

Задача А. Альфа Дерево

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

У вас есть полное бинарное дерево глубины n ($0 \leq n \leq 30$).

В дереве 2^n листьев, они пронумерованы слева направо числами от 0 до $2^n - 1$.

В i -м листе записано число $x_i = (ai^2 + bi + c) \bmod m$.

Есть фишка, которая изначально находится в корне дерева. Двое играют в игру, двигая фишку вниз по дереву. Когда фишка достигает листа дерева, игра заканчивается. Цель первого игрока – максимизировать число в листе, цель второго – минимизировать.

Формат входных данных

Числа n, a, b, c, m . При этом $10 \leq m \leq 10^9$.

Все a, b, c сгенерированы равномерным распределением на $[0, m)$.

Формат выходных данных

Выведите результат игры при оптимальной игре обоих.

Система оценки

Оценка потестовая. Все тесты независимы.

Пример

стандартный ввод	стандартный вывод
3 10 7 9 20	11

Замечание

Взятие остатка по модулю — небыстрая операция. Чем их меньше, тем лучше.

Задача В. Бинарная игра

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

Искандер и Оля любят придумывать ребусы. Но больше, чем придумывать ребусы, они любят придумывать какие-нибудь игры на строках. Вот и сейчас им в голову пришла забавная игра со следующими правилами:

- Выбирается какой-то набор *запрещённых* двоичных (состоящих из нулей и единиц) строк f_1, f_2, \dots, f_n .
- Выбирается некоторая стартовая бинарная строка s , такая что ни одна из запрещённых строк не входит в неё как подстрока.
- Игроки по очереди дописывают в конец строки s по одному символу «0» или «1». Оля ходит первой.
- Проигрывает тот, после чьего хода хотя бы одна из запрещённых строк f_1, f_2, \dots, f_n входит в s как подстрока.
- В случае если при оптимальной игре обоих игроков игра может продолжаться сколь угодно долго, то объявляется ничья.

Вы обожаете портить другим людям их любимые развлечения, поэтому решили написать программу, которая будет определять исход игры по заданному набору запрещённых строк и стартовой строке s .

Формат входных данных

В первой строке входных данных записаны два целых числа n и m ($0 \leq n \leq 100\,000$, $0 \leq m \leq 1\,000\,000$) — количество запрещённых строк и изначальная длина строки s .

В каждой из последующих n строк содержится одна запрещённая строка. Гарантируется, что все эти строки непусты, состоят из символов «0» и «1» и никакая из них не является подстрокой строки s . Дополнительно гарантируется, что **суммарная длина** всех запрещённых строк не превосходит 1 000 000.

В последней строке входных данных записана стартовая строка s длины m , состоящая только из символов «0» и «1». Обратите внимание, строка s может быть пустой, в этом случае соответствующая строка входных данных отсутствует (в том числе символ перевода строки). Длина s не превосходит 1 000 000.

Формат выходных данных

В зависимости от результата игры при оптимальной игре обоих игроков выведите:

- «Olya» (без кавычек), если Оля может победить вне зависимости от того как будет играть Искандер. Напомним, что Оля ходит первой.
- «Iskander» (без кавычек), если Искандер может победить не зависимо от ходов Оли.
- «Friendship» (без кавычек), если при оптимальной игре обоих игроков игра будет продолжаться бесконечно долго.

Примеры

стандартный ввод	стандартный вывод
1 0 1	Friendship
3 1 000 001 011 0	Olya
2 3 1001 000 100	Iskander

Замечание

Если вы не слушали вводную лекцию про анализ игр на ациклических и циклических графах, рекомендую первую главу данной статьи:

<https://ejudge.lksh.ru/archive/2014/07/A/games.pdf>

Задача С. Вас снова замяукали!

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

Два котёнка попали в запутанный лабиринт со множеством комнат и переходов между ними. Котят долго по нему плутали, обошли все комнаты по много раз, нашли выход (да даже и не один, а несколько), в общем, изучили там всё, что смогли. Теперь этот лабиринт котят используют в своих играх.

