Содержание

В некоторых задачах большой ввод и вывод. Имеет смысл пользоваться супер быстрым вводом-выводом:
`http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write.cpp.html`

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу:
`http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html`

Обязательные задачи

Задача A. Транзитивное замыкание [0.4 sec, 256 mb]

Дан ориентированный граф. Найдите его транзитивное замыкание, то есть для каждой пары вершин a, b определите, есть ли путь из a в b.

Формат входных данных

На первой строке число вершин n ($1 \leqslant n \leqslant 1\,000$). Следующие n строк имеют длину n, состоят из нулей и единиц и задают матрицу смежности графа. Единица в i-й строке, j-м столбце обозначает ребро из i в j.

Формат выходных данных

Выведите матрицу смежности транзитивного замыкания данного графа.

Примеры

floyd32.in	floyd32.out
3 010 001 000	011 001 000

Задача B. Одна кучка [0.3 sec, 256 mb]

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

Формат входных данных

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

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

Формат выходных данных

Выведите m строк, в каждой ответ на вопрос "кто выиграет" — `First` или `Second`.

Пример

heaps2.in	heaps2.out
3 1 2 3 8 1 2 3 4 5 6 7 8	First First First Second First First First Second

Задача C. Две кучки [0.5 sec, 256 mb]

Два игрока играют в игру. На столе лежат две кучки: в первой a камней, во второй — b. Игроки ходят по очереди. Каждым ходом игрок выбирает одну кучку и берет какое-то количество камней из нее. Первый игрок может брать $a_1, a_2, \ldots, a_k$ камней, второй — $b_1, b_2, \ldots, b_l$. Проигрывает тот, кто не может сделать ход. Определите победителя!

Формат входных данных

В первой строке записаны a и b. Во второй строке записаны k и последовательность a_i, на третьей — l и b_i. $1 \leqslant a, b \leqslant 1\,000$, $1 \leqslant k, l, \leqslant 10$, $1 \leqslant a_i, b_j \leqslant 1\,000$.

Формат выходных данных

Если выигрывает первый игрок, выведите `First`. Иначе выведите `Second`.

Пример

heaps.in	heaps.out
2 2 2 1 2 1 1	First
2 2 1 1 2 1 2	Second

Задача D. Произведение графов [0.3 sec, 256 mb]

Пусть дан ориентированный ациклический 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, прямое произведение двух игр на графах. Прямое произведение игр — это следующая игра: изначально на каждом графе в начальной позиции стоит по фишке. За ход игрок двигает обе фишки по рёбрам (каждую фишку двигает в собственном графе). Проигрывает тот, кто не может сделать ход. То есть тот, кто не может сделать ход хотя бы в одной игре.

Ваша задача — опеределить, 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 ($1 \leqslant N_1, M_1 \leqslant 100\,000$). На следующих M_1 строках содержится по два числа x и y ($1 \leqslant x, y \leqslant N_1$).

В следующих $M_2 + 1$ 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 ($1 \leqslant v_1 \leqslant N_1$, $1 \leqslant v_2 \leqslant N_2$).

Учтите, что в графах могут быть кратные рёбра.

Формат выходных данных

На каждую из T пар начальных вершин выведите строку “`first`”, если при правильной игре выиграет первый, и “`second`”, если второй.

Пример

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

Задача E. Ретроанализ для маленьких [0.6 sec, 256 mb]

Дан ориентированный весёлый граф из n вершин и m ребер. Оля и Коля 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 та: 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 F. Choco. Шоколадка [0.3 sec, 256 mb]

Двое играют в такую игру: перед ними лежит шоколадка размера $N \times M$. Игроки ходят по очереди. За ход разломить любой имеющийся кусок шоколадки на 2 «непустых» куска, при этом запрещено ломать куски размером не больше, чем $1 \times S$ (т.е. нельзя ломать куски, у которых один размер равен 1, а другой не превосходит S), куски можно поворачивать. Ломать, конечно, можно только вдоль линий, нанесенных на шоколадке, т.е. после разлома должны получаться два прямоугольника с целочисленными ненулевыми сторонами.

Проигрывает тот, кто не может сделать хода.

Формат входных данных

Во входном файле находятся три целых числа N, M и S ($0 < N, M, S \leq 100$).

Формат выходных данных

Выведите в выходной файл одно число 1 или 2 — номер игрока, который выигрывает при правильной игре.

Пример

choco.in	choco.out
2 2 1	1

Задача G. Жестокая задача [0.3 sec, 256 mb]

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.

