

## Задача А. Ним

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

Двое играют в игру. Есть несколько кучек спичек. За один ход разрешается взять любое ненулевое количество спичек из любой кучки. Кто не может сделать ход, тот проиграл. Определите, кто выигрывает при правильной игре.

### Формат входных данных

В первой строке входного файла записано натуральное число  $N$  — количество кучек. Во второй строке записаны  $N$  целых чисел — количество спичек в кучках. Все числа во входном файле находятся в пределах от 1 до 100000.

### Формат выходных данных

Выведите через пробел два числа: номер кучки  $i$ , из которой нужно взять спички ( $1 \leq i \leq N$ ) и количество спичек, которые нужно взять из этой кучки. Если в данной игре первый игрок не может выиграть, выведите «0».

### Примеры

<code>nim.in</code>	<code>nim.out</code>
1 10	1 0
2 1 1	0

## Задача В. Битва за кольцо

Имя входного файла: `rings.in`  
Имя выходного файла: `rings.out`  
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

<code>rings.in</code>	<code>rings.out</code>
2 3 1 2 1 1 1	G 1 1
2 3 2 1 2 1 1	S

## Задача С. МаксМинТриГейм

Имя входного файла: `mmtree.in`  
Имя выходного файла: `mmtree.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

МаксМинТриГейм это игра для двух игроков. В эту игру играют на дереве. Дерево имеет  $N$  узлов, помеченных от 0 до  $N-1$ . Каждый узел дерева также имеет целочисленную стоимость. Игроки ходят поочередно. В свою очередь, текущий игрок выбирает одно ребро дерева и стирает его. Это неизбежно делит дерево на две составляющие. Текущий игрок решает, какую из компонент сохранить, и полностью стирает другую. Игра заканчивается, когда остался только один узел. Стоимость этого узла является результатом игры.

Первый игрок (то есть тот, кто начинает игру) хочет максимизировать результат. Естественно, цель второго игрока является сведение к минимуму результата.

Вам предоставляется дерево и стоимости каждой вершины. Найдите результат игры, при условии, что оба игрока играют оптимально.

### Формат входных данных

В первой строке входного файла одно целое число —  $N$ . ( $2 \leq N \leq 50$ )

Во второй строке  $N - 1$  целых  $p[i]$  ( $0 \leq p[i] \leq i$ ) означает, что для каждого  $i$  от 0 до  $N - 1$ , существует ребро между вершинами  $p[i]$  и  $i + 1$  (нумерация начинается с 0).

В третьей строке —  $N$  целых чисел — стоимости каждой вершины в порядке от вершины 0 до вершины  $N - 1$ . ( $0 \leq cost[i] \leq 10^9$ )

### Формат выходных данных

Выведите одно число — результат игры.

### Примеры

<code>mmtree.in</code>	<code>mmtree.out</code>
2 0 4 6	6
3 0 1 4 6 5	5

## Задача D. Ретроанализ для маленьких

Имя входного файла: `retro.in`  
Имя выходного файла: `retro.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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 так: на первой строке 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.

### Примеры

<code>retro.in</code>	<code>retro.out</code>
5 5	DRAW
1 2	DRAW
2 3	DRAW
3 1	FIRST
1 4	SECOND
4 5	FIRST
2 1	SECOND
1 2	FIRST
4 4	FIRST
1 2	SECOND
2 3	SECOND
3 1	
1 4	

## Задача Е. 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 — определить, 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 ( $1 \leq N_1, M_1 \leq 10\,000$ ). На следующих  $M_1$  строках содержится по два числа  $x$  и  $y$  ( $1 \leq x, y \leq N_1$ ). В следующей строке вводятся  $N_2$  и  $M_2$  — количество вершин и рёбер во втором графе соответственно. Далее в следующих  $M_2$  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 выходных данных

На каждую из  $T$  пар начальных вершин выведите строку `“first”`, если при правильной игре выиграет первый, `“second”`, если второй, или `“draw”`, если будет ничья.

### Примеры

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