Задача A. Катый ноль

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

Реализуйте эффективную структуру данных, позволяющую изменять элементы массива и вычислять индекс k-го слева нуля на данном отрезке в массиве.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leqslant N \leqslant 200\,000$) — 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 ($1 \leqslant M \leqslant 200\,000$) — количество запросов. Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (`s` — вычислить индекс k-го нуля, `u` — обновить значение элемента). Следом за `s` вводится три числа — левый и правый концы отрезка и число k ($1 \leqslant k \leqslant N$). Следом за `u` вводятся два числа — номер элемента и его новое значение.

Формат выходных данных

Для каждого запроса s выведите результат. Все числа выводите в одну строку через пробел. Если нужного числа нулей на запрашиваемом отрезке нет, выводите -1 для данного запроса.

Примеры

kthzero.in	kthzero.out
5 0 0 3 0 2 3 u 1 5 u 1 0 s 1 5 3	4

Замечание

TL для Python 10 секунд

Задача B. Разреженные таблицы

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

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

Элементы $a_2, a_3, \ldots, a_n$ 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:

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

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

sparse.in	sparse.out
10 8 12345 3 9	5 3 1565158

Замечание

Пояснение к тесту из примера: запросы и результаты.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
12345	305498	7048017	11694653	1565158	2591019	9471233	570265	13137658	1325095

#	u	v	ans
1	3	9	570265
2	10	1	12345
3	1	2	12345
4	10	10	1325095
5	5	9	570265
6	2	1	12345
7	3	2	305498
8	5	3	1565158

Задача C. Перестановки

Имя входного файла:	`permutation.in`
Имя выходного файла:	`permutation.out`
Ограничение по времени:	1.5 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. Звезды

Имя входного файла:	`stars.in`
Имя выходного файла:	`stars.out`
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 $1 \leqslant n \leqslant 128$. 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 ($0 \leqslant x, y, z < N$) и k ($-20000 \leqslant k \leqslant 20000$) — 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 ($0 \leqslant x_1 \leqslant x_2 < N$, $0 \leqslant y_1 \leqslant y_2 < N$, $0 \leqslant z_1 \leqslant z_2 < N$), которые означают, что Петя попросил подсчитать количество звезд в кубиках (x, y, z) из области: $x_1 \leqslant x \leqslant x_2$, $y_1 \leqslant y \leqslant y_2$, $z_1 \leqslant z \leqslant z_2$;

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.

Примеры

stars.in	stars.out
2 2 1 1 1 1 1 1 1 0 0 0 1 1 0 1 0 3 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	0 1 4 2

Задача E. Объединение прямоугольников

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

От вас требуется решить задачу объединения прямоугольников

Формат входных данных

В первой строке входных данных дано число $n \leqslant 50000$ – количество прямоугольников. В следующих n строках задаются прямоугольники в формате $x_1, y_1, x_2, y_2 (0 \leqslant x_1, y_1, x_2, y_2 \leqslant 50000)$ – координаты левого нижнего и правого верхнего углов прямоугольника

Формат выходных данных

В единственной строке выходной данных выведите одно число – площадь объединения всех данных прямоугольников.

Примеры

стандартный ввод	стандартный вывод
2 0 0 2 2 1 3 2 4	5
3 0 0 2 4 4 1 6 3 1 3 5 6	23

Задача F. 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 $a[i], \ldots, a[j]$. 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 ($1 \leqslant n, m \leqslant 100\,000$). Следующие m строк содержат по три целых числа i, j и q, означающих, что $Q(i, j) = q$ ($1 \leqslant i \leqslant j \leqslant n$, $-2^{31} \leqslant q \leqslant 2^{31} - 1$).

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

rmq.in	rmq.out
3 2 1 2 1 2 3 2	consistent 1 2 2
3 3 1 2 1 1 1 2 2 3 2	inconsistent

Задача G. Подпалиндромы

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

Дано слово и запросы двух типов:

- заменить i-ю букву в слове на букву c;
- проверить, является ли подстрока $s_j \dots s_k$ палиндромом.

Формат входных данных

В первой строке записано слово из n строчных латинских букв. Во второй строке записано целое число m — количество запросов ($5 \leqslant n, m \leqslant 10^5$). Следующие m строк содержат запросы. Каждый запрос имеет вид «`change` i a» или «`palindrome?` j k», где i, j, k — целые числа ($1 \leqslant i \leqslant n; 1 \leqslant j \leqslant k \leqslant n$), а символ c — строчная латинская буква.

Формат выходных данных

На все запросы второго типа выведите «`Yes`», если подслово $s_j \dots s_k$ является палиндромом, и «`No`» в противном случае.

