

Задача А. Система непересекающихся множеств

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

Напишите программу, которая будет содержать реализацию структуры данных для совокупности непересекающихся подмножеств (disjoint sets) и обрабатывать запросы таких видов:

RESET n — создать новую серию подмножеств: множество из одного только элемента 0, из одного только элемента 1, и так до множества из одного только элемента $n-1$ включительно. Если структура уже содержала какую-то другую совокупность непересекающихся подмножеств, вся соответствующая информация утрачивается. На стандартный выход (экран) при этом следует вывести два слова через пробел «RESET DONE».

JOIN j k — объединить подмножества, которым принадлежат элемент j и элемент k . Если элементы и так принадлежали одному подмножеству, вывести на стандартный выход (экран) слово «ALREADY», после него через пробелы те же числа j и k в том же порядке. Если элементы до сих пор принадлежали разным подмножествам, то действие происходит только с данными в памяти, на экран ничего не выводится.

CHECK j k — проверить, одному ли подмножеству принадлежат элемент j и элемент k ; вывести на стандартный выход (экран) слово «YES» (если одному) или слово «NO» (если разным).

Формат входных данных

Во входных данных содержится последовательность запросов RESET, JOIN и CHECK — каждый в отдельной строке, согласно вышеописанному формату. Гарантированно, что первая строка содержит запрос RESET, а общее количество запросов RESET не превышает 5. Общее количество всех запросов не превышает 200000. Значение n в каждом запросе RESET не превышает 100000. В каждом запросе JOIN и в каждом запросе CHECK оба числа будут в диапазоне от 0 до $n-1$, где n — параметр последнего выполненного запроса RESET.

Формат выходных данных

Для запросов RESET, CHECK и тех запросов JOIN, где элементы и так принадлежат одному подмножеству, выводить на стандартный выход (экран) соответствующий результат (в отдельной строке).

Пример

стандартный ввод	стандартный вывод
RESET 15	RESET DONE
JOIN 14 10	NO
JOIN 13 8	ALREADY 4 1
JOIN 0 9	NO
JOIN 8 3	YES
JOIN 4 1	
JOIN 10 5	
JOIN 8 4	
CHECK 2 11	
JOIN 4 1	
JOIN 2 6	
CHECK 9 1	
JOIN 6 5	
CHECK 10 5	

Замечание

Ответы «NO» даются на запросы «CHECK 2 11» и «CHECK 9 1», ответ «ALREADY 4 1» — на второй из запросов «JOIN 4 1» (10-я строка), «YES» — на «CHECK 5 10».

Задача В. Вес компоненты

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

В неориентированный взвешанный граф добавляют ребра. Напишите программу, которая в некоторые моменты находит сумму весов ребер в компоненте связности.

Формат входных данных

В первой строке записано два числа n и m ($1 \leq n \leq 10^5$, $1 \leq m \leq 5 \cdot 10^5$) - количество вершин в графе и количество производимых добавлений и запросов. Далее следует m строк с описанием добавления или запроса. Каждая строка состоит из двух или четырех чисел. Первое из чисел обозначает код операции. Если первое число 1, то за ним следует еще три числа x , y , w . Это означает, что в граф добавляется ребро из вершины x в вершину y веса w . ($1 \leq x < y \leq n$, $1 \leq w \leq 10^3$). Кратные ребра допустимы. Если первое число 2, то за ним следует ровно одно число x . Это означает, что необходимо ответить на вопрос, какова сумма ребер в компоненте связности, которой принадлежит вершина x ($1 \leq x \leq n$).

Формат выходных данных

Для каждой операции с кодом 2 выведите ответ на поставленную задачу. Ответ на каждый запрос выводите на отдельной строке.

Пример

стандартный ввод	стандартный вывод
6 10	0
2 1	1
1 1 2 1	3
2 1	6
1 2 4 2	3
2 1	0
1 1 4 3	
2 1	
1 3 5 3	
2 5	
2 6	

Задача С. Разрезанные таблицы (простые)

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

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Формат входных данных

В первой строке входного файла даны два натуральных числа n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 10^5$) — количество элементов в массиве и количество запросов. Вторая строка содержит n натуральных числа a_i .

