

Задача А. Игра с окружностью

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

На листе бумаги нарисована окружность, на которой отмечено n различных точек. Гидран и Диореп играют в игру, делая ходы по очереди. Ход состоит в том, чтобы выбрать две различные отмеченные точки и провести между ними хорду; это можно сделать, только если новая хорда не имеет общих точек (в том числе и концов) со всеми ранее проведёнными хордами. Проигрывает тот, кто не может сделать ход. Первым ходит Гидран. Кто выигрывает при правильной игре?

Формат входных данных

На ввод подаётся несколько строк. Первая из них содержит целое число t — количество тестов ($1 \leq t \leq 50$). Каждая из следующих t строк описывает один тест. Описание теста состоит из одного целого числа n — количества отмеченных точек на окружности ($3 \leq n \leq 10^9$).

Формат выходных данных

Для каждого заданного n выведите на отдельной строке «53», если выигрывает Гидран, и «34», если выигрывает Диореп. Ответы должны быть выведены в том же порядке, в котором числа n заданы во вводе.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	53
4	34
5	

Задача В. Игра в разделение

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

Есть клетчатый прямоугольник размера $w \times h$ клеток. Алиса и Боб делают ходы по очереди; начинает Алиса. Каждый ход состоит из двух частей. Сначала один из имеющихся прямоугольников делится на две непустые части вертикальной или горизонтальной линией, проходящей по линиям сетки. Затем один из двух получившихся прямоугольников также делится на две непустые части линией, также проходящей по линиям сетки и перпендикулярной первой линии.

Проигрывает тот, кто не может сделать ход. Кто выигрывает при правильной игре? Если выигрывает Алиса, какой её ход приводит к выигрышу при дальнейшей правильной игре?

Формат входных данных

В первой строке ввода заданы через пробел два целых числа w и h — ширина и высота исходного прямоугольника ($1 \leq w, h \leq 100$).

Формат выходных данных

В первой строке выведите имя победителя: «Alice», если при правильной игре победит Алиса, и «Bob», если победу одержит Боб. Если победит Алиса, в следующих двух строках выведите ход Алисы, ведущий к дальнейшему выигрышу при правильной игре. В первой из этих строк выведите через пробел координаты концов первой линии, которую должна провести Алиса, а во второй — координаты концов второй линии. Для каждого конца линии сначала выводите координату по оси Ox , а затем — координату по оси Oy . Считайте, что левый нижний угол исходного прямоугольника имеет координаты $(0, 0)$, а правый верхний угол — координаты (w, h) .

Концы каждой линии можно выводить в любом порядке. Если правильных ходов несколько, можно вывести любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 2	Alice 0 1 4 1 3 0 3 1
3 3	Bob

Пояснения к примерам

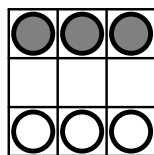
В первом примере Алиса любым возможным ходом разрезает исходный прямоугольник на три такие части, что в них невозможно сделать ещё один ход.

Во втором примере любой ход Алисы оставляет для дальнейших ходов ровно один квадрат размера 2×2 клетки. После этого Боб делит этот квадрат на три непустые части и заканчивает игру.

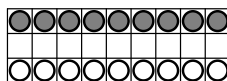
Задача С. Игра в пешки

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

В свободное время Дед Мороз Петрович и Дед Мороз Егорыч играют в следующую игру. На доске 3×3 пешки расставляются следующим образом:



Пешки ходят и бьют по обычным шахматным правилам, к которым добавляется ещё одно: бить обязательно. Проигрывает тот, кто не может сделать ход. Первыми ходят белые. В течение 100 лет Петрович играл белыми и всегда выигрывал. Однажды Егорычу это надоело, и он принёс доску 3×5 . Но и теперь он, играя чёрными, до сих пор проигрывает. «Как так?» — подумал Егорыч и решил купить доску $3 \times N$:



Вот тут Петровичу надо подумать, какими играть, чтобы выиграть. Вам нужно помочь ему это сделать.

Формат входных данных

Целое число N ($1 \leq N \leq 10^9$).

Формат выходных данных

«White», если Петровичу надо играть белыми, и «Black», если надо играть чёрными. Егорыч и Петрович каждый раз ходят по наилучшей для себя стратегии.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	White
4	Black
5	White

Задача D. Вариация Нима

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 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 -й строке выведите «FIRST», если в i -м тесте при правильной игре выигрывает первый игрок, и «SECOND», если второй.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	FIRST
1 1	SECOND
2 1 1	FIRST
3 1 2 3	

Задача Е. Игра на графе

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

Есть ориентированный граф. В одной из вершин графа стоит фишка. Двое — Саша и Александра — играют в игру. Нужно за один ход (его делают оба игрока сообща) сдвинуть фишку по одному из рёбер. Ход делается так: из вершины, в которой сейчас стоит фишка, выходят K рёбер. На доске записываются числа от 1 до K и игроки их по очереди вычёркивают. Первой всегда вычёркивает Саша. Последнее вычёркнутое число — номер ребра, по которому делается ход.

