

Задача А. Огромный ним

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

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 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 | Win |
| 1 10 | 8 3 |
| 2 | Lose |
| 2 5 | |
| 2 5 | |

Задача В. Дровосек

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
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 находятся 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 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 $N - 1$).

Пример

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

Задача С. Операции с простыми возвращаются

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.3 секунд
Ограничение по памяти: 256 мегабайт

Дэвид Блейн и Вася сидят на уроке математики и скучают. Через некоторое время Дэвид Блейн предлагает поиграть в следующую игру:

- он записывает на листочке некоторое большое целое число и ходит первым;
- игроки ходят по очереди;
- на каждом ходу игрок обязан разделить текущее число на его простой делитель p . На этом же ходу игрок может умножить (а может и нет) результат деления на простое число q ($1 < q < p$);
- проигрывает тот, кто не может сделать ход.

Мягко говоря, Вася не очень доверяет Дэвиду Блейну и боится, что тот выписывает только проигрышные для Васи начальные числа. Помогите ему определить, так это или нет.

Формат входных данных

Первая строка входного файла содержит единственное число n ($1 \leq n \leq 10^{12}$).

Формат выходных данных

В выходной файл выведите `Vasya`, если у Васи есть выигрышная стратегия, независимо от ходов Дэвида Блейна. Иначе выведите `David`.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 16 | Vasya |

Задача D. Вариация нима

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 256 мегабайт

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

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

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 | FIRST |
| 1 1 | SECOND |
| 2 1 1 | FIRST |
| 3 1 2 3 | |

Задача Е. Ретроанализ

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 0.3 секунд |
| Ограничение по памяти: | 256 мегабайт |

Дан ориентированный весёлый граф из n вершин и m ребер. Двое 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 | FIRST |
| 2 1 | SECOND |
| 1 2 | FIRST |
| 4 4 | FIRST |
| 1 2 | SECOND |
| 2 3 | SECOND |
| 3 1 | |
| 1 4 | |

Задача F. Чапаев на дереве

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1.5 секунд |
| Ограничение по памяти: | 512 мегабайт |

Вова и Марина любят играть в игры, а особенно — придумывать к ним свои правила. Недавно они открыли для себя веселую игру «Чапаев», в которой игроки должны сбивать щелчками шашки вражеского цвета с шахматной доски (также эта игра известна под названием «Щелкунчики»). Вдоволь наигравшись, они решили модифицировать правила, добавив игре математическую сложность.

Теперь они играют в «Чапаева» не на шахматной доске, а на доске в форме дерева! Их дерево состоит из N вершин. Вершина 1 является корнем дерева, а из каждой из оставшихся вершин проведено ребро в некоторую вершину с меньшим номером — её непосредственного предка.

В игре участвуют шашки одного цвета, изначально расположенные в некоторых вершинах дерева. За один ход игрок выбирает некоторую шашку и щелчком отправляет ее к корню по ребрам дерева, сбивая при этом с доски все встреченные на пути шашки. Сама шашка, по которой производился удар, после попадания в корень дерева также слетает с доски.

Игроки делают ходы по очереди. Проигрывает тот игрок, к ходу которого на доске не остается шашек.

Придуманная ими игра замечательна также тем, что на одной и той же доске можно играть, начиная с разных начальных позиций шашек. Практика показала, что самые интересные партии получаются, если исходно расставить фишки во все вершины, являющиеся потомками (непосредственными или косвенными) некоторой вершины $Root$, при этом в саму вершину $Root$ фишка не ставится.

Дети решили сыграть N партий, перебрав в качестве вершины $Root$ каждую вершину дерева по одному разу. Если у очередной вершины $Root$ нет потомков, и на доске исходно не оказывается ни одной фишки, то игры не происходит, и дети переходят к следующей расстановке. В каждой партии Марина ходит первой.

Вова интересуется у вас, в скольких партиях Марина сможет одержать победу, если игроки будут действовать оптимально.

Формат входных данных

В первой строке находится целое число N ($1 \leq N \leq 500\,000$) — количество вершин в дереве.

Во второй строке следуют целые числа p_2, p_3, \dots, p_N , разделенные пробелами, где p_i — это номер вершины, являющейся предком вершины i ($1 \leq p_i \leq i$).

Формат выходных данных

Выведите единственное целое число — количество партий, в которых Марина одержит победу.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 7 1 2 3 1 5 5 | 3 |

Замечание

Разберем тест из условия. Доска для игры показана на рисунках ниже. Дети сыграют четыре партии, выбирая в качестве $Root$ вершины 1, 2, 3 и 5. Если выбрать в качестве $Root$ любую из трех оставшихся вершин, на доске исходно не окажется ни одной фишки, поэтому игры не произойдет.

Если выбрать в качестве $Root$ вершину 5, фишки будут исходно находиться в вершинах 6 и 7. В такой партии Марина проигрывает: после того, как она сбивает любую из этих двух фишек с доски, Вова сбивает оставшуюся и заканчивает партию.

Если выбрать в качестве $Root$ вершину 2 или 3, у Марины будет возможность выиграть игру за один ход, щелкнув по фишке из вершины 4 (при этом, в случае $Root = 2$, она по пути также собьет фишку из 3 вершины по правилам игры)

Можно убедиться, что если выбрать в качестве *Root* вершину 1, у Марины также будет выигрышная стратегия. Для этого первым ходом Марина должна сбить фишку из вершины 2. Пример партии с таким начальным расположением показан ниже.

Таким образом, Марина выигрывает в трех партиях