Следующие m строк содержат запросы.

Формат выходных данных

В выходной файл выведите ответ на каждый запрос.

Примеры

стандартный ввод	стандартный вывод
5 5	4
7 9 9 4 5	7
3 3	7
0 2	7
0 2	4
0 0	
0 4	

Задача D. Разреженные таблицы

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

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Формат входных данных

В первой строке входного файла даны три натуральных числа n , m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^7$) и a_1 ($0 \leq a_1 < 16\,714\,589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leq u_1, v_1 \leq n$) — первый запрос.

Элементы a_2, a_3, \dots, a_n задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при $n = 10$, $a_1 = 12345$ получается следующий массив: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

Запросы генерируются следующим образом:

$$\begin{aligned} u_{i+1} &= ((17 \cdot u_i + 751 + ans_i + 2i) \bmod n) + 1, \\ v_{i+1} &= ((13 \cdot v_i + 593 + ans_i + 5i) \bmod n) + 1, \end{aligned}$$

где ans_i — ответ на запрос номер i .

Обратите внимание, что u_i может быть больше, чем v_i .

Формат выходных данных

В выходной файл выведите u_m , v_m и ans_m (последний запрос и ответ на него).

Примеры

стандартный ввод	стандартный вывод
10 8 12345 3 9	5 3 1565158

Задача E. RSQ

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

Формат входных данных

В первой строке находится число n — размер массива. ($1 \leq n \leq 500\,000$) Во второй строке находится n чисел a_i — элементы массива. Далее содержится описание операций, их количество не превышает 1 000 000. В каждой строке находится одна из следующих операций:

- `set i x` — установить $a[i]$ в x .
- `sum i j` — вывести значение суммы элементов в массиве на отрезке с i по j , гарантируется, что $(1 \leq i \leq j \leq n)$.

Все числа во входном файле и результаты выполнения всех операций не превышают по модулю 10^{18} .

Формат выходных данных

Выведите последовательно результат выполнения всех операций `sum`. Следуйте формату выходного файла из примера.

Пример

<code>rsq.in</code>	<code>rsq.out</code>
5	14
1 2 3 4 5	15
sum 2 5	10
sum 1 5	9
sum 1 4	12
sum 2 4	22
set 1 10	20
set 2 3	10
set 5 2	
sum 2 5	
sum 1 5	
sum 1 4	
sum 2 4	

Задача F. Звезды

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

Вася любит наблюдать за звездами. Но следить за всем небом сразу ему тяжело. Поэтому он наблюдает только за частью пространства, ограниченной кубом размером $n \times n \times n$. Этот куб поделен на маленькие кубики размером $1 \times 1 \times 1$. Во время его наблюдений могут происходить следующие события:

1. В каком-то кубике появляются или исчезают несколько звезд.
2. К нему может заглянуть его друг Петя и поинтересоваться, сколько видно звезд в части пространства, состоящей из нескольких кубиков.

Формат входных данных

Первая строка входного файла содержит натуральное число $1 \leq n \leq 128$. Координаты кубиков — целые числа от 0 до $n - 1$. Далее следуют записи о происходивших событиях по одной в строке. В начале строки записано число m . Если m равно:

- 1, то за ним следуют 4 числа — x, y, z ($0 \leq x, y, z < N$) и k ($-20000 \leq k \leq 20000$) — координаты кубика и величина, на которую в нем изменилось количество видимых звезд;
- 2, то за ним следуют 6 чисел — $x_1, y_1, z_1, x_2, y_2, z_2$ ($0 \leq x_1 \leq x_2 < N, 0 \leq y_1 \leq y_2 < N, 0 \leq z_1 \leq z_2 < N$), которые означают, что Петя попросил подсчитать количество звезд в кубиках (x, y, z) из области: $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z_1 \leq z \leq z_2$;
- 3, то это означает, что Васе надоело наблюдать за звездами и отвечать на вопросы Пети. Эта запись встречается во входном файле только один раз и будет последней.

Количество записей во входном файле не больше 100 002.

Формат выходных данных

Для каждого Петиного вопроса выведите искомое количество звезд.

