

Задача А. Две кучки

Имя входного файла: `heaps.in`
Имя выходного файла: `heaps.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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, \dots, a_k камней, второй — b_1, b_2, \dots, b_l . Проигрывает тот, кто не может сделать ход. Определите победителя!

Формат входного файла

В первой строке записаны a и b . Во второй строке записаны k и последовательность a_i , на третьей — l и b_i . $1 \leq a, b \leq 1000$, $1 \leq k, l \leq 10$, $1 \leq a_i, b_j \leq 1000$.

Формат выходного файла

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

Пример

<code>heaps.in</code>	<code>heaps.out</code>
2 2 2 1 2 1 1	First
2 2 1 1 2 1 2	Second

Задача В. Малыш и Карлсон

Имя входного файла: `karlsson.in`
Имя выходного файла: `karlsson.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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 входного файла

Во входном файле содержится 3 целых числа a, b, c ($1 \leq a, b, c \leq 5000$) — размеры пирога.

Формат выходного файла

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

<code>karlsson.in</code>	<code>karlsson.out</code>
1 1 1	NO
2 1 1	YES 1 1 1

Задача С. Сумма игр

Имя входного файла: `smith.in`
Имя выходного файла: `smith.out`
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 — опеределить, кто выиграет при правильной игре.

Формат входного файла

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 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 ($1 \leq v_1 \leq N_1, 1 \leq v_2 \leq N_2$).

Формат выходного файла

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

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

Задача D. Терминатор

Имя входного файла: `terminator.in`
Имя выходного файла: `terminator.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Два игрока играют в настольную игру. Игровое поле представляет собой квадратный лабиринт, 8×8 клеток. В некоторых клетках располагаются стенки. Один игрок управляет фишкой-терминатором, а второй — фишкой-беглецом. Игроки ходят по очереди, ходы пропускать нельзя (гарантируется, что ход всегда возможен). За один ход игрок может переместить свою фишку в любую из свободных клеток, расположенных рядом с исходной по горизонтали, вертикали или по диагонали (то есть ходом короля). Терминатор, кроме того, может стрелять в беглеца ракетами. Выстрел идет по прямой в любом направлении по горизонтали, вертикали или диагонали. Если беглец оказывается на линии выстрела терминатора и не прикрыт стенками, то терминатор незамедлительно

делает выстрел (вне зависимости от того, чей ход), и беглец проигрывает. Начальное положение фишек задано. Первый ход делает беглец. Он выигрывает, если сделает ход с восьмой строки за пределы игрового поля, так как остальные границы поля окружены стенками.

Вопрос задачи: может ли беглец выиграть при оптимальной игре обеих сторон?

Формат входного файла

Во входном файле задано игровое поле. Свободная клетка обозначена цифрой 0, а клетка со стенкой — цифрой 1. Клетка, в которой находится беглец, обозначена цифрой 2, а клетка с терминатором — цифрой 3.

Формат выходного файла

В выходной файл выведите число 1, если беглец выигрывает, и -1 — в противном случае.

Примеры

terminator.in	terminator.out
01000000	-1
10100000	
31100000	
00020000	
00000000	
00000000	
00000000	
00000000	