

Задача А. Ваня и треугольники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Ване было скучно и он нарисовал n различных точек на плоскости. После этого он соединил попарно все точки и увидел, что в итоге образовалось большое количество треугольников с вершинами в отмеченных точках. Он просит вас посчитать количество образовавшихся треугольников с ненулевой площадью.

Формат входных данных

В первой строке находится целое число n ($1 \leq n \leq 2000$) — количество отмеченных точек на плоскости.

В следующих n строках находятся по два целых числа x_i, y_i ($-100 \leq x_i, y_i \leq 100$) — координаты i -й точки. Гарантируется, что среди точек нет совпадающих.

Формат выходных данных

В первой строке выведите целое число — количество треугольников с ненулевой площадью среди отмеченных.

Примеры

стандартный ввод	стандартный вывод
4 0 0 1 1 2 0 2 2	3
3 0 0 1 1 2 0	1
1 1 1	0

Замечание

Пояснение к первому тесту из условия. Образовано 3 треугольника: $(0, 0) - (1, 1) - (2, 0)$; $(0, 0) - (2, 2) - (2, 0)$; $(1, 1) - (2, 2) - (2, 0)$.

Пояснение ко второму тесту из условия. Образован 1 треугольник: $(0, 0) - (1, 1) - (2, 0)$.

Пояснение к третьему тесту из условия. Одна точка не образует ни одного треугольника.

Задача В. Двойная сортировка II

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

Даны две перестановки a и b , обе размера n . Перестановка размера n — это массив из n элементов, в который каждое целое число от 1 до n входит ровно один раз. Элементы в каждой перестановке пронумерованы от 1 до n .

Вы можете выполнять следующую операцию любое количество раз:

- выбрать целое число i от 1 до n ;
- обозначим за x такое целое число, что $a_x = i$. Поменять местами a_i и a_x ;
- обозначим за y такое целое число, что $b_y = i$. Поменять местами b_i и b_y .

Вы должны отсортировать обе перестановки **в порядке возрастания** (то есть должны выполняться условия $a_1 < a_2 < \dots < a_n$ и $b_1 < b_2 < \dots < b_n$) за **минимальное количество операций**. Обратите внимание, что обе перестановки должны быть отсортированы после того, как вы выполните выбранную вами последовательность операций.

Формат входных данных

В первой строке задано одно целое число n ($2 \leq n \leq 3000$).

Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$; все a_i различны).

В третьей строке заданы n целых чисел b_1, b_2, \dots, b_n ($1 \leq b_i \leq n$; все b_i различны).

Формат выходных данных

Сначала выведите одно целое число k ($0 \leq k \leq 2n$) — минимальное количество операций, за которое можно отсортировать обе перестановки. Обратите внимание: можно показать, что $2n$ операций всегда достаточно.

Затем выведите k целых чисел op_1, op_2, \dots, op_k ($1 \leq op_j \leq n$), где op_j — значение i , выбранное для j -й операции.

Если ответов несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
5 1 3 2 4 5 2 1 3 4 5	1 2
2 1 2 1 2	0
4 1 3 4 2 4 3 2 1	2 3 4

Задача С. Магазин сладостей

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

В магазине продаётся n типов конфет с номерами от 1 до n . Одна конфета типа i стоит b_i монет. Всего в магазине есть a_i конфет типа i .

Вам необходимо расфасовать все конфеты по пачкам, каждая пачка должна состоять только из конфет одного типа. Формально, для каждого типа конфет i вам необходимо выбрать число d_i , обозначающее количество конфет типа i в одной пачке, так, чтобы a_i делилось без остатка на d_i .

Тогда стоимость одной пачки конфет типа i будет равна $b_i \cdot d_i$. Обозначим эту стоимость за c_i , то есть $c_i = b_i \cdot d_i$.

После расфасовки пачки конфет будут помещены на полку. Рассмотрим стоимости пачек, расположенных на полке, в порядке c_1, c_2, \dots, c_n . Для описания стоимостей будут использоваться ценники. Один ценник может описать стоимость всех товаров от l до r включительно, если $c_l = c_{l+1} = \dots = c_r$. Каждый из товаров от 1 до n должен быть описан хотя бы одним ценником. К примеру, если $c_1, \dots, c_n = [4, 4, 2, 4, 4]$, для описания всех товаров будет достаточно 3 ценника, первый ценник описывает товары 1, 2, второй: 3, третий: 4, 5.