Пример

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

Задача G. Разрезание графа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 512 мегабайт

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

cut - разрезать граф, то есть удалить из него ребро;

ask - проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа cut рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа ask.

Формат входных данных

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа n , количество рёбер m и количество операций k ($1 \leq n \leq 50000$, $0 \leq m \leq 100000$, $m \leq k \leq 150000$).

Следующие m строк задают рёбра графа; i -я из этих строк содержит два числа u_i и v_i ($1 \leq u_i, v_i \leq n$), разделённые пробелами - номера концов i -го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют k строк, описывающих операции. Операция типа cut задаётся строкой "cut u v " ($1 \leq u, v \leq n$), которая означает, что из графа удаляют ребро между вершинами u и v . Операция типа ask задаётся строкой "ask u v " ($1 \leq u, v \leq n$), которая означает, что необходимо узнать, лежат ли в данный момент вершины u и v в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа cut ровно один раз.

Формат выходных данных

Для каждой операции ask во входном файле выведите на отдельной строке слово YES, если две указанные вершины лежат в одной компоненте связности, и NO в противном случае. Порядок ответов должен соответствовать порядку операций ask во входном файле.

Пример

стандартный ввод	стандартный вывод
3 3 7	YES
1 2	YES
2 3	NO
3 1	NO
ask 3 3	
cut 1 2	
ask 1 2	
cut 1 3	
ask 2 1	
cut 2 3	
ask 3 1	

Задача Н. Острова

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

Одно разбросанное на островах Океании государство решило создать сеть автомобильных дорог (вернее, мостов). По каждому мосту можно перемещаться в обе стороны. Был разработан план очередности строительства мостов и известно, что после постройки всех мостов можно будет проехать по ним с каждого острова на каждый (возможно, через некоторые промежуточные острова

Однако, этот момент может наступить до того, как будут построены все мосты. Вам необходимо определить такое минимальное количество мостов, после строительства которых (в порядке, определенном планом), можно будет попасть с любого острова на любой другой.

Формат входных данных

Первая строка содержит два числа: число островов N ($1 \leq N \leq 10000$) и количество мостов в плане M ($1 \leq M \leq 50000$). Далее идет M строк, каждая содержит два числа x и y ($1 \leq x, y \leq N$) - номера городов, которые соединяет очередной мост в плане.

Формат выходных данных

Программа должна вывести единственное число - минимальное количество построенных мостов, после которого можно будет попасть с любого острова на любой другой.

Пример

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

Задача I. Уничтожение массива

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Вам дан массив, состоящий из n неотрицательных чисел a_1, a_2, \dots, a_n .

В этом массиве один за другим зачеркиваются числа. Вам задана перестановка чисел от 1 до n — порядок, в котором это происходит.

После зачеркивания очередного числа вам необходимо найти в этом массиве подотрезок с максимальной суммой, не содержащий ни одного зачеркнутого числа. Сумму чисел в пустом подотрезке считайте равной 0.

Формат входных данных

В первой строке входных данных записано число n ($1 \leq n \leq 100000$) — длина массива. Во второй строке записаны n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$). В третьей строке входных данных записана перестановка чисел от 1 до n — порядок, в котором зачеркиваются числа.

Формат выходных данных

В выходной файл выведите n строк, каждая из которых должна содержать одно число — максимальную сумму на подотрезке заданного массива, не содержащем зачеркнутых чисел, после выполнения очередного действия.

Примеры

стандартный ввод	стандартный вывод
4 1 3 2 5 3 4 1 2	5 4 3 0
5 1 2 3 4 5 4 2 3 5 1	6 5 5 1 0
8 5 5 4 4 6 6 5 5 5 2 8 7 1 3 4 6	18 16 11 8 8 6 6 0
10 10 4 9 0 7 5 10 3 10 9 5 2 8 1 3 9 6 10 4 7	37 37 19 19 19 15 10 10 10 0

Задача J. Прямоугольники

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Когда-то тут была легенда про вёдра, но её съели.