Пример

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

Задача H. Малыш и Карлсон [0.3 sec, 256 mb]

На свой День рождения Малыш позвал своего лучшего друга Карлсона. Мама испекла его любимый пирог прямоугольной формы $a \times b \times c$ сантиметров. Карлсон знает, что у Малыша еще есть килограмм колбасы. Чтобы заполучить ее, он предложил поиграть следующим образом: они по очереди разрезают пирог на две ненулевые по объему прямоугольные части с целыми измерениями и съедают меньшую часть (в случае, когда части равные, можно съесть любую). Проигрывает тот, кто не может сделать хода (то есть когда размеры будут $1 \times 1 \times 1$). Естественно, победителю достается колбаса.

Малыш настаивает на том, чтобы он ходил вторым.

Помогите Карлсону выяснить, сможет ли он выиграть, и если сможет — какой должен быть его первый ход для этого.

Считается, что Малыш всегда ходит оптимально.

Формат входных данных

Во входном файле содержится 3 целых числа a, b, c ($1 \leqslant a, b, c \leqslant 5\,000$) — размеры пирога.

Формат выходных данных

В случае, если Карлсон не сможет выиграть у Малыша, выведите `NO`. В противном случае в первой строке выведите `YES`, во второй — размеры пирога после первого хода Карлсона в том же порядке, что и во входном файле.

Примеры

karlsson.in	karlsson.out
1 1 1	NO
2 1 1	YES 1 1 1

Бонус

Задача I. Игры на графе [0.3 sec, 256 mb]

> 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 входном файле содержится описание одного или нескольких тестов. В первой строке каждого теста заданы число вершин V и число рёбер E ($1 \leqslant V \leqslant 100\,000$, $1 \leqslant E \leqslant 100\,000$), а также номер начальной позиции a ($1 \leqslant a \leqslant V$). Далее следуют E строк — описания рёбер в формате u_i v_i ($1 \leqslant u_i, v_i \leqslant V$), 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.

Пример

gg.in
3 2 1 1 2 1 3 1 1 1 1 1 4 3 1 1 2 1 3 3 4 0 0 0
gg.out
First player always wins in game 1. Players can avoid first player winning in game 2. Players can avoid first player winning in game 3.

Задача J. Conway [4 sec, 256 mb]

В комнате есть M лампочек и N переключателей. Для удобства гарантируется, что N **нечётное**. Каждая лампочка имеет собственную мощность p_i и может быть только в двух состояниях: полностью гореть, или полностью не гореть. Если лампочка включена, освещённость в комнате увеличивается на p_i единиц.

Каждый переключатель соединён с некоторым множеством лампочек. Смена состояния переключателя меняет состояние всех лампочек, соединённых с ним. Никаких ограничений на соединения нет, это значит, что любой переключатель может быть соединён с любым множеством лампочек, а каждая лампочка может быть соединёна с любым множеством переключателей.

Джон изобрёл новую игру и пригласил своих друзей Роланда и Патрика сыграть разок. Роланд любит свет и пытается к концу игры сделать комнату максимально освещённой. У Патрика ровно обратная цель – уменьшить освещённость комнаты настолько, насколько возможно.

Джон выбирает величину K, чтобы определить победителя. В конце игры, если суммарная мощность всех включённых лампочек хотя бы K, побеждает Роланд, иначе Патрик. Игра устроена следующим образом.

- Переключатели занумерованы числами от 1 до N.
- Игроки делают ровно по $\frac{N-1}{2}$ ходов. Ходят они по очереди. Первым ходит Роланд.
- Когда Роланд совершает свой i-й ход, он может использовать переключатели с номерами $2 \cdot i-1$ и $2 \cdot i$. Это первый и второй переключатели на его первом ходу, третий и четвёртый переключатели на его втором ходу, и так далее. Заметьте, что Роланд на каждом ходу может выбрать любое из 4 подмножеств доступных переключателей (какой-то один, оба, ни одного). Если какая-то лампочка подключена сразу к двум переключателям, которые использует на своём ходу Роланд, лампочка меняет своё состояние дважды.
- Правила для Патрика абсолютно такие же, за исключением индексов переключателей, которыми он может пользоваться. На своём i-м ходу он может пользоваться переключателями с номерами $2 \cdot i$ и $2 \cdot i+1$. Это второй и третий на его первом ходу, четвёртый и пятый на его втором ходу, и так далее.

Джону нравится смотреть за игрой своих друзей. Особенно если он заранее знает результат игры при оптимальных действиях обоих игроков. Он просит вас написать программу, которая определит победителя в каждой из T игр. Следует предполагать, что и Роланд, и Патрик играют оптимально.