Чаще всего котят играют в следующую игру: начиная в какой-то комнате лабиринта, котят поочередно выбирают, в какую из комнат им перейти. Котят изначально находятся в одной комнате и ходят вместе. Как только котёнок, который должен выбрать следующую комнату, не может этого сделать, он признаётся проигравшим. Обычно в таких играх выигрывающий игрок стремится выиграть как можно быстрее, а проигрывающий стремится как можно дольше оттянуть свое поражение. Но у котят свои представления о победе и поражении. Если котёнок знает, что, начиная из текущей комнаты, он выиграет (вне зависимости от действий другого котёнка), то он стремится играть как можно дольше, чтобы продлить себе удовольствие от выигрыша (естественно, при этом выигрывающий котёнок должен гарантировать себе, что будет постоянно уверен в выигрыше). Котёнок, который знает, что проиграет (при условии, конечно, что другой котёнок будет действовать оптимально), старается проиграть как можно быстрее, чтобы начать новую игру, в которой и взять реванш.

Если котят будут ходить бесконечно долго, но никто из них не сможет выиграть, то котят считают игру завершившейся вничью и замяукивают Вас.

Вас попросили для каждой комнаты в лабиринте узнать, выиграет или проиграет котёнок, начинающий ходить из данной комнаты. Если котёнок, начинающий из этой комнаты, выигрывает, требуется узнать максимальное количество ходов, которое он сможет играть, если же проигрывает — минимальное количество, которое ему придётся играть.

Формат входных данных

В первой строке ввода находятся два числа n и m — число комнат и переходов между комнатами в лабиринте ($1 \leq n \leq 100\,000$, $0 \leq m \leq 100\,000$). Далее следует m строк с описаниями переходов. Описание перехода состоит из двух чисел a и b , означающих, что котёнок, начинающий игру в комнате с номером a , может выбрать комнату b в качестве следующей.

Формат выходных данных

Выведите n строк — для каждой комнаты результат игры для котёнка, который начнет игру из этой комнаты. Если игра закончится вничью, выведите «DRAW». Если начинающий котёнок выиграет, выведите «WIN K », где K — количество ходов, которые сможет играть выигрывающий котёнок. Если котёнок сможет играть сколь угодно долго, сохраняя возможность в любой момент выиграть, выведите «WIN INF». Если котёнок, начинающий из этой комнаты, проиграет, выведите «LOSE K », где K — количество ходов, которые придется играть проигрывающему котёнку. Если же котёнку придется играть сколь угодно долго, при том, что его соперник сможет в любой момент выиграть, выведите «LOSE INF».

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 1 3 2 4 3 4	LOSE 2 WIN 1 WIN 1 LOSE 0
6 6 1 2 2 3 3 4 4 1 4 5 5 6	DRAW DRAW DRAW DRAW WIN 1 LOSE 0
6 6 1 2 2 3 3 4 4 1 2 6 4 5	LOSE INF WIN INF LOSE INF WIN INF LOSE 0 LOSE 0

Задача D. Сумма игр

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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.

Пример

стандартный ввод	стандартный вывод
3 2	first
1 2	second
2 3	
2 1	
1 2	
2	
1 1	
3 2	

Задача F. Кучки с конфетами

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

Есть n кучек с конфетами на столе. Они пронумерованы целыми числами от 1 до n , в i -й кучке находится a_i конфет.

Два игрока играют в игру. За ход можно сделать одно из двух действий:

1. Выбрать любую кучку, в которой осталось максимальное число конфет и съесть все конфеты из этой кучки;
2. Съесть по 1 конфете из всех непустых кучек.

Игрок, который съест последнюю конфету на столе проигрывает. Определите, кто выигрывает при правильной игре.

Формат входных данных

В первой строке находится единственное целое число n ($1 \leq n \leq 10^5$) — количество кучек. Во второй строке находится n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — количества конфет в каждой из кучек.

