

Задача А. Двоичный Гаусс

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

Дан набор из n битовых векторов размера n и еще один вектор. Получить этот вектор как хог исходных.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 300$). В следующих n строках записаны исходные вектора, в последней строке записан вектор, который нужно получить.

Формат выходных данных

Если решений нет, выведите **No solution**. Если решений несколько, выведите **Multiple solutions**. Если решение единственное, выведите номера векторов (вектора нумеруются с 0), которые нужно сложить, чтобы получить данный вектор. Номера выведите в порядке возрастания.

Примеры

стандартный ввод	стандартный вывод
3 100 111 101 010	1 2
3 100 111 011 010	No solution
3 111 010 101 000	Multiple solutions

Задача В. Обобщенные числа фибоначчи

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

Мы чуть-чуть обобщили для вас последовательность Фибоначчи, теперь:

$$f_1 = f_2 = 1$$

$$f_i = a \cdot f_{i-1} + b \cdot f_{i-2} + c \cdot 2^i + d \cdot i + e, \text{ для } i > 2$$

Дано n , найдите значение f_n , взятое по модулю $10^9 + 7$.

Формат входных данных

Неотрицательные целые числа: a, b, c, d, e, n . ($0 \leq a, b, c, d, e \leq 10^9$; $1 \leq n \leq 10^{18}$)

Формат выходных данных

Выведите f_n , взятое по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
1 1 0 0 0 8	21
1 2 3 4 5 6	775

Задача С. Всё, везде и Гаусс

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

Для множества неотрицательных целых чисел S назовем $X(S)$ множество всех возможных XOR-значений, которые можно получить, взяв XOR любого подмножества S :

$$X(S) = \{x_1 \oplus x_2 \oplus \dots \oplus x_k \mid x_1, \dots, x_k \in S\},$$

где $\oplus$ обозначает операцию побитового исключающего ИЛИ. Допускается подмножество из нуля элементов, и XOR его элементов это 0, поэтому $0 \in X(S)$ всегда.

Требуется поддерживать множество S и обрабатывать запросы четырёх типов:

- **add** x — добавить число x в S ;
- **can** x — проверить $x \in X(S)$;
- **cnt** — вывести количество различных значений в $X(S)$;
- **max** — вывести максимальный элемент множества $X(S)$.

Изначально S пусто.

Формат входных данных

В первой строке входных данных дано одно число Q ($1 \leq Q \leq 10^5$) — количество запросов. В каждой из следующих Q строк вводится запрос одного из четырех типов.

- **add** x — добавить число x в S ($0 \leq x < 2^{62}$);
- **can** x — проверить $x \in X(S)$ ($0 \leq x < 2^{62}$);
- **cnt** — найти $|X(S)|$;
- **max** — найти максимальное число в $X(S)$.

Формат выходных данных

Для каждого запроса, кроме **add**, необходимо вывести ответ:

- **can** x : вывести 1, если $x \in X(S)$, иначе 0;
- **cnt**: вывести $|X(S)|$;
- **max**: вывести максимальный элемент из $X(S)$.

Примеры

стандартный ввод	стандартный вывод
6 add 11 add 3 max add 38 max cnt	11 46 8
24 max cnt can 0 can 1 add 9 max cnt can 12 can 3 add 5 can 12 max cnt add 12 max cnt can 4 add 8 can 4 can 1 can 0 can 2 cnt max	0 1 1 0 9 2 0 0 1 12 4 12 4 0 1 1 1 0 8 13

Задача D. Полные квадраты

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

Множество целых положительных чисел будем называть *полноквадратным*, если произведение его элементов является полным квадратом (равно 1, 4, 9, 16, 25, 36, ...).

Задано множество A . Определите, сколько непустых подмножеств B множества A являются полноквадратными.

Формат входных данных

В первой строке записано целое число N — количество элементов множества ($1 \leq N \leq 100$). Во второй строке записаны N попарно различных чисел a_i — элементы множества ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — количество полноквадратных подмножеств по модулю 1 000 000 007.

