

Задача А. Перестановки

Имя входного файла: `permutation.in`
Имя выходного файла: `permutation.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по N , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y , по величине лежат в интервале от k до l . Сделайте то же самое.

Формат входных данных

В первой строке лежит два натуральных числа — $1 \leq N \leq 100\,000$ — количество чисел, которые выписал Вася и $1 \leq M \leq 100\,000$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leq x \leq y \leq N$ и $1 \leq k \leq l \leq N$.

Формат выходных данных

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Примеры

<code>permutation.in</code>	<code>permutation.out</code>
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

Задача В. Калевич наносит ответный удар

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

И снова жители Берляндии могут убедиться, что талантливый человек талантлив во всём. А талант — это всегда поиск новых способов самовыражения. На этот раз гений Калевича поразил всех своей новой картиной «Прямоугольные рамки». Шедевр полон непостижимых смыслов и скрытых тем.

Узкие черные прямоугольные рамки нарисованы на белом прямоугольном холсте. Никакие две рамки не имеют общих точек. Стороны каждой рамки параллельны сторонам холста.

Картина очень большая, и уменьшенная копия не может передать идею оригинальной картины. Поэтому Калевич захотел, чтобы вместо копий использовали упрощённые версии шедевра. Упрощённая версия представляет из себя последовательность площадей всех граней оригинала. *Грань* — связанная замкнутая белая область картины.

Формат входных данных

Первая строка содержит целое число N ($1 \leq N \leq 60\,000$) — число рамок на картине. Вторая строка содержит целые числа W и H ($1 \leq W, H \leq 10^8$).

Введём декартову систему координат таким образом, чтобы левый нижний угол холста имел координаты $(0, 0)$, и правый верхний угол имел координаты (W, H) . Стороны холста будут параллельны осям координат.

Следующие N строк содержат описания граней. Каждое описание состоит из координат двух противоположных углов соответствующей грани x_1, y_1, x_2, y_2 ($1 \leq x_1, x_2 < W$; $1 \leq y_1, y_2 < H$; $x_1 \neq x_2, y_1 \neq y_2$). Все координаты целые. Никакие две грани не имеют общих точек.

Формат выходных данных

Выведите желаемую последовательность в неубывающем порядке.

Примеры

<code>kalivech.in</code>	<code>kalivech.out</code>
1	1 8
3 3	
2 1 1 2	

Задача С. RMQ наоборот

Имя входного файла: `rmq.in`
Имя выходного файла: `rmq.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Рассмотрим массив $a[1..n]$. Пусть $Q(i, j)$ — ответ на запрос о нахождении минимума среди чисел $a[i], \dots, a[j]$. Вам даны несколько запросов и ответы на них. Восстановите исходный массив.

Формат входных данных

Первая строка входного файла содержит число n — размер массива, и m — число запросов ($1 \leq n, m \leq 100\,000$). Следующие m строк содержат по три целых числа i, j и q , означающих, что $Q(i, j) = q$ ($1 \leq i \leq j \leq n, -2^{31} \leq q \leq 2^{31} - 1$).

Формат выходных данных

Если искомого массива не существует, выведите строку «`inconsistent`».

В противном случае в первую строку выходного файла выведите «`consistent`». Во вторую строку выходного файла выведите элементы массива. Элементами массива должны быть целые числа в интервале от -2^{31} до $2^{31} - 1$ включительно. Если решений несколько, выведите любое.

Примеры

<code>rmq.in</code>	<code>rmq.out</code>
3 2 1 2 1 2 3 2	<code>consistent</code> 1 2 2
3 3 1 2 1 1 1 2 2 3 2	<code>inconsistent</code>

Задача D. Ладьи и прямоугольники

Имя входного файла: `rooks.in`
Имя выходного файла: `rooks.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

У Поликарпа есть шахматная доска размера $n \times m$, на которой расставлены k ладей. Поликарп еще не придумал правила игры, в которую он будет играть. Однако он уже выделил на доске q прямоугольных участков особой стратегической важности, которые должны быть надежно защищены. По мнению Поликарпа, прямоугольный участок доски надежно защищен, если все его свободные клетки бьются ладьями, стоящими на этом участке. Ладьи на остальной части доски на защиту участка не влияют. Расстановка ладей фиксирована и не может быть изменена. Напомним, что ладья бьет все клетки, расположенные с ней на одной вертикали или горизонтали, если между клеткой и ладьей нет других фигур. Помогите Поликарпу определить, все ли стратегически важные участки надежно защищены.

Формат входных данных

В первой строке содержатся четыре целых числа n , m , k и q ($1 \leq n, m \leq 100\,000$, $1 \leq k, q \leq 200\,000$) — размеры доски, количество ладей и количество стратегически важных участков. Будем считать, что клетки доски пронумерованы числами от 1 до n по горизонтали и от 1 до m по вертикали. Следующие k строк содержат пары целых чисел « $x\ y$ », описывающие положение ладей ($1 \leq x \leq n$, $1 \leq y \leq m$). Гарантируется, что все ладьи стоят в разных клетках. Следующие q строк описывают стратегически важные участки четверками чисел « $x_1\ y_1\ x_2\ y_2$ » ($1 \leq x_1 \leq x_2 \leq n$, $1 \leq y_1 \leq y_2 \leq m$). Соответствующий прямоугольный участок состоит из клеток (x, y) , для которых $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$. Стратегически важные участки могут пересекаться или совпадать.

Формат выходных данных

Выведите q строк. Для каждого стратегически важного участка выведите «YES», если он надежно защищен, и «NO» в противном случае.

Примеры

<code>rooks.in</code>	<code>rooks.out</code>
4 3 3 3 1 1 3 2 2 3 2 3 2 3 2 1 3 3 1 2 2 3	YES YES NO
1 1 1 1 1 1 1 1 1 1	YES