

Задача А. Ферзя в угол

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

64 мегабайта

В левом нижнем углу доски $M \times N$ стоит ферзь. Двое игроков по очереди ходят ферзем, перемещая его на любое число клеток по вертикали вверх, по горизонтали вправо, или по диагонали вправо-вверх. Выигрывает тот, кто поставит ферзя в правый верхний угол доски. Определите, какой из игроков имеет выигрышную стратегию. Гарантируется, что нужно сделать хотя бы один ход.

Формат входных данных

На вход программе подается два натуральных числа M и N , не превосходящих 100.

Формат выходных данных

Программа должна вывести номер игрока (1 или 2), который имеет выигрышную стратегию.

Пример

стандартный ввод	стандартный вывод
3 4	1

Задача В. Функция Гранди

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

Дан ориентированный ациклический граф. Посчитайте функцию Гранди для каждой стартовой вершины.

Формат входных данных

На первой строке будут даны числа n и m — количество вершин и рёбер в графе ($1 \leq n, m \leq 100\,000$). На следующих m строках содержится по два числа x и y ($1 \leq x, y \leq n$).

Учтите, что в графе могут быть кратные рёбра.

Формат выходных данных

Выведите n чисел — значение функции Гранди для каждой стартовой вершины.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 2 3 1 3	2 1 0
2 1 2 1	0 1

Задача С. Малыш и Карлсон

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

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 ($1 \leq a, b, c \leq 5000$) — размеры пирога.

Формат выходных данных

В случае, если Карлсон не сможет выиграть в 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

стандартный ввод	стандартный вывод
1 1 1	NO
1 2 1	YES 1 1 1
1 1 10	YES 1 1 7

Задача D. Огромный ним

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

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 входного файла 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 1 10	Win 8 3
2 2 5 2 5	Lose

Задача Е. 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 F. Королевская игра

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

Император Шардонии и король Флатляндии во время своей встречи решили сыграть в новую настольную игру. Разумеется, играть они будут шариками на клетчатом плоском поле.

Поле для игры представляет из себя клетчатый прямоугольник, строки которого пронумерованы от 0 до 100, а столбцы — от 0 до 100. По некоторым клеткам поля будут раскиданы N шариков. Игроки ходят по очереди. В свой ход игрок может взять шарик с позиции (l_i, c_i) и выбрать положительное целое число u . После этого игрок может передвинуть этот шарик в одну из следующих ячеек:

- $(l_i - u, c_i)$
- $(l_i, c_i - u)$
- $(l_i - u, c_i - u)$

Разумеется, ход можно сделать только если соответствующая клетка существует на поле. Выигрывает игрок, который смог передвинуть какой-либо шарик в позицию $(0, 0)$.

Вы — главный советник императора Шардонии. Он вызвался ходить первым, и спрашивает вас, может ли он выиграть, если оба игрока будут играть оптимально. Ответьте ему на вопрос, потому что иначе вас казнят.

Формат входных данных

На вход программе подается число N ($1 \leq N \leq 1000$) — количество шариков на поле.

В следующих N строках расположены числа l_i, c_i ($0 \leq l_i, c_i \leq 100$) — координаты шариков.

Формат выходных данных

Вы должны вывести символ «Y», если Император может выиграть, и «N» иначе.

Примеры

стандартный ввод	стандартный вывод
2 1 3 2 3	Y
1 1 2	N

Задача G. Битва за кольцо

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

Саруман Белый и Гэндальф Серый решили сыграть в игру. Победителю достается Кольцо Всевластия. Перед игроками лежат кольца, соединенные в K цепочек. Для каждого кольца известно содержание золота в нем 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 ($1 \leq K \leq 50$). 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

стандартный ввод	стандартный вывод
2 3 1 2 1 1 1	G 1 1
2 3 2 1 2 1 1	S
1 10 1 2 3 4 5 6 7 8 9 10	G 1 10

Задача Н. Дровосек

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
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 входного файла находится 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 в выходной файл 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 $n - 1$).

Примеры

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

Задача I. Ним в поддавки

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

Всем вам хорошо известна игра ним: на столе лежит кучка из a камней, своим ходом можно взять из неё любое число камней от 1 до a . Однако, эта игра необычная: если обычно игрок, который не может сделать ход, проигрывает, здесь всё наоборот — игрок, который не может сделать ход.