Вам даны числа $a_1, b_1, a_2, b_2, \dots, a_n, b_n$. Ваша задача выбрать числа d_i так, чтобы a_i делилось на d_i для всех i , и необходимое количество ценников для описания стоимостей c_1, c_2, \dots, c_n было минимально возможным.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 100\,000$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($2 \leq n \leq 200\,000$) — количество видов конфет.

Каждая из следующих n строк каждого набора входных данных содержит два целых числа a_i и b_i ($1 \leq a_i \leq 10^9$, $1 \leq b_i \leq 10\,000$) — количество конфет и стоимость одной конфеты типа i соответственно.

Гарантируется, что сумма n по всем наборам входных данных не превышает 200 000.

Формат выходных данных

Для каждого набора входных данных, выведите минимальное количество ценников необходимое для описания стоимостей всех товаров в магазине.

Пример

стандартный ввод	стандартный вывод
5	2
4	1
20 3	3
6 2	2
14 5	5
20 7	
3	
444 5	
2002 10	
2020 2	
5	
7 7	
6 5	
15 2	
10 3	
7 7	
5	
10 1	
11 5	
5 1	
2 2	
8 2	
6	
7 12	
12 3	
5 3	
9 12	
9 3	
1000000000 10000	

Замечание

В первом наборе входных данных можно выбрать $d_1 = 4$, $d_2 = 6$, $d_3 = 7$, $d_4 = 5$. Тогда стоимости товаров будут равны: $[12, 12, 35, 35]$. Для их описания хватит 2 ценников, первый ценник для c_1, c_2 и второй ценник для c_3, c_4 . Можно показать, что при любом корректном выборе d_i для описания всех товаров понадобится как минимум 2 ценника. Также обратите внимание, что этот пример иллюстрируется картинкой в условии задачи.

Во втором наборе входных данных при $d_1 = 4$, $d_2 = 2$, $d_3 = 10$ стоимости всех товаров будут равны 20. Таким образом 1 ценника хватит для описания всех товаров. Обратите внимание, что a_i делится на d_i для всех i , что является необходимым условием.

В третьем наборе входных данных нетрудно понять, что для описания 2, 3 и 4 товара может быть использован один ценник. И дополнительно понадобится по ценнику для товаров 1 и 5. Итого 3 ценника.

Задача D. Бесконечный забор

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

Вы являетесь лидером повстанцев и планируете начать революцию в вашей стране. Но злое Правительство узнало о ваших планах и решило наказать вас в форме исправительных работ.

Вы обязаны покрасить забор в два цвета, который состоит из 10^{100} досок, следующим образом (будем считать, что доски пронумерованы слева направо, начиная с 0):

- если номер доски делится на r (это доски с номерами $0, r, 2r$ и так далее), то вы обязаны покрасить ее в красный;
- если номер доски делится на b (это доски с номерами $0, b, 2b$ и так далее), то вы обязаны покрасить ее в синий;
- если номер доски делится и на r и на b , то **вы можете выбрать цвет**, в который покрасите эту доску;
- в противном случае, вам не надо красить доску совсем (тратить лишнюю краску в принципе запрещено).

Более того, Правительство добавило еще одно условие, чтобы усложнить вам задачу. Давайте выпишем номера всех **покрашенных** досок забора в порядке возрастания: если в данном списке найдется k последовательных досок одного цвета, то Правительство объявит вас лицом, неспособным к исправительным работам, и отправит вас на казнь. Если вы не покрасите забор, согласно заданным выше условиям, вас тоже казнят.

Вопрос в следующем: сможете ли вы выполнить работу (время выполнения не имеет значения) или же казнь неизбежна и вам необходимо сбежать любым способом.

Формат входных данных

В первой строке задано единственное число T ($1 \leq T \leq 1000$) — количество наборов входных данных.

В следующих T строках заданы описания наборов — по одному в строке. В каждом наборе заданы три целых числа r, b, k ($1 \leq r, b \leq 10^9, 2 \leq k \leq 10^9$) — соответствующие коэффициенты.

Формат выходных данных

Выведите T слов — по одному в строке. Для каждого набора выведите REBEL (регистр не важен), если казнь неизбежна или OBEY (регистр не важен), в противном случае.

