. 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 F. 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 котят используют в своих играх.

Чаще всего котят играют в следующую игру: начиная в какой-то комнате лабиринта, котят поочередно выбирают, в какую из комнат им перейти. Котят 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 котят будут ходить бесконечно долго, но никто из них не сможет выиграть, то котят 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 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. 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 I. Игры уголовников

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Случилось ужасное! Товарищ майор узнал, что вы репостнули мем про Иисуса, написали комментарий, оскорбляющий власть, лайкнули очередной видосик Навального, да и ещё и сходили на нанесанкционированный митинг! Теперь вы официально экстремист, а за это вам грозит тюремный срок на 100 лет!

В тюрьме вы сразу повстречали заядлого уголовника Антона, который предложил вам сыграть в особую тюремную игру уголовников. Игра заключается в следующем: есть 2 игрока, один — чётный, а другой — нечётный. В каждом раунде игры оба игрока показывают один или два пальца. Если a — сумма показанных пальцев, то если a чётно, то нечётный игрок платит чётному a монет, а если a нечётно, то чётный игрок платит нечётному a монет. Условия одинаковы, поэтому Антон предложил вам самим выбрать, чётным вы будете или нечётным. После этого каждый день своего десятилетнего заключения (всего $365 * 100 = 36500$ дней) вы сыграете с Антоном по партии. Ваша цель — не остаться в минусе и в конце своего заключения не потерять свои деньги после всех 36500 партий. Вам лениво каждый раз приходиться к Антону и играть с ним в эту игру, поэтому вы решили написать программу, играющую за вас.

Протокол взаимодействия

Это интерактивная задача.

В первой строке выведете 0, если вы хотите играть за чётного игрока, или 1, если вы хотите играть за нечётного.

После этого следующие 36500 ходов пройдут следующим образом. Выведите одно число 1 или 2 — количество пальцев, которое вы показываете в этом раунде. В ответ на это программа жюри, имитирующая Антона, выведет вам одно число 1 или 2 — количество пальцев, которое показывает Антон. Гарантируется, что в i -й ход выведенное интерактором число никак не зависит от того, какое число вы вывели в i -й ход, т.е. программа Антон играет честно.

После 36500 ходов программа жюри выведет ваш баланс — разницу полученных и отданных вами монет. В случае, если баланс неотрицательный, ваша программа будет считаться верной.

Все строки завершайте символом перевода строки и сбросом буфера ввода. Все числа программы жюри выводятся в новой строке. Для удобства в примере показана игра, состоящая из 4 партий.

Пример

стандартный ввод	стандартный вывод
	0
1	1
2	1
1	2
2	2
0	