Пример

стандартный ввод	стандартный вывод
4 49 20 500 7	3

Задача Е. XOR-раскраска

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

Дан неориентированный граф с n вершинами и m рёбрами. Художник Том хочет покрасить каждую вершину графа в один из цветов. У Тома есть 2^{60} различных красок, пронумерованных числами от 0 до $2^{60} - 1$. К сожалению, краска цвета 0 закончилась, поэтому использовать её нельзя.

Том хочет добиться следующего: для каждой вершины XOR цветов всех её соседей (вершин, соединённых с ней ребром) должен быть равен 0. Помогите Тому — найдите подходящую раскраску или сообщите, что это невозможно.

Формат входных данных

В первой строке входных данных вводятся два целых числа n и m — количество вершин и рёбер в графе ($1 \leq n \leq 60$, $0 \leq m \leq \frac{n(n-1)}{2}$). В каждой из следующих m строк вводятся u_i и v_i — номера вершин, соединённых i -м ребром ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$). Гарантируется, что все рёбра различны.

Формат выходных данных

Если искомого способа покраски не существует, выведите в единственной строке выходных данных слово «No».

Иначе, выведите в первой строке слово «Yes». В следующей строке выведите n чисел $c_1, c_2, \dots, c_n$ — цвета вершин в искомой покраске ($1 \leq c_i \leq 2^{60} - 1$).

Примеры

стандартный ввод	стандартный вывод
2 0	Yes 1152921504606846975 1
4 4 1 2 2 3 3 4 4 1	Yes 179 239 179 239
6 7 1 2 2 3 3 4 4 1 2 5 4 6 5 6	Yes 3 1 2 1 1 1

Замечание

XOR пустого множества полагается равным нулю, поэтому, если у вершины нет соседей, условие для неё автоматически выполнено.

Задача F. Сумма сумм

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

Дан массив из n чисел $A_1 \dots A_n$. Вы должны покрасить числа в нём в два цвета: красный и синий. При этом обязательно должно быть хотя бы одно число каждого цвета. После этого *ценностью* цвета называется побитовое исключающее или (XOR) всех чисел этого цвета. А *ценностью* массива называется сумма ценностей синего и красного цвета.

Найдите максимальную возможную ценность массива при оптимальной покраске чисел.

Формат входных данных

В первой строке вводится одно число n ($2 \leq n \leq 10^5$). Во второй строке вводятся n чисел $A_1 \dots A_n$ ($0 \leq A_i < 2^{60}$).

Формат выходных данных

Выведите одно число — максимальную возможную ценность массива.

Примеры

стандартный ввод	стандартный вывод
3 3 6 5	12
4 23 36 66 65	188

Замечание

В первом примере нужно покрасить в красный цвет числа 3 и 5, а в синий число 6. Тогда ценности цветов $3 \oplus 5 = 6$ и 6, соответственно.

Задача G. Шоппинг

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Сегодня долгожданный для всех школьников — первый день каникул нового учебного года. Наша главная героиня — Дени — учится в 10 классе. Она хорошо подготовилась к сегодняшнему дню и выяснила, что в центре города находятся N магазинов. Теперь Дени планирует вместе со своими друзьями посетить некоторые из них. В городе есть M пар магазинов (x_i, y_i) , соединенных двусторонними дорогами. Для каждой дороги известно время, которое требуется для перемещения по ней, оно одно и то же для перемещения в обоих направлениях. Никакой магазин не соединен дорогой сам с собой, никакая пара магазинов не соединена более чем одной дорогой.

Дени очень суеверна и одно из её суеверий заключается в том, что она верит, что время, потраченное на перемещения между магазинами, должно нацело делиться на D . При этом Дени с друзьями не может перемещаться между магазинами слишком долго, её путь должен занимать суммарно не больше K . Дени очень любопытна. Она хочет выяснить, сколько существует различных способов начать свой путь в некотором магазине, перемещаться по дорогам между магазинами, и закончить путь в некотором магазине (возможно посещая по пути некоторые магазины и/или дороги более одного раза). Дени помнит, что у нее есть друг-программист — вы — и она просит написать программу, которая вычислит количество корректных способов перемещаться между магазинами. Дени считает способ корректным, если её время в пути не превышает K и делится на D . Вы немедленно указываете Дени, что количество путей может быть слишком большим, поэтому Дени просит вывести остаток от деления количества путей на число 1 000 000 007.