Пример

стандартный ввод	стандартный вывод
4	OBEY
1 1 2	REBEL
2 10 4	OBEY
5 2 3	OBEY
3 2 2	

Задача Е. Максимумов должно быть много

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

Дано дерево (связный граф без циклов). В каждой вершине дерева записано число. Назовём характеристикой MAD (maximum double) дерева максимальное число, которое встречается в вершинах этого дерева **хотя бы** 2 раза. Если же никакое число не встречается в дереве больше одного раза, то положим $MAD = 0$.

Заметим, что если удалить ребро из дерева, то оно распадётся на два дерева. Вычислим MAD в каждом из деревьев и возьмем максимум из этих двух значений. Полученный результат назовем *значением* удаленного ребра.

Для каждого ребра найдите его значение. Обратите внимание, что мы в действительности не удаляем ребра из дерева, и каждое значение должно быть вычислено независимо.

Формат входных данных

В первой строке находится одно целое число n ($2 \leq n \leq 10^5$) — количество вершин в дереве.

Далее в $n - 1$ строке находятся по два целых числа u и v ($1 \leq u, v \leq n$) — концы рёбер дерева. Гарантируется, что данные ребра образуют дерево.

В последней строке находятся n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — числа, записанные в вершинах.

Формат выходных данных

Для каждого ребра в порядке ввода выведите одно число — максимум из MAD двух деревьев, получающихся после удаления из начального дерева данного ребра.

Примеры

стандартный ввод	стандартный вывод
5	0
1 2	2
2 3	1
2 4	2
1 5	
2 1 3 2 1	
6	1
1 2	1
1 3	0
1 4	1
4 5	1
4 6	
1 2 3 1 4 5	

Замечание

В первом примере после удаления ребра $(1, 2)$ ни в одном из получившихся поддеревьев никакое число не будет повторяться 2 раза, поэтому ответ равен $\max(0, 0) = 0$.

После удаления ребра $(2, 3)$ в большем поддереве будет два раза повторяться 1 и два раза повторяться 2, поэтому MAD этого дерева будет равен 2.

После удаления ребра $(2, 4)$ в большем поддереве будет повторяться только число 1, а во втором будет только одно число, поэтому ответом будет 1.

Во втором примере, если ребро $1 \leftrightarrow 4$ не удалено, то в одном из поддеревьев будет две 1, поэтому ответ — 1. А если удалено ребро $1 \leftrightarrow 4$, то в обоих поддеревьях нет повторяющихся значений, поэтому ответом будет 0.

Задача F. Li Hua и дерево

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

У Li Hua есть дерево с n вершинами и $n - 1$ ребром. Корнем дерева является вершина 1. Каждая вершина i имеет важность a_i . Обозначим *размер* поддерева как количество вершин в нем, а *важность* как сумму важности вершин в нем. Обозначим *тяжелым сыном* нелистой вершины сына с **наибольшим** *размером* поддерева. Если таких вершин несколько, то *тяжелым сыном* будет та, у которой индекс **минимален**.

Li Hua хочет выполнить m операций:

- «1 x » ($1 \leq x \leq n$) — вычислить *важность* поддерева, корнем которого является x .
- «2 x » ($2 \leq x \leq n$) — повернуть *тяжелого сына* поддерева x вверх. Формально, обозначим как son_x *тяжелого сына* x , fa_x как отца x . Он хочет удалить ребро между x и fa_x и добавить ребро между son_x и fa_x . Гарантируется, что x не является корнем, но **не** гарантируется, что x не является листом. Если x является листом, проигнорируйте эту операцию.

Предположим, что вы Li Hua. Пожалуйста, решите эту задачу.

Формат входных данных

Первая строка содержит 2 целых числа n, m ($2 \leq n \leq 10^5, 1 \leq m \leq 10^5$) — количество вершин в дереве и количество операций.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — важность каждой вершины.

Следующие $n - 1$ строк содержат рёбра дерева. В i -й строке записаны два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$) — соответствующее ребро. Данные рёбра образуют дерево.

Следующие m строк содержат операции — по одной операции в строке. j -я операция содержит два целых числа t_j, x_j ($t_j \in \{1, 2\}, 1 \leq x_j \leq n, x_j \neq 1$ если $t_j = 2$) — j -я операция.