Формат выходных данных

Если первый игрок имеет выигрышную стратегию, выведите «First» (без кавычек), иначе выведите «Second» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
2 1 3	First
3 1 2 1	First
3 1 2 3	Second

Задача G. 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 99, а столбцы — от 0 до 99. 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 ($1 \leq l_i, c_i \leq 100$) — 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 Н. 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

Задача I. Дровосек

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 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 J. 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 К. 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 L. 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 М. 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 N. 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 О. Смак

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

Правильно ли я понимаю, что наше рабочее место на данной кулинарной передаче ограничено одним столом, одним тестом и тремя мисками? Вы же понимаете, что мы просто физически здесь не поместимся?

Михаил Аноприенко

Иван Ургант пригласил Ивана Сафонова на свою кулинарную передачу «Смак». В сегодняшнем выпуске они будут готовить огромный пирог из теста, которое уже заготовлено помощниками Урганта и налито в миску. В миске всегда находится целое неотрицательное число граммов теста. Оба Ивана, начиная с Урганта, по очереди применяют один из двух кулинарных приёмов:

1. Набрать мерным стаканом один грамм теста из миски и добавить в форму для пирога.
2. Взбить тесто миксером. При этом часть теста расплёскивается. Если в миске перед взбиванием было G граммов теста, то останется $\varphi(G)$ граммов теста. φ — это функция Эйлера, которая по данному положительному целому числу G возвращает количество целых чисел от 1 до G включительно, которые взаимно просты с G , то есть не имеют с G общих делителей, больших единицы. Приведём таблицу, в которой вычислены некоторые значения этой функции:

G	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\varphi(G)$	1	1	2	2	4	2	6	4	6	4	10	4	12	6	8	8	16	6	18	8

Когда тесто в миске полностью закончится, пирог будет считаться приготовленным; более того, тот из Иванов, чей приём потратил последний грамм теста в миске, и будет считаться поваром, приготовившим пирог. Считая, что оба Ивана желают считаться создателями пирога и будут для этого каждый раз оптимально выбирать один из двух приёмов, определите, чей кулинарный приём завершит готовку.

Формат входных данных

В первой строке находится одно целое положительное число G ($1 \leq G \leq 10^9$) — изначальное количество теста в граммах.

Формат выходных данных

Если Иван Ургант может гарантированно завершить готовку пирога, в первой строке выведите «Ivan Urgant». Во второй строке выведите «Draw», если для этого Урганту нужно первым действием набрать грамм теста из миски и налить в форму, или же выведите «Mix», если первым действием ему надо взбить тесто. Если Ургант при выполнении любого из двух приёмов может гарантировать, что именно он завершит готовку пирога, то вы можете вывести любое одно из двух слов «Draw» и «Mix».

Если Иван Сафонов может гарантированно завершить готовку пирога, в первой строке выведите «Ivan Safonov».

Примеры

стандартный ввод	стандартный вывод
1	Ivan Urgant Draw
2	Ivan Safonov
4	Ivan Urgant Mix
14	Ivan Urgant Draw

Замечание

Если в миске один грамм теста, Ургант просто должен налить его в форму и сразу завершить готовку пирога.

Если в миске два грамма теста, то любой из двух кулинарных приёмов Урганта приведёт к тому, что в миске останется один грамм теста, после чего Сафонов легко доготовит пирог.

Если в миске четыре грамма теста, то Ургант проиграет, если нальёт один грамм теста в форму, и выиграет, если вместо этого он взобьёт тесто и оставит в миске $\varphi(4) = 2$ грамма.

Если в миске 14 граммов теста, то всё наоборот: если Иван Ургант его взобьёт, то оставит в миске $\varphi(14) = 6$ граммов теста, в результате чего он проиграет; если же он нальёт один грамм теста в форму, то в миске останется 13 граммов теста, и можно доказать, что в такой ситуации Иван Сафонов не может гарантировать, что именно он приготовит пирог.

