

Задача А. 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 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.

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 В. Странная игра

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

Двое играют в простую игру на доске $n \times n$. У первого игрока есть одна белая фишка, а у второго — одна чёрная. Игроки ходят по очереди, первым ходит первый игрок (белые).

Первый игрок имеет право двигать свою фишку на одну клетку в одном из четырёх основных направлений (влево, вправо, вверх, вниз). Второй игрок при своем ходе также выбирает одно из этих четырёх направлений, но может передвинуть свою фишку как на одну клетку в этом направлении, так и на две. Выигрывает тот, кто первым съедает фишку соперника.

Определите победителя и число ходов, требуемое для победы, при оптимальной игре сторон.

Формат входных данных

Во входном файле даны пять чисел — n ($2 \leq n \leq 20$), а также координаты белой и чёрной фишек. Фишки стоят в разных клетках доски.

Формат выходных данных

Выведите `WHITE` x , если выигрывают белые, `BLACK` x , если выигрывают чёрные, `DRAW`, если игра закончится вничью. Здесь x — число ходов обеих сторон (полуходов) до момента окончания игры.

Примеры

<code>strange-game.in</code>	<code>strange-game.out</code>
2 1 1 2 2	BLACK 2
2 2 2 1 2	WHITE 1
3 1 1 3 3	BLACK 6

Задача С. Малыш и Карлсон

Имя входного файла: `karlsson.in`
Имя выходного файла: `karlsson.out`
Ограничение по времени: 2 секунды
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 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

<code>karlsson.in</code>	<code>karlsson.out</code>
1 1 1	NO
1 2 1	YES 1 1 1
1 1 10	YES 1 1 7

Задача D. Жестокая задача

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

Штирлиц и Мюллер стреляют по очереди. В очереди n человек, стоящих друг за другом. Каждым выстрелом убивается один из стоящих. Кроме того, если у кого-то из стоящих в очереди убиты все его соседи, то этот человек в ужасе убегает. Проигрывает тот, кто не может ходить. Первым стреляет Штирлиц. Требуется определить, кто выиграет при оптимальной игре обеих сторон, и если победителем будет Штирлиц, то найти все возможные первые ходы, ведущие к его победе.

Формат входных данных

Входной файл содержит единственное число n ($2 \leq n \leq 5\,000$) — количество человек в очереди.

Формат выходных данных

Если выигрывает Мюллер, выходной файл должен состоять из единственного слова `Mueller`. Иначе в первой строке необходимо вывести слово `Schtirlitz`, а в последующих строках — номера людей в очереди, которых мог бы первым ходом убить Штирлиц для достижения своей победы. Номера необходимо выводить в порядке возрастания.

Примеры

<code>cruel.in</code>	<code>cruel.out</code>
3	Schtirlitz 2
4	Mueller
5	Schtirlitz 1 3 5

Задача Е. Альфа дерево

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
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.

Пример

стандартный ввод	стандартный вывод
3 10 7 9 20	11

Замечание

Взятие остатка по модулю – быстрая операция. Чем их меньше, тем лучше.

Задача F. Японский компьютер

Имя входного файла:	computer.in
Имя выходного файла:	computer.out
Ограничение по времени:	2 секунды
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

computer.in	computer.out
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 G. Монетки

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

На столе лежат в ряд N кучек монеток. В i -й кучке лежит ровно a_i монеток. При этом оказалось, что $a_i \leq a_j$ при $i < j$.

Катя и Серёжа играют с этими монетками в игру. За ход можно взять любое ненулевое количество монеток из любой кучки, но условие ($a_i \leq a_j$ при $i < j$) должно сохраниться. Выигрывает, как обычно, взявший последнюю монету.

Вам требуется определить, кто выиграет при правильной игре. Серёжа ходит первый.

Формат входных данных

В первой строке задано число N ($1 \leq N \leq 100\,000$). В следующей строке задано N чисел — a_1, a_2, \dots, a_N ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Если выиграет Серёжа, выведите «Sergey», иначе выведите «Katya».

Примеры

стандартный ввод	стандартный вывод
1 10	Sergey
2 10 15	Sergey
2 10 10	Katya