В этом графе есть особые вершины типа 1 — выигрышные для Саши. А ещё есть особые вершины типа 2 — выигрышные для Александры.

Как только фишка попадает в одну из таких вершин, игра заканчивается победой одного из двух игроков. Если игра продолжается бесконечно, её называют ничейной.

Вам дан граф и начальное положение фишки. Кто выиграет при оптимальной игре обеих сторон?

Формат входных данных

В первой строке ввода заданы числа N, M, S, K_1, K_2 — число вершин и рёбер в графе, начальное положение фишки, число вершин первого типа и число вершин второго типа. Вторая строка содержит K_1 чисел от 1 до N — вершины первого типа. В третьей строке находится K_2 чисел от 1 до N — вершины второго типа. Все $K_1 + K_2$ чисел различны.

Следующие M строк содержат рёбра. Ребро из вершины a в вершину b задаётся парой чисел (a, b) .

Ограничения:

- $1 \leq n, m \leq 100\,000$.
- Из всех вершин, кроме выигрышных, выходит хотя бы по одному ребру.

Формат выходных данных

Исход игры — `Sasha wins`, или `Alexandra wins`, или `Draw`.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 4 1 1 0 2 1 1 1 2 2 1 2 2	Sasha wins
3 9 1 1 1 2 3 1 1 1 2 1 3 2 1 2 2 2 3 3 1 3 2 3 3	Draw

Задача F. Игры на графе

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 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 ребро 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 входных данных 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 . Ввод завершается тремя нулями. Сумма всех E по всем тестам не превосходит 100 000, количество тестов во входных данных не пре-

восходит 1000.

Формат выходных данных

Следуйте формату примера максимально точно — проверка производится автоматически.

Пример

<i>стандартный ввод</i>
3 2 1 1 2 1 3 1 1 1 1 1 0 0 0
<i>стандартный вывод</i>
First player always wins in game 1. Players can avoid first player winning in game 2.

Задача G. Игра с шестиугольниками

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

Рассмотрим полосу высоты 3, состоящую из правильных шестиугольников. Верхний и нижний ряды полосы состоят из n шестиугольников, а средний ряд содержит $n + 1$ шестиугольник.

Два игрока по очереди ставят одинаковые отметки в шестиугольники. В каждом шестиугольнике может стоять не более одной отметки. Выигрывает тот, после чьего хода оказываются отмечены три шестиугольника такие, что один из них — соседний по стороне с двумя другими, а два отрезка, проведённые из его центра в центры этих двух соседей, образуют угол в 120 градусов.

На рисунке слева показан пример полосы для $n = 6$, а на рисунке справа — пример трёх отметок, образующих выигрышную комбинацию шестиугольников.



Дана позиция в этой игре, в которой никто ещё не выиграл. Кто выигрывает при правильной игре, начинающейся с этой позиции: первый игрок или второй? Первым игроком считается тот, кто делает из данной позиции первый ход. Выражение «при правильной игре» можно определить рекурсивно: если игрок может сделать такой ход, чтобы либо сразу выиграть, либо после любого ответного хода противника по-прежнему иметь возможность выиграть при правильной игре, он обязательно сделает один из таких ходов. Если же такого хода не существует, игрок может сделать любой из возможных ходов.

Формат входных данных

В первой строке входного файла задано целое число t — количество тестов ($1 \leq t \leq 10$). Далее записаны сами тесты. Каждый тест представляет собой позицию в игре и задаётся четырьмя строками. В первой из этих строк записано целое число n — количество шестиугольников в верхнем и нижнем рядах полосы. ($1 \leq n \leq 5000$). Напомним, что в таком случае средний ряд

содержит $n + 1$ шестиугольник. Следующие три строки задают сами полосы: верхнюю, нижнюю и среднюю. Символ «.» (точка) соответствует пустому шестиугольнику, а «#» (решётка) — отмеченному. Гарантируется, что в заданной позиции никто ещё не выиграл.

Формат выходных данных

В ответ на каждый из t тестов выведите на отдельной строке слово «FIRST», если из позиции, заданной в этом тесте, выигрывает первый игрок, или же слово «SECOND», если из этой позиции выигрывает второй игрок.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>иллюстрация</i>
3	FIRST	
2	FIRST	
..	SECOND	
...		
..		
3		
#.#		
...#		
##.		
2		
..		
.#.		
..		
..		
..		

Пояснение к примеру

В первом тесте первый игрок может поставить отметку в центральный шестиугольник и заставить второго играть с позиции, представленной в третьем тесте.

Во втором тесте у первого игрока существует несколько способов выиграть первым же ходом.

В третьем тесте в ответ на любой из шести ходов существует два различных хода, несущих победу второму игроку.