Формат выходных данных

Для каждого запроса вида «1 x » выведите ответ в отдельной строке.

Примеры

стандартный ввод	стандартный вывод
7 4 1 1 1 1 1 1 1 1 2 1 3 2 4 2 5 3 6 6 7 1 6 2 3 1 6 1 2	2 3 3
10 14 -160016413 -90133231 -671446275 -31484757914810579234 121155052 -359359950 83112406 -70488962 1 6 1 10 10 8 1 4 3 4 2 7 2 5 3 2 9 8 1 4 2 2 2 4 1 4 1 10 2 10 1 9 1 6 2 8 2 10 1 5 1 8 1 1 2 5	-2346335269 -476287915 -704889624 121155052 -1360041415 228601709 -2861484545

Задача G. Простой подарок

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3.5 секунд
Ограничение по памяти:	256 мегабайт

- Мужик, у тебя простые числа есть?
- Нету.
- На, мужик, простые числа.

Пингвин

В отличие от Гриши, который вел себя хорошо, Олег за весь год так и не научился решать задачи на теорию чисел. Поэтому вместо Деда Мороза к нему пришел его сокомандник Андрей и торжественно вручил ему множество из n **различных простых** чисел вместе с простой задачей: Олегу необходимо найти k -е в порядке возрастания положительное целое число, среди простых делителей которого встречаются числа **только** из этого множества.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 16$) — количество чисел во множестве.

Во второй строке n различных простых чисел p_1, p_2, \dots, p_n ($2 \leq p_i \leq 100$) — числа во множестве. Гарантируется, что числа идут в порядке возрастания.

В третьей строке задано число k ($1 \leq k$). Гарантируется, что k -е в порядке возрастания положительное целое число, среди простых делителей которого встречаются числа только из данного множества не превосходит 10^{18} .

Формат выходных данных

Выведите k -е в порядке возрастания число, удовлетворяющее условию. Гарантируется, что оно не превосходит 10^{18} .

Замечание

Для первого примера последовательность с простыми делителями из набора $\{2, 3, 5\}$ выглядит так:

(1, 2, 3, 4, 5, 6, 8, ...)

Седьмым по счету числом (в 1-индексации) как раз и будет восьмерка.

Задача Н. Помочь Хиасату

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

Хиасат недавно зарегистрировал новый аккаунт на сайте NeskoForces и, как только его друзья об этом узнали, каждый из них захотел, чтобы его имя было названием хендла профиля Хиасата.

К счастью для Хиасата, он может менять свой профиль в некоторые моменты времени. Также он знает моменты времени, когда его друзья будут заходить на его страницу. Формально, дана последовательность событий двух видов:

- 1 — Хиасат может поменять свой хендл.
- $2 s$ — друг s посещает профиль Хиасата.

Друг s будет счастлив, если каждый раз, когда он посетит страницу Хиасата, его хендл будет равен s .

Хиасат просит вас помочь ему, выясните наибольшее количество счастливых друзей, которого можно добиться.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n \leq 10^5, 1 \leq m \leq 40$) — количество событий и количество друзей.

Затем следуют n строк, задающих событие одно из двух видов:

- 1 — Хиасат может поменять свой хендл.
- $2 s$ — друг s ($1 \leq |s| \leq 40$) посещает профиль Хиасата.

Гарантируется, что имя каждого друга состоит только из строчных латинских букв.

Гарантируется, что первое событие обязательно первого типа, и что каждый друг посетит профиль Хиасата хотя бы один раз.

Формат выходных данных

Выведите одно целое число — наибольшее число счастливых друзей.

Примеры

стандартный ввод	стандартный вывод
5 3 1 2 motarack 2 mike 1 2 light	2
4 3 1 2 alice 2 bob 2 tanyaromanova	1
6 4 1 2 a 2 b 1 2 c 2 d	2

Замечание

В первом примере, оптимально поменять хендл на «motarack» в первом событии и на «light» в четвёртом. Таким образом, «motarack» и «light» будут счастливы, а «mike» — нет.

Во втором примере можно выбрать «alice», «bob» или «tanyaromanova», и ровно этот друг и будет счастлив.