Задача Р. Число совершенных паросочетаний

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Совершенным называется паросочетание, покрывающее все вершины графа.

Дан произвольный неориентированный граф.

Найдите количество совершенных паросочетаний по модулю $10^9 + 7$.

Формат входных данных

Число вершин n ($1 \leq n \leq 30$). Число рёбер m ($0 \leq m \leq \frac{n(n-1)}{2}$)

Формат выходных данных

Выведите одно число – количество совершенных паросочетаний по модулю $10^9 + 7$.

Пример

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

Задача Q. Японский компьютер

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

Как известно, для обороны границ японские инженеры разрабатывают огромных боевых человекоподобных роботов. Каждый такой робот управляется японским компьютером. Понятно, что для повышения эффективности работа программа в компьютере должна быть как можно более оптимальной, чтобы компьютер мог выполнять как можно больше программ за как можно меньшее время.

На данный момент японским программистам задали следующую задачу (её смысл секретен, поэтому здесь его описывать нельзя): изначально в памяти компьютера находится единственное число x . Требуется получить в его памяти следующие числа: a_1x, a_2x, \dots, a_nx . При этом компьютер может выполнять следующие операции:

1. Сложение двух чисел
2. Вычитание двух чисел
3. Побитовый сдвиг влево (сдвиг на k бит эквивалентен умножению на 2^k)

Все полученные промежуточные значения сохраняются в памяти, так что ими можно пользоваться при вычислении других значений.

При вычислениях никогда не должно получаться значение большее, чем $42x$. Гарантируется, что при выполнении этого ограничения, в компьютере не происходит переполнений. Также, компьютер не может работать с отрицательными числами, так что вычитать большее число из меньшего также запрещено.

Порядок, в котором в памяти будут появляться числа a_1x, a_2x, \dots, a_nx , не имеет значения.

Формат входных данных

В первой строке находится число n — количество требуемых значений ($1 \leq n \leq 41$). Во второй строке находится n чисел a_i ($2 \leq a_i \leq 42$). Все a_i различны. Само число x вам не дано, так что ваша последовательность операций должна быть верной для любого x .

Формат выходных данных

В первой строке выведите единственное число — минимальное количество требуемых операций. Далее выведите требуемые операции в следующем формате:

1. Сдвиг влево ax на k бит: “ $a<<k$ ”
2. Сложение ax и bx : “ $a+b$ ”
3. Вычитание ax из bx : “ $b-a$ ”

Запись операций не должна содержать пробелов.

Примеры

стандартный ввод	стандартный вывод
3 3 5 18	5 1+1 2+1 3+2 1<<4 16+2
1 29	4 1+1 2+1 1<<5 32-3
4 12 19 41 42	8 1+1 2+1 3<<2 12+12 24-2 22-3 19+22 41+1

Задача R. Коробки

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

Дано N коробок в пространстве. Про каждую коробку известны её три измерения. Коробки можно вращать — то есть выбирать, какое измерение будет длиной, какое — шириной, а какое — высотой. Из этих коробок необходимо построить башню максимальной высоты. Коробки разрешается ставить друг на друга, только если длина коробки сверху не превосходит длины коробки снизу, а ширина коробки сверху — ширины коробки снизу.

Формат входных данных

В первой строке задано целое число N ($1 \leq N \leq 30$). Следующие N строк содержат по три целых числа a_i, b_i и c_i ($1 \leq a_i, b_i, c_i \leq 10^6$) — три измерения i -ой коробки.

Формат выходных данных

В первой строке выведите максимальную высоту башни H . Во второй строке выведите целое число k — количество коробок в искомой башне. В последующих k строках выведите по четыре числа для каждой коробки — её номер m_i , длину l_i , ширину d_i и высоту h_i (коробки нужно выводить в порядке снизу вверх). Коробки нумеруются с единицы в порядке, заданном во входном файле.

Пример

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