Задача Н. Магараджи возвращаются домой

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

Джек и Джилл играют в игру на бесконечной вверх и вправо шахматной доске. Строки и столбцы пронумерованы натуральными числами от 1 до бесконечности. В начальной позиции на доске находятся k камней. На одной клетке могут располагаться несколько камней одновременно. За один ход игрок передвигает один камень на одну или несколько клеток вниз, влево, по диагонали (на одно и то же количество клеток вниз и влево) или ходом коня, если при этом номер столбца и номер строки строго уменьшаются (на одну клетку вниз и на две влево, либо на две клетки вниз и на одну влево). Камень не должен покидать доску, но может перепрыгивать другие камни и перемещаться на клетки, где находятся другие камни. Игрок, который не может сделать ход, проигрывает. Джек и Джилл ходят по очереди, Джек начинает игру.

Помогите Джеку найти выигрышный ход или определите, что у Джилл существует выигрышная стратегия.

Формат входных данных

В первой строке задано целое число t — количество тестовых случаев ($1 \leq t \leq 100$).

Каждый тестовый случай занимает несколько строк. В первой из них задано целое число k — количество камней в начальной позиции ($1 \leq k \leq 10$). В следующих k строках заданы позиции камней, по одной на строке. Позиция камня задана двумя целыми числами r и c — номер строки и номер столбца соответственно ($1 \leq r, c \leq 2000$).

Формат выходных данных

В ответ на каждый тестовый случай выведите три числа i , r и c , разделённые пробелами и означающие, что Джек может победить при правильной игре, если первым ходом передвинет i -й камень на клетку (r, c) . Камни пронумерованы начиная с 1.

Если решений несколько, выведите лексикографически наименьшее: решение с наименьшим номером камня, в случае неоднозначности — с наименьшим номером строки, в случае неоднозначности — с наименьшим номером столбца.

Если в начальной позиции у Джека нет выигрышной стратегии, выведите «-1 -1 -1» (без кавычек).

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	3 1 1
5	-1 -1 -1
2 3	2 1 1
3 2	
3 3	
3 3	
3 3	
1	
2 4	
2	
1 1	
3 2	

Задача I. Функция Смита

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

Знаете, что такое функция Гранди?

А знаете, что такое функция Смита? Это примерно как функция Гранди, но для **ориентированных графов с циклами**.

Далее описан алгоритм вычисления функции Смита для ориентированного графа. Реализуйте его эффективно.

Алгоритм:

1. Вначале значение в каждой вершине не определено.
2. Пока значения как-то меняются, повторяйте следующие шаги.
3. **Для** каждой вершины v вычислите $m(v) = \text{mex}(\{f_1, f_2, \dots, f_k\})$.
Здесь f_1, f_2, \dots, f_k — значения в детях вершины v .
Число $\text{mex}(F)$ — это минимальное неотрицательное целое число, не встречающееся во множестве F .
4. **Если** есть вершина v , для всех дуг (v, x) из которой значение в вершине x уже известно **или** из x есть ребро в вершину со значением $m(v)$, **то** значение в вершине v устанавливается равным $m(v)$.
- 5.

Определим значения функции Смита как результат выполнения этого алгоритма. Если для какой-то вершины v значение осталось не определённым, будем обозначать это значение числом -1 .

Заметим, что алгоритм лишь определяет значения — ваша реализация может вычислять их любым другим способом, лишь бы значения получились те же. Также заметим, что определение корректно: алгоритм завершит работу за конечное количество шагов, а результат не зависит от порядка рассмотрения вершин.

Формат входных данных

В первой строке заданы целые числа N и M — количество вершин и рёбер в графе ($1 \leq N \leq 100\,000$, $0 \leq M \leq 100\,000$). Каждая из следующих M строк содержит два числа x и y , означающих, что в графе есть дуга из x в y ($1 \leq x, y \leq N$).

Формат выходных данных

Выведите N целых чисел — значения функции Смита для вершин $1, 2, \dots, N$.

Пример

<i>стандартный ввод</i>		<i>стандартный вывод</i>
5	5	0 2 0 1 -1
2	1	
3	2	
4	3	
2	4	
5	5	

Задача J. Peep Blue

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

Суперкомпьютер Peep Blue умеет играть в игры. Однако Серёжа не побоялся принять его вызов.

Правила, предложенные компьютером, таковы. Есть n кучек камней. Игроки ходят по очереди, начинает Серёжа. На первом ходу каждому из игроков разрешается убрать любой набор кучек (в том числе пустой, но не все). На втором и следующих ходах каждый из игроков может взять любое ненулевое число камней из любой кучки. Проигрывает тот, кто не может сделать ход.

Ваша задача — определить, может ли Серёжа одержать победу, а также какой первый ход ему следует сделать. Среди всех первых ходов, приносящих победу, выберите такой, который оставит в игре как можно больше камней: это должно усложнить задачу Peep Blue.

Формат входных данных

В первой строке ввода записано целое число n — количество кучек камней ($1 \leq n \leq 100\,000$). В следующей строке записаны n натуральных чисел, a_i , не превышающих $2^{30} - 1$, — количества камней в кучках.

Формат выходных данных

Если Серёжа не может одержать победу, выведите единственное число -1 . Иначе выведите максимально возможное суммарное число камней, которое можно оставить.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 1 2 3 4	9