Жюри олимпиады

Есть таблица T размера $N \times M$. Элементами таблицы являются прямоугольники T_{ij} , где $0 \leq i < N$ и $0 \leq j < M$. Прямоугольник T_{ij} задаётся четвёркой чисел $(x_1^{ij}, y_1^{ij}, x_2^{ij}, y_2^{ij})$, где (x_1^{ij}, y_1^{ij}) и (x_2^{ij}, y_2^{ij}) — координаты противоположных углов прямоугольника. Стороны прямоугольника параллельны осям координат.

Далее вам поступают запросы. Каждый запрос состоит из четырёх чисел: (r_1, c_1, r_2, c_2) . Ответом на такой запрос является площадь фигуры, являющейся пересечением всех прямоугольников T_{ij} таких, что $\min(r_1, r_2) \leq i \leq \max(r_1, r_2)$ и $\min(c_1, c_2) \leq j \leq \max(c_1, c_2)$. Запросов очень много, поэтому мы просим вас вывести сумму ответов на все запросы по модулю $10^9 + 7$.

Формат входных данных

В первой строке записаны два целых числа N и M — размеры таблицы T ($1 \leq N, M \leq 127$). Далее в N строках описывается таблица T : в $(i + 1)$ -й строке $(j + 1)$ -я четвёрка чисел $x_1^{ij} y_1^{ij} x_2^{ij} y_2^{ij}$ описывает прямоугольник T_{ij} . Гарантируется, что $|x_k^{ij}|, |y_k^{ij}| \leq 10^6$.

Дальше в отдельной строке записано четыре числа. Первое из них, число Q — количество запросов ($1 \leq Q \leq 5 \cdot 10^6$). Следующие три числа — это A, B, v_0 ($0 \leq A, B, v_0 < 10^9 + 7$). При помощи этих чисел генерируется бесконечная последовательность $\{v_i\}$ по правилу $v_i = (A \cdot v_{i-1} + B) \bmod (10^9 + 7)$.

После этого k -й запрос (запросы нумеруются с единицы) задаётся следующей четвёркой чисел: $(v_{4k-3} \bmod N, v_{4k-2} \bmod M, v_{4k-1} \bmod N, v_{4k} \bmod M)$.

Формат выходных данных

Выведите сумму ответов на все запросы по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
2 2 0 0 2 2 1 1 3 3 0 3 2 1 1 2 3 0 1 500000003 4 2	1
3 2 8 -1 -7 6 6 8 9 10 -4 -10 4 9 -3 -8 6 9 -2 -9 3 8 -5 7 7 3 5 303164476 273973578 65779139	85

Замечание

В первом примере запрос имеет вид $(1, 0, 0, 1)$, то есть это запрос ко всей таблице. Пересечением всех прямоугольников является квадрат с углами в точках $(1, 1)$ и $(2, 2)$. Его площадь равна 1.

Во втором примере запросы имеют вид $(0, 1, 1, 1)$, $(1, 0, 2, 0)$, $(0, 0, 2, 1)$, $(0, 1, 1, 1)$, $(0, 1, 0, 0)$. На второй запрос ответ — 85, на остальные — 0.

Задача К. Дороги не только в Берляндии

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Правительство Берляндии решило улучшать отношения с соседними странами. В первую очередь было решено построить новые дороги так, чтобы из всех городов Берляндии и соседних стран можно было добраться до всех остальных. Всего в Берляндии и соседних странах n городов и ровно $n - 1$ двухсторонних дорог. Из-за недавнего финансового кризиса правительство Берляндии сильно стеснено в средствах, поэтому, чтобы построить новую дорогу, оно вынуждено закрывать какую-то из существующих. За один день можно закрыть одну существующую дорогу и сразу же построить новую. Ваша задача — определить, какое минимальное количество дней потребуется, чтобы из любого города можно было добраться до любого другого, а так же составить план закрытия старых и постройки новых дорог.

Формат входных данных

В первой строке входных данных записано целое число n ($2 \leq n \leq 1000$) — число городов в Берляндии и соседних странах. Далее в $n - 1$ строках записаны описания дорог. Каждая дорога описывается двумя целыми числами a_i, b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$) — пара городов, которые дорога соединяет. Между каждой парой городов не может быть больше одной дороги.