Задача I. Длинные ноги

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

Робот стоит в клетке $(0, 0)$ бесконечной сетки. Длина его ног может регулироваться. Изначально длина его ног равна 1.

Пусть робот сейчас стоит в клетке (x, y) , а длина его ног равна m . За один ход он может исполнить одно из следующих трех действий:

- прыгнуть в клетку $(x + m, y)$;
- прыгнуть в клетку $(x, y + m)$;
- увеличить длину ног на 1, то есть сделать ее равной $m + 1$.

Какое наименьшее количество ходов ему потребуется, чтобы достичь клетки (a, b) ?

Формат входных данных

В первой строке записано одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных.

В единственной строке каждого набора входных данных записаны два целых числа a and b ($1 \leq a, b \leq 10^9$) — финальная клетка.

Формат выходных данных

На каждый набор входных данных выведите одно целое число — наименьшее количество ходов, которые потребуются роботу, чтобы достичь клетки (a, b) из клетки $(0, 0)$.

Пример

стандартный ввод	стандартный вывод
3	2
1 1	5
1 6	6
8 4	

Замечание

В первом наборе входных данных робот может сначала прыгнуть в $(0, 1)$, затем в $(1, 1)$. Если он когда-нибудь увеличит длину своих ног, то он сможет только перепрыгнуть $(1, 1)$.

Во втором наборе робот может прыгнуть в $(1, 0)$, затем увеличить длину своих ног до 2, затем прыгнуть трижды до $(1, 6)$.

В третьем наборе робот может увеличить длину своих ног трижды, чтобы сделать ее равной 4. Затем прыгнуть в $(0, 4)$. Затем прыгнуть дважды до $(8, 4)$.

Задача J. Макс клика

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

Макс, опытейший игрок в доту, постоянно кликал.

Дан случайный неориентированный граф G из n вершин и m ребер.

Подкликой называется такое подмножество вершин $A: \forall a, b \in A, a \neq b \exists$ ребро (a, b) .

Ваша задача — найти подклику $A: |A|$ максимально.

Формат входных данных

На первой строке число вершин $n \geq 1$ и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходных данных

На первой строке выведите k — количество вершин в максимальной подклике. На следующей строке k целых чисел от 1 до n — номера вершин в подклике. Вершины можно выводить в любом порядке. Если максимальных подклик несколько, выведите любую.

Система оценки

Подзадача 1 (10 баллов) $n \leq 20$.

Подзадача 2 (10 баллов) $n \leq 25$.

Подзадача 3 (20 баллов) $n \leq 40$.

Подзадача 4 (20 баллов) $n \leq 60$.

Подзадача 5 (20 баллов) $n \leq 70$.

Подзадача 6 (20 баллов) $n \leq 80$.

Пример

стандартный ввод	стандартный вывод
5 8	4
5 4	3 5 1 4
3 5	
1 5	
1 3	
2 3	
1 4	
5 2	
3 4	

Задача К. Сокровища

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

Дочь короля Флатландии собирается выйти за прекрасного принца. Принц хочет подарить принцессе сокровища, но он не уверен какие именно бриллианты из своей коллекции выбрать.

В коллекции принца n бриллиантов, каждый характеризуется весом w_i и стоимостью v_i . Принц хочет подарить наиболее дорогие бриллианты, однако король умен и не примет бриллиантов суммарного веса больше R . С другой стороны, принц будет считать себя жадным всю оставшуюся жизнь, если подарит бриллиантов суммарным весом меньше L .

Помогите принцу выбрать набор бриллиантов наибольшей суммарной стоимости, чтобы суммарный вес был в отрезке $[L, R]$.

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 32$), L и R ($0 \leq L \leq R \leq 10^{18}$). Следующие n строк описывают бриллианты и содержит по два числа — вес и стоимость соответствующего бриллианта ($1 \leq w_i, v_i \leq 10^{15}$).

Формат выходных данных

Первая строка вывода должна содержать k — количество бриллиантов, которые нужно подарить принцессе. Вторая строка должна содержать номера даримых бриллиантов.

Бриллианты нумеруются от 1 до n в порядке появления во входных данных.

Если составить подарок принцессе невозможно, то выведите 0 в первой строке вывода.