Формат входных данных

Первая строка входных данных содержит целое число T — количество тестовых примеров ($1 \leqslant T \leqslant 5$). Далее каждый тестовый пример описывает отдельную игру следующим образом. Сперва строка с тремя целыми числами N, M и K ($3 \leq N \leq 33$, N нечётно, $1 \leq M \leq 32$, $0 \leq K \leq 2 \cdot 10^9$) — количество переключателей, количество лампочек, уровень освещённости для определения победителя. На следующей строке идут M чисел p_i ($1 \leq p_i \leq 5 \cdot 10^7$) — мощности лампочек. Далее N строк описывают соединения между переключателями и лампочками. Каждая из строк содержит M символов. Если i-й переключатель соединён с j-й лампочкой, то j-й символ i-й строки равен '`1`', иначе символ равен '`0`'.

Формат выходных данных

Для каждого тестового примера на отдельной строке выведите имя победителя – или "`Roland`", или "`Patrick`".

Примеры

conway.in	conway.out
2 3 2 10 10 10 01 00 11 3 5 1 1 2 3 4 5 01011 11000 10011	Roland Patrick

Замечание

В первой игре у Роланда есть следующая оптимальная стратегия: на своём первом ходу он использует и первый, и второй переключатель. Рассмотрим любой возможный ход Патрика: к концу игры будет ровно одна лампочка, которая останется включённой, поэтому Роланд по любому выиграет.

Во второй игре, не зависимо от хода Роланда, Патрик может выключить все лампочки.

Задача K. Вас снова замяукали! [0.3 sec, 256 mb]

Два котёнка попали в запутанный лабиринт со множеством комнат и переходов между ними. Котята 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 поочерёдно 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 ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$). 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

labyrinth.in	labyrinth.in
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 L. Вариация Нима [0.3 sec, 256 mb]

На столе лежат n кучек камней: a_1 камней в первой кучке, a_2 камней во второй, ..., 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 ($1 \leqslant t \leqslant 100$). Следующие t строк содержат сами тесты. Каждая из них начинается с целого числа n — количества кучек ($1 \leqslant n \leqslant 100$). Далее следует n целых чисел a_1, a_2, ..., a_n через пробел — количество камней в кучках ($1 \leqslant a_i \leqslant 10^9$).

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

varnim.in	varnim.out
3 1 1 2 1 1 3 1 2 3	FIRST SECOND FIRST

Задача M. Монетки [0.3 sec, 256 mb]

На столе лежат в ряд N кучек монеток. В i-й кучке лежит ровно a_i монеток. При этом оказалось, что $a_i \leqslant a_j$ при $i < j$.

Катя и Серёжа играют с этими монетками в игру. За ход можно взять любое ненулевое количество монеток из любой кучки, но условие ($a_i \leqslant a_j$ при $i < j$) должно сохраниться. Выигрывает, как обычно, взявший последнюю монету.

Вам требуется определить, кто выйграет при правильной игре. Серёжа ходит первый.

Формат входных данных

В первой строке задано число N ($1 \leqslant N \leqslant 100\,000$). В следующей строке задано N чисел — $a_1, a_2, \ldots, a_N$ ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Если выйграет Серёжа, выведите «Sergey», иначе выведите «Katya».

Пример

coins.in	coins.out
1 10	Sergey
2 10 15	Sergey
2 10 10	Katya

Задача N. Битва за кольцо [0.3 sec, 256 mb]

Саруман Белый и Гэндальф Серый решили сыграть в игру. Победителю достаётся Кольцо Всевластия. Перед игроками лежат кольца, соединённые в K цепочек. Для каждого кольца известно содержание золота в нём в процентах – целое число от 1 до 100. Ходят по очереди. За ход разрешается выбрать одну из цепочек и какое-то кольцо из этой цепочки и дематериализовать все кольца из данной цепочки с процентным содержанием золота не больше, чем у выбранного. При этом, понятно, цепочка может распасться на несколько. Игра продолжается на оставшихся цепочках. Тот, кто дематериализовал последнее кольцо, выиграл. Первым ходит Гэндальф. Определите, может ли Гэндальф выиграть и, если может, какой первый ход он должен для этого сделать.

Формат входных данных

В первой строке дано целое число K ($1 \leqslant K \leqslant 50$). В следующих K строках приведены описания цепочек в следующем формате: сперва дана длина цепочки – целое число от 1 до 100, затем – процентные содержания золота в кольцах цепочки. Числа в строке разделены пробелом.

Формат выходных данных

Выведите “S”, если Кольцо Всевластия достанется Саруману. В противном случае выведите в первой строке “G”, а во второй пару чисел, описывающих выигрышный первый ход Гэндальфа – номер цепочки и номер кольца в ней. Цепочки и кольца внутри цепочек нумеруются с 1. Если существует несколько выигрышных первых ходов, выведите ход с наименьшим номером цепочки, если и таких несколько – с наименьшим номером кольца.

Примеры

rings.in	rings.out
2 3 1 2 1 1 1	G 1 1
2 3 2 1 2 1 1	S