Формат выходных данных

Выведите число t —наименьшее требуемое число дней для перестройки дороги таким образом, чтобы из любого города можно было добраться до любого другого. Далее выведите t строк — план закрытия старых и постройки новых дорог. Каждая строка должна описывать действия в очередной день в формате $i j u v$ —закрытие дороги между городами i, j и постройка новой дороги между городами u, v . Города нумеруются, начиная с 1. Если решений несколько, выведите любое.

Примеры

стандартный ввод	стандартный вывод
2 1 2	0
7 1 2 2 3 3 1 4 5 5 6 6 7	1 3 1 1 4

Задача L. Вечеринка

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

Умный Бобёр решил устроить вечеринку. У Бобра много знакомых, и некоторые из них дружат друг с другом, а некоторые друг другу не нравятся. Чтобы вечеринка удалась на славу, Умный Бобёр хочет пригласить только тех своих знакомых, которые дружат, и не приглашать тех, кто не нравится друг другу. Отношения дружбы и антипатии симметричны.

Более формально, для каждого приглашённого человека должны выполняться следующие условия:

- все его друзья должны быть также приглашены на вечеринку;
- среди приглашенных не должно быть людей, которые ему не нравятся;
- все приглашенные на вечеринку должны быть связаны с ним дружбой напрямую или через цепь общих друзей произвольной длины. Будем говорить, что люди a_1 и a_p связаны цепью общих друзей, если существует последовательность людей a_2, a_3, \dots, a_{p-1} такая, что все пары людей a_i и a_{i+1} ($1 \leq i < p$) — друзья.

Помогите Бобру определить максимальное количество знакомых, которых он может пригласить.

Формат входных данных

В первой строке входных данных записано целое число n ($2 \leq n \leq 2000$) — количество знакомых Бобра. Во второй строке записано целое число k ($0 \leq k \leq \min(10^5, \frac{n \cdot (n-1)}{2})$) — количество пар друзей. В следующих k строках через пробел записаны пары чисел u_i, v_i ($1 \leq u_i, v_i \leq n; u_i \neq v_i$) — номера людей, которые входят в i -ю пару друзей. В следующей строке записано число m ($0 \leq m \leq \min(10^5, \frac{n \cdot (n-1)}{2})$) — количество пар людей, которые друг другу не нравятся. В следующих m строках перечислены пары таких людей в том же формате, что и пары друзей. Каждая пара людей упоминается во входных данных не более одного раза ($0 \leq k + m \leq \frac{n \cdot (n-1)}{2}$). В частности, два человека не могут быть друзьями и одновременно не нравиться друг другу.

Формат выходных данных

Выведите единственное число — максимальное количество людей, которых Бобер сможет пригласить на вечеринку. Если группу людей, удовлетворяющую всем требованиям, выбрать невозможно, выведите 0.

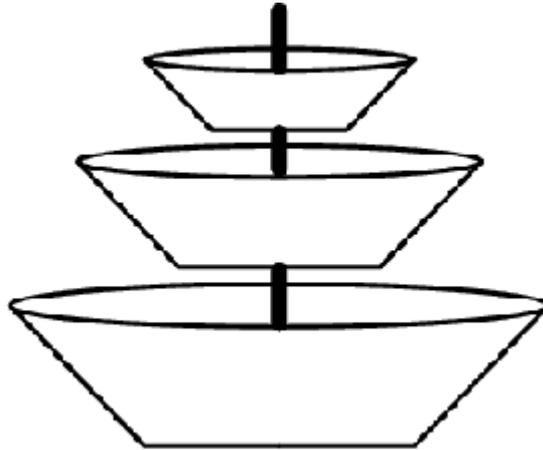
Пример

стандартный ввод	стандартный вывод
9	3
8	
1 2	
1 3	
2 3	
4 5	
6 7	
7 8	
8 9	
9 6	
2	
1 6	
7 9	

Задача М. Сосуды

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

Есть система из n сосудов, расположенных один над другим так, как показано на рисунке ниже. Представим, что сосуды пронумерованы числами от 1 до n в порядке от самого верхнего к самому нижнему, объем i -го сосуда равен a_i литров.