Как и в привычном вам варианте игры, на столе лежит не одна кучка камней, а n . Каждый игрок своим ходом может взять из любой кучки любое число камней. Вам нужно определить, кто победит в этой игре.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100$). Во второй строке заданы размеры кучек a_i ($1 \leq a_i \leq 1000$).

Формат выходных данных

В единственной строке выведите «FIRST», если победит первый игрок, или «SECOND», если победит второй.

Примеры

стандартный ввод	стандартный вывод
2 1 2	FIRST
1 1	SECOND
2 9 4	FIRST

Задача J. Шоколадка

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

Двое играют в такую игру: перед ними лежит шоколадка размера $N \times M$. Игроки ходят по очереди. За ход разрешается разломить любой имеющийся кусок шоколадки на два куска, при этом запрещено ломать куски размером не больше, чем $1 \times S$ (то есть нельзя ломать куски, у которых один размер равен одному, а другой не превосходит S). Куски можно поворачивать. Ломать куски можно только вдоль линий на шоколадке, то есть после разлома размеры кусков должны являться натуральными числами. Проигрывает тот, кто не может сделать ход.

Формат входных данных

В единственной строке заданы три числа: N , M и S ($1 \leq N, M, S \leq 100$).

Формат выходных данных

Выведите одно число 1 или 2 в зависимости от того, какой из игроков выиграет.

Пример

стандартный ввод	стандартный вывод
2 2 1	1

Задача К. Рикка и Теория Игр

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

Теория игр — крайне интересная область дискретной математики.

Функция SG крайне важна в теории игр. Пусть дан ориентированный ациклический граф G_1 с множеством вершин V_1 и множеством ориентированных ребер E_1 . Для каждой вершины $u \in V_1$ определим функцию $sg(u)$ следующим образом:

$$sg(u) = mex(\{sg(v) \mid (u, v) \in E_1\})$$

Функция $mex(S)$ определяется следующим образом. Пусть дано множество неотрицательных целых чисел S . Тогда $mex(S)$ — минимальное неотрицательное число, которое не принадлежит множеству S .

Сегодня Рикка хочет обобщить функцию SG на неориентированные графы. Дан неориентированный граф G с множеством вершин V и множеством неориентированных ребер E . Функция f , определенная для каждой вершины из V , называется корректной функцией SG для графа G тогда и только тогда, когда:

- Для каждой вершины $u \in V$ $f(u)$ — неотрицательное целое число;
- Для каждой вершины $u \in V$ $f(u) = mex(\{f(v) \mid (u, v) \in E\})$.

Нетрудно заметить, что для некоторых графов существует не одна корректная функция SG. Теперь Рикка хочет обнаружить связь между всеми корректными функциями SG. Для начала он хочет посчитать количество корректных функций SG для заданного неориентированного графа G .

Формат входных данных

В первой строке записаны два целых числа n и m ($1 \leq n \leq 17$, $0 \leq m \leq \frac{n(n-1)}{2}$) — количество вершин и ребер в заданном неориентированном графе.

В каждой из следующих m строк записаны два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$), описывающие ребра графа.

Гарантируется, что граф не содержит петель и кратных ребер.

Формат выходных данных

Выведите одно число — количество корректных функций SG для заданного графа.

Пример

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

Замечание

Для простоты будем обозначать функцию f так: $[f(1), f(2), \dots, f(n)]$.

Для графа из примера существует 6 корректных функций SG: $[0, 1, 0, 1, 0]$, $[0, 1, 2, 0, 1]$, $[0, 2, 1, 0, 1]$, $[1, 0, 1, 0, 1]$, $[1, 0, 1, 2, 0]$ и $[1, 0, 2, 1, 0]$.

Задача L. Корневой ним

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Для игры в корневой ним используются следующие правила. Перед двумя игроками лежит кучка из n камней. Они по очереди забирают оттуда камни. Если в кучке сейчас лежат k камней, то игрок может взять из неё от 1 до $\lfloor \sqrt{k} \rfloor$ камней, включительно. Например, из кучки из 10 камней можно брать 1, 2 или 3 камня. Проигрывает игрок, который не может сделать ход.

По заданному n определите, победит ли первый игрок при правильной игре обоих игроков.

Формат входных данных

Входной файл содержит единственное число n ($1 \leq n \leq 10^{12}$) — количество камней в кучке.

Формат выходных данных

Выведите WIN в случае победы первого игрока, и LOSE, если ему победить не удастся.

Примеры

<code>stdin</code>	<code>stdout</code>
3	WIN
5	LOSE