Формат входных данных

На первой строке ввода находятся четыре целых числа N , M , D и K ($2 \leq N \leq 30$, $2 \leq M \leq 435$, $2 \leq D \leq K \leq 10^9$).

На каждой из следующих строк находятся по три целых числа x_i , y_i и t_i ($1 \leq x_i, y_i \leq N$, $1 \leq t_i \leq 10$) — они задают двустороннюю дорогу между магазинами x_i и y_i , перемещение по которой занимает t_i ($1 \leq i \leq M$).

Формат выходных данных

Выведите остаток от деления количества искомых путей на 1 000 000 007.

Примеры

стандартный ввод	стандартный вывод
3 3 2 2 1 2 1 2 3 2 3 1 1	8
5 7 5 10 1 3 8 2 5 7 3 4 3 1 4 2 2 3 1 1 5 4 4 5 4	58
5 9 2 20 1 2 1 2 3 2 3 1 1 3 4 1 4 5 2 5 3 1 1 5 1 2 4 1 2 5 1	989802661
5 7 5000000 5000000 1 3 8 2 5 7 3 4 3 1 4 2 2 3 1 1 5 4 4 5 4	598634781

Задача Н. Хорошие раскраски — deluxe edition

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

Эта задача не относится к серии задач «Хорошие раскраски», но иных названий для задач я придумать не умею.

Дан гиперграф из n вершин и m ребер. В этой задаче ребра гиперграфа соединяют тройки вершин, i -е ребро соединяет тройку вершин (a_i, b_i, c_i) . Раскраска вершин этого гиперграфа в 3 цвета называется *хорошей*, если для каждого ребра (a, b, c) , выполняется одно из двух условий:

- Все вершины a, b, c раскрашены в один и тот же цвет.
- Все вершины a, b, c раскрашены в попарно различные цвета.

Некоторые вершины гиперграфа уже раскрашены, а именно i -я вершина раскрашена в цвет x_i , если $x_i \neq 0$, и не раскрашена ни в какой цвет, если $x_i = 0$.

Вам требуется определить, существует ли какая-нибудь хорошая раскраска графа, и если да, то вывести любую подходящую.

Формат входных данных

В первой строке указана пара чисел n и m ($1 \leq n \leq 300$, $1 \leq m \leq 2000$) — количество вершин и ребер в гиперграфе.

В последующих m строках указаны тройки чисел a_i, b_i, c_i ($1 \leq a_i < b_i < c_i \leq n$).

В последней строке указано n чисел $x_1, \dots, x_n$ ($0 \leq x_i \leq 3$).

Формат выходных данных

В первой строке выведите Yes или No, в зависимости от того, существует ли хорошая раскраска графа в 3 цвета.

В случае положительного ответа, в следующей строке выведите числа $y_1, \dots, y_n$ ($1 \leq y_i \leq 3$) — найденную раскраску.

Примеры

стандартный ввод	стандартный вывод
6 2 1 2 3 4 5 6 1 2 0 0 0 0	Yes 1 2 3 1 1 1
4 2 1 2 3 2 3 4 1 0 0 2	No

Задача I. Строим матрицу

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Нюша обожает троичную систему счисления. У неё была очень модная матрица A размера $N \times N$, заполненная числами 0, 1 и 2. Для некоторых подпрямоугольников этой матрицы Нюша вычислила произведения элементов по модулю 3. Формально, она посчитала

$$e_i = \left(\prod_{x=a_i}^{b_i} \prod_{y=c_i}^{d_i} A_{x,y} \right) \bmod 3.$$

Нюша рассказала Барашу все полученные значения, но во время прогулки потеряла свою модную матрицу и очень расстроилась.

Помогите Барашу восстановить какую-то матрицу A , для которой были бы такие же произведения в подпрямоугольниках, или сообщите, что это невозможно.

Формат входных данных

В первой строке входных данных даны два целых числа N и Q ($1 \leq N, Q \leq 2000$) — размер матрицы и количество подпрямоугольников. В каждой из следующих строк даны числа a_i, b_i, c_i, d_i, e_i ($1 \leq a_i \leq b_i \leq N$, $1 \leq c_i \leq d_i \leq N$, $0 \leq e_i \leq 2$) — координаты подпрямоугольника и произведение в нём.