Пример

стандартный ввод	стандартный вывод
3 6 8	1
3 10	2
7 3	
8 2	

Задача L. Потерянный массив

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

Маленькому Леше улыбнулась удача: ему удалось раздобыть очень редкий и необычный массив A , состоящий из n целых чисел. Заполучив в свои руки такой ценный предмет, мальчик сразу же отнес его в свою подземную лабораторию, намереваясь постичь тайны устройства массива.

Как опытный исследователь, Леша решил не спешить с проведением жутких экспериментов над бедным массивом, а сперва понаблюдать за ним. Он записал на листик бумаги результаты m наблюдений за массивом. Каждое наблюдение описывается тремя числами l_i , r_i и s_i , которые означают, что сумма элементов по модулю 2 на подотрезке $[l_i, r_i]$ массива равна s_i . Леша подготовил все необходимое для проведения $m+1$ -го наблюдения, но тут он услышал, что мама зовет его обедать.

По возвращении в свою лабораторию, мальчик увидел страшное: массив исчез. Совсем. Все надежды мальчика на получение Нобелевской премии в области исследования массивов рассыпались на глазах. Но потом Леша вспомнил, что у него все же осталась некоторая информация о массиве. Основываясь на записанных фактах, он решил построить компьютерную модель массива и проводить эксперименты над ней. Конечно, мальчик понимал, что он не сможет однозначно восстановить утерянный массив, потому что существует бесконечно много массивов, удовлетворяющих записанным фактам.

Все мы знаем, что операции с большими числами занимают много времени, а если еще к тому же числа становятся отрицательными... Чтобы сделать свою компьютерную модель как можно более быстрой, Леша решил построить массив, удовлетворяющий известным фактам, элементы которого неотрицательны, а их сумма минимально возможная.

Помогите ему сделать это!

Формат входных данных

Первая строка входных данных содержит числа n и m - количество элементов в массиве и количество фактов. Следующие m строк содержат числа l_i , r_i и s_i - описание i -го факта.

$$1 \leq n \leq 40$$

$$0 \leq m \leq 100$$

$$1 \leq l_i \leq r_i \leq n$$

$$0 \leq s_i \leq 1$$

$$0 \leq a_i \leq 10^9$$

Формат выходных данных

Вывод должен содержать n целых чисел a_i — элементы восстановленного массива. Если существует несколько возможных ответов, выведете лексикографически минимальных среди них. Массив X лексикографически меньше массива Y если существует такой индекс $k \leq |X|$, что $x_i = y_i$ для всех $i < k$ и $x_k < y_k$.

Пример

стандартный ввод	стандартный вывод
3 3 2 2 1 3 3 0 2 3 1	0 1 0

Задача М. Необычная сортировка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Вам дана последовательность различных целых чисел a_1, a_2, \dots, a_n . Все числа в последовательности лежат в отрезке $[0, 2^k - 1]$, где k дано.

Давайте определим функцию $f(x)$ для числа x , которое лежит в отрезке $[0, 2^k - 1]$ как количество инверсий в последовательности $a_1 \oplus x, a_2 \oplus x, \dots, a_n \oplus x$.

Теперь давайте отсортируем все целые числа из отрезка $[0, 2^k - 1]$ по возрастанию значений функции f , а затем по возрастанию самих чисел.

Вам дана позиция p . Найдите p -е число в этом порядке сортировки всех целых чисел из отрезка $[0, 2^k - 1]$.

Формат входных данных

В первой строке находится единственное целое число t ($1 \leq t \leq 20$) — количество тестовых случаев. Далее находится описание t тестовых случаев в следующем формате:

Первая строка содержит три целых числа n, k, p , разделенных пробелом ($1 \leq n \leq 10^6, 1 \leq k \leq 30, 1 \leq p \leq 2^k$) — количество чисел в последовательности, параметр k и заданная позиция.

В следующей строке находятся n различных целых чисел a_1, a_2, \dots, a_n , разделенных пробелами ($0 \leq a_i < 2^k$).

Сумма всех n во всех тестовых случаях не превосходит 10^6 .

Формат выходных данных

Для каждого тестового случая выведите единственное целое число — p -е целое число в порядке сортировки всех целых чисел из отрезка $[0, 2^k - 1]$, описанном в условии задачи.

Пример

стандартный ввод	стандартный вывод
2	4
4 3 5	2
2 0 3 7	
2 2 1	
2 0	