Изначально все сосуды пусты. В некоторые сосуды наливают воду, при этом вся вода, не поместившаяся в i -й сосуд, переливается в $(i + 1)$ -й. Жидкость, не поместившаяся в n -й сосуд, выливается на пол. Ваша задача состоит в том, чтобы промоделировать наливание воды в сосуды. Для этого вам потребуется обрабатывать два типа запросов:

1. Долить в p_i -й сосуд x_i литров воды;
2. Вывести, сколько воды находится в k_i -м сосуде.

При ответе на второй вопрос можно считать, что вся вода, налитая до этого момента, уже успела перелиться между сосудами.

Формат входных данных

В первой строке записано целое число n — количество сосудов ($1 \leq n \leq 2 \cdot 10^5$). Во второй строке записаны n целых чисел a_1, a_2, \dots, a_n — вместимости сосудов ($1 \leq a_i \leq 10^9$). Вместимости сосудов не обязательно возрастают от верхних сосудов к нижним (см. второй пример.) В третьей строке записано целое число m — количество запросов ($1 \leq m \leq 2 \cdot 10^5$). Каждая из следующих m строк содержит описание одного запроса. Запрос первого типа задается как «1 $p_i x_i$ », запрос второго типа — как «2 k_i » ($1 \leq p_i \leq n, 1 \leq x_i \leq 10^9, 1 \leq k_i \leq n$).

Формат выходных данных

На каждый запрос второго типа в отдельной строке выведите количество воды в соответствующем сосуде.

Примеры

стандартный ввод	стандартный вывод
2 5 10 6 1 1 4 2 1 1 2 5 1 1 4 2 1 2 2	4 5 8
3 5 10 8 6 1 1 12 2 2 1 1 6 1 3 2 2 2 2 3	7 10 5

Задача N. Турнир

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

Ура! Король устраивает рыцарский турнир. Он уже разослал послание всем рыцарям королевства, а они, в свою очередь, дали согласие на участие в этом грандиозном событии.

Что же касается вас, то вы — простой крестьянин. Не удивительно, что рыцарский турнир вы проспали (ведь он приходился в выходной), и теперь вам очень хочется узнать его результаты. В этот раз рыцарский турнир проходил следующим образом:

- В турнире принимали участие n рыцарей. Каждому рыцарю был присвоен уникальный номер — целое число от 1 до n .
- Турнир проводился в m сражений, в i -ом сражении все еще не выбывшие рыцари с номерами не меньше l_i и не больше r_i боролись за право продолжить участие в турнире.
- После i -го сражения среди всех рыцарей, которые боролись, победил только один — рыцарь с номером x_i , он продолжил участие в турнире. Остальные рыцари выбыли из турнира.

Вы узнали у своих друзей информацию про все сражения, и теперь для каждого рыцаря вам интересно знать, каким рыцарем он был побежден. Считается, что рыцарь с номером a победил рыцаря с номером b , если было такое сражение, в котором боролись оба этих рыцаря, а победителем из этого сражения вышел рыцарь с номером a .

Напишите программу, вычисляющую для каждого рыцаря, каким рыцарем он был побежден.

Формат входных данных

В первой строке входных данных записано два целых числа n, m ($2 \leq n \leq 3 \cdot 10^5$; $1 \leq m \leq 3 \cdot 10^5$) — количество рыцарей и количество сражений, соответственно. В каждой из следующих m строк записано три целых числа l_i, r_i, x_i ($1 \leq l_i < r_i \leq n$; $l_i \leq x_i \leq r_i$) — описание i -го сражения.

Гарантируется, что в каждом сражении участвовали как минимум два рыцаря.

Формат выходных данных

Выведите n целых чисел. Если i -ый рыцарь был повержен, то i -ое число должно быть равно номеру рыцаря, который победил рыцаря с номером i . Если i -ый рыцарь не был никем побежден, то i -ое число должно быть равно 0.

Примеры

стандартный ввод	стандартный вывод
4 3 1 2 1 1 3 3 1 4 4	3 1 4 0
8 4 3 5 4 3 7 6 2 8 8 1 8 1	0 8 4 6 4 8 6 1