Формат выходных данных

Если искомой матрицы не существует, в единственной строке выходных данных выведите «No».

Если искомая матрица существует, в первой строке выходных данных выведите «Yes». Затем выведите N строк по N чисел — какую-то подходящую матрицу A из 0, 1 и 2.

Примеры

стандартный ввод	стандартный вывод
2 3 1 2 1 2 1 1 2 1 1 2 1 1 1 2 2	Yes 1 2 2 1
3 4 1 2 1 2 0 2 3 1 3 1 1 2 2 2 2 1 2 2 3 2	Yes 0 1 2 1 2 2 2 2 1
1 3 1 1 1 1 0 1 1 1 1 1 1 1 1 1 2	No

Задача J. Лёшин взлом

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

Лёша нашел очень интересную игру-головоломку. В этой игре нужно замысловатым образом вскрывать замки. Замок состоит из N переключателей, расположенных вдоль одной прямой слева направо. Переключатель под номером i расположен в A_i нанометрах от левого края замка и находится в одном из двух состояний — включенном или выключенном. Цель Лёши — перевести все переключатели в выключенное состояние.

Для того чтобы добиться своей цели, у Лёши есть M специальных приспособлений для взлома. Приспособление под номером j при прикладывании к замку меняет состояние всех переключателей, находящихся на расстоянии от L_j до R_j нанометров от левого края замка. Таким образом, всякий переключатель, попавший в этот диапазон, из выключенного станет включенным, а из включенного наоборот выключенным.

Помогите Лёше справиться с очередным уровнем и опишите необходимые действия, или скажите, что это невозможно.

Формат входных данных

В первой строке даны два целых числа N и M ($1 \leq N \leq 10^5$, $1 \leq M \leq 2 \cdot 10^5$) — число переключателей и приспособлений для взлома.

В следующих N строках находится описание переключателей — пары целых чисел A_i и B_i ($1 \leq A_i \leq 10^9$, $0 \leq B_i \leq 1$) — расстояние от левого края замка до переключателя в нанометрах и состояние (1 обозначает включен, 0 обозначает выключен), соответственно.

В следующих M строках находится описание приспособлений для взлома — пары целых чисел L_j и R_j ($1 \leq L_j \leq R_j \leq 10^9$) — отрезок действия приспособления.

Формат выходных данных

Если уровень пройти невозможно, то выведите одно число -1 .

Иначе выведите число K ($0 \leq K \leq M$) и после этого еще K чисел $C_1 \dots C_K$ — различные номера используемых приспособлений.

Примеры

стандартный ввод	стандартный вывод
3 4 5 1 10 1 8 0 1 10 4 5 6 7 8 9	2 1 4
4 2 2 0 3 1 5 1 7 0 1 4 4 7	-1
3 2 5 0 10 0 8 0 6 9 66 99	0
7 7 6 1 3 0 4 1 7 1 1 1 5 1 2 0 2 6 1 3 1 5 6 7 1 4 2 6 4 6	4 3 4 6 7

Замечание

Как видно в третьем примере, множество использованных приспособлений может быть пустым.

Задача К. Кузнечик

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

Кузнечик двигается из клетки 0 в клетку n . За один прыжок он может прыгать на 1, 2 или 3 клетки вперед. Ваша задача — посчитать, сколько способов допрыгать у него есть. Ситуация усложняется тем, что иногда некоторые клетки становятся непроходимыми, а иногда наоборот. Вам нужно быстро пересчитывать число способов добраться из 0 в n после каждого изменения.

Формат входных данных

Первая строка содержит два числа n и m ($2 \leq n \leq 10^5$, $1 \leq m \leq 10^5$) — число клеток и число изменений. Следующие m строк содержат описания изменений. Каждое изменение описывается числом x ($1 \leq x \leq n - 1$), номером клетки, для которой изменился статус (то есть если она была проходимой, то стала непроходимой и наоборот).

Формат выходных данных

Выведите $m + 1$ число — число способов добраться из 0 до n в начале и после каждого изменения по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 4 1 2 1 2	4 2 1 2 4
100 2 33 56	347873931 85092847 684577937