Содержание

У вас не получается читать/выводить данные, открывать файлы?
Воспользуйтесь примерами (c++) (python).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь быстрым вводом-выводом.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу.

Обратите внимание на `GNU C++` компиляторы с суффиксом `inc`, они позволяют пользоваться дополнительной библиотекой. Под ними можно сдать вот это.

Must have

Задача 12A. 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

stdin	stdout
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 12B. 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

stdin	stdout
3	Schtirlitz 2
4	Mueller
5	Schtirlitz 1 3 5

Обязательные задачи

Задача 12C. 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

stdin	stdout
3 1 2 3 8 1 2 3 4 5 6 7 8	First First First Second First First First Second

Задача 12D. 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

stdin	stdout
2 2 2 1 2 1 1	First
2 2 1 1 2 1 2	Second

Задача 12E. 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

stdin	stdout
3 2 1 2 2 3 2 1 1 2 2 1 1 3 2	first second

Задача 12F. 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

stdin	stdout
2 2 1	1

Дополнительные задачи

Задача 12G. 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

stdin
3 2 1 1 2 1 3 1 1 1 1 1 4 3 1 1 2 1 3 3 4 0 0 0
stdout
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 12H. 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

stdin	stdout
1 10	Sergey
2 10 15	Sergey
2 10 10	Katya

Задача 12I. 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

stdin	stdout
2 3 1 2 1 1 1	G 1 1
2 3 2 1 2 1 1	S