Задача A. Range Variation Query

Имя входного файла:	`rvq.in`
Имя выходного файла:	`rvq.out`
Ограничение по времени:	0.5 секунда
Ограничение по памяти:	64 мегабайта

В начальный момент времени последовательность a_n задана следующей формулой: $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значениями среди элементов $a_i, a_{i+1}, \ldots, a_j$;
- присвоить элементу a_i значение j.

Формат входных данных

Первая строка входного файла содержит натуральное число k — количество запросов ($1 \leqslant k \leqslant 100\,000$). Следующие k строк содержат запросы, по одному на строке. Запрос номер i описывается двумя целыми числами x_i, y_i.

Если $x_i > 0$, то требуется найти разность между максимальным и минимальным значениями среди элементов $a_{x_i}, \ldots, a_{y_i}$. При этом $1 \leqslant x_i \leqslant y_i \leqslant 100\,000$.

Если $x_i < 0$, то требуется присвоить элементу $a_{|x_i|}$ значение y_i. В этом случае $-100\,000 \leqslant x_i \leqslant -1$ и $|y_i| \leqslant 100\,000$.

Формат выходных данных

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значениями на соответствующем отрезке.

Примеры

rvq.in	rvq.out
7 1 3 2 4 -2 -100 1 5 8 9 -3 -101 2 3	34 68 250 234 1

Задача B. Дерево отрезков с операцией на отрезке

Имя входного файла:	`segment-tree.in`
Имя выходного файла:	`segment-tree.out`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	64 мегабайта

Реализуйте эффективную структуру данных для хранения элементов и увеличения нескольких подряд идущих элементов на одно и то же число.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leqslant N \leqslant 100\,000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leqslant M \leqslant 30\,000$) — количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (g — получить текущее значение элемента по его номеру, a — увеличить все элементы на отрезке).

Следом за g вводится одно число — номер элемента.

Следом за a вводится три числа — левый и правый концы отрезка и число add, на которое нужно увеличить все элементы данного отрезка массива ($1 \leqslant add \leqslant 100\,000$).

Формат выходных данных

Выведите в одну строку через пробел ответы на каждый запрос g.

Примеры

segment-tree.in	segment-tree.out
5 2 4 3 5 2 5 g 2 g 5 a 1 3 10 g 2 g 4	4 2 14 5

Задача C. Ближайшее большее число справа

Имя входного файла:	`nearandmore.in`
Имя выходного файла:	`nearandmore.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан массив a из n чисел. Нужно обрабатывать запросы:

0. `set(i, x)` – присвоить новое значение элементу массива `a[i] = x;`
1. `get(i, x)` – найти $\min k$: $k \geqslant i$ и $a_k \geqslant x$.

Формат входных данных

Первая строка входных данных содержит два числа: длину массива n и количество запросов m ($1 \leqslant n, m \leqslant 200\,000$).

Во второй строке записаны n целых чисел – элементы массива a ($0 \leqslant a_i \leqslant 200\,000$).

Следующие m строк содержат запросы, каждый запрос содержит три числа t, i, x. Первое число t равно 0 или 1 – тип запроса. $t = 0$ означает запрос типа `set`, $t = 1$ соответствует запросу типа `get`, $1 \leqslant i \leqslant n$, $0 \leqslant x \leqslant 200\,000$. Элементы массива нумеруются с единицы.

Формат выходных данных

На каждой запрос типа `get` на отдельной строке выведите соответствующее значение k. Если такого k не существует, выведите -1.

Примеры

nearandmore.in	nearandmore.out
4 5 1 2 3 4 1 1 1 1 1 3 1 1 5 0 2 3 1 1 3	1 3 -1 2

Задача D. Мега-инверсии

Имя входного файла:	`mega.in`
Имя выходного файла:	`mega.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Инверсией в перестановке $p_1, p_2, \ldots, p_N$ называется пара (i, j) такая, что $i < j$ и $p_i > p_j$. Назовём мега-инверсией в перестановке $p_1, p_2, \ldots, p_N$ тройку (i, j, k) такую, что $i < j < k$ и $p_i > p_j > p_k$. Напишите алгоритм для быстрого подсчёта количества мега-инверсий в перестановке.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leqslant N \leqslant 100\,000$). Следующие N чисел описывают перестановку: $p_1, p_2, \ldots, p_N$ ($1 \leqslant p_i \leqslant N$), все p_i попарно различны. Числа разделяются переводами строк.

Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное количеству мега-инверсий в перестановке $p_1, p_2, \ldots, p_N$.

Примеры

mega.in	mega.out
4 4 3 2 1	4

Задача A. 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 — $1 \leqslant N \leqslant 100\,000$ — количество чисел, которые выписал Вася и $1 \leqslant M \leqslant 100\,000$ — 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 $1 \leqslant x \leqslant y \leqslant N$ и $1 \leqslant k \leqslant l \leqslant N$.

Формат выходных данных

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

Примеры

permutation.in	permutation.out
4 2 1 2 3 4 1 2 2 3 1 3 1 3	1 3

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 ($1 \leqslant n, m \leqslant 100\,000$, $1 \leqslant k, q \leqslant 200\,000$) — 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 ($1 \leqslant x \leqslant n$, $1 \leqslant y \leqslant m$). 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» ($1 \leqslant x_1 \leqslant x_2 \leqslant n$, $1 \leqslant y_1 \leqslant y_2 \leqslant m$). Соответствующий прямоугольный участок состоит из клеток (x, y), для которых $x_1 \leqslant x \leqslant x_2$, $y_1 \leqslant y \leqslant y_2$. 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

rooks.in	rooks.out
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

Задача B. Максимизируй то

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

Всем хорошо известна следующая задача:

> Дан массив из целых чисел a_1, a_2, ..., a_n. Найдите такой непустой подотрезок a_l, a_{l+1}, ..., a_r этого массива ($1 \leqslant l \leqslant r \leqslant n$), что сумма чисел $a_l + a_{l+1} + \ldots + a_r$ является максимально возможной.

Дан массив из целых чисел. Вера очень хотела бы для нескольких подотрезков этого массива решить предыдущую задачу, но не смогла. Помогите ей!

Формат входных данных

Входные данные содержат один или несколько тестовых примеров. Описание каждого из них начинается с двух чисел n и m — длины массива и числа интересующих Веру подотрезков.

В следующей строке содержится n чисел — элементы массива. Каждое из этих чисел по абсолютной величине не превосходит 10^4.

Далее следуют описания подотрезков, каждое описание состоит из двух чисел l и r, обозначающих левый и правый конец подотрезка ($1 \leqslant l \leqslant r \leqslant n$).

Суммарная длина всех массивов, а также суммарное число подотрезков не превосходит 10^5.

Формат выходных данных

Для каждого из тестовых примеров выведите m чисел: искомую максимальную сумму для каждого из подотрезков.

Примеры

maxsum.in	maxsum.out
10 3 -100 1 2 3 4 -10 50 -100 -1 2 1 10 1 5 9 9 5 2 -1 2 -1 2 -1 1 5 2 4	50 10 -1 3 3

Объединение прямоугольников

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

На плоскости задано N прямоугольников с вершинами в точках с целыми координатами и сторонами, параллельными осям координат. Необходимо найти площадь их объединения.

Формат входных данных

В первой строке входного файла указано число N ($1 \leqslant N \leqslant 100000$). В следующих N строках заданы по 4 целых числа x_1, y_1, x_2, y_2 — сначала координаты левого нижнего угла прямоугольника, потом правого верхнего ($-10^9 \leqslant x_1 \leqslant x_2 \leqslant 10^9$, $-10^9 \leqslant y_1 \leqslant y_2 \leqslant 10^9$). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

Формат выходных данных

В выходной файл выведите единственное число — ответ на задачу.

Примеры

union-fast.in	union-fast.out
3 1 1 3 5 5 2 7 4 2 4 6 7	23