Примеры

substring-palindromes.in	substring-palindromes.out
abcda 5 palindrome? 1 5 palindrome? 1 1 change 4 b palindrome? 1 5 palindrome? 2 4	No Yes Yes Yes

Задача H. Друзья и последовательности

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

Майк и !Майк соперничают еще со школьных лет, они противоположны во всем что делают, кроме программирования. Сегодня у них возникла проблема, которую сами друзья сами решить не могут, но вместе с вами — кто знает?

Каждый из них знает две последовательности n чисел a и b. По запросу в виде пары целых чисел (l, r) Майк может сразу сообщить значение $\max\limits_{i=l}^{r} a_i$, а !Майк — значение $\min\limits_{i=l}^{r} b_i$.

Предположим, что робот задает им каждый из возможных различных запросов в виде пары целых чисел (l, r) $(1 \leqslant l \leqslant r \leqslant n)$ (то есть он сделает ровно $n(n+1)/2$ запросов) и считает, сколько раз их ответы на один и тот же запрос совпадают, то есть для скольких пар выполняется $\max\limits_{i=l}^{r} a_i = \min\limits_{i=l}^{r} b_i$.

Сколько случаев совпадения посчитает робот?

Формат входных данных

В первой строке содержится единственное целое число n $(1 \leqslant n \leqslant 200\,000)$.

Во второй строке содержатся n целых чисел $a_1, a_2, \ldots, a_n$ $(-10^9 \leqslant a_i \leqslant 10^9)$ — элементы последовательности a.

В третьей строке содержатся n целых чисел $b_1, b_2, \ldots, b_n$ $(-10^9 \leqslant b_i \leqslant 10^9)$ — элементы последовательности b.

Формат выходных данных

Выведите одно целое число — количество совпадений ответов, которые посчитает робот, то есть для скольких пар выполняется $\max\limits_{i=l}^{r} a_i = \min\limits_{i=l}^{r} b_i$.

Примеры

friends.in	friends.out
6 1 2 3 2 1 4 6 7 1 2 3 2	2
3 3 3 3 1 1 1	0

Задача I. K-ый максимум

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

Напишите программу, реализующую структуру данных, позволяющую добавлять и удалять элементы, а также находить k-й максимум.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество команд ($n \leqslant 100\,000$). Последующие n строк содержат по одной команде каждая. Команда записывается в виде двух чисел c_i и k_i — тип и аргумент команды соответственно ($|k_i| \leqslant 10^9$). Поддерживаемые команды:

- $+1$ (или просто 1): Добавить элемент с ключом k_i.
- 0: Найти и вывести k_i-й максимум.
- -1: Удалить элемент с ключом k_i.

Гарантируется, что в процессе работы в структуре не требуется хранить элементы с равными ключами или удалять несуществующие элементы. Также гарантируется, что при запросе k_i-го максимума, он существует.

Формат выходных данных

Для каждой команды нулевого типа в выходной файл должна быть выведена строка, содержащая единственное число — k_i-й максимум.

Примеры

kthmax.in	kthmax.out
11 +1 5 +1 3 +1 7 0 1 0 2 0 3 -1 5 +1 10 0 1 0 2 0 3	7 5 3 10 7 3

Задача J. Размен денег

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

За свою долгую жизнь Боря собрал коллекцию из n монет. Он выложил все эти монеты в ряд. При этом i-я в ряду монета имеет номинал a_i.

Боря собирается в очередное путешествие, но у него осталось очень мало времени на сборы. Поэтому он хочет взять некоторый отрезок лежащих подряд монет и надеется, что ему их хватит.

Боря хочет ответить на несколько запросов. В каждом запросе Боря хочет узнать, какую минимальную сумму он не сможет заплатить без сдачи, если он возьмет все монеты с l_i-й по r_i-ю. Более формально, он хочет найти такое минимальное натуральное число z, что нельзя выбрать подмножество монет с номерами от l_i до r_i, суммарный номинал которых равен z.

Формат входных данных

В первой строке задано два целых числа n и m ($1 \leqslant n, m \leqslant 150\,000$) — количество монет у Бори и количество запросов. В следующей строке задано n чисел a_i ($1 \leqslant a_i \leqslant 10^9$) — номинал i-й монеты.

В следующих m строках задано по два числа l_i и r_i ($1 \leqslant l_i \leqslant r_i \leqslant n$) — описание запросов.

Формат выходных данных

На каждый из m запросов выведите минимальную сумму, которую нельзя заплатить без сдачи, воспользовавшись монетами с l_i-й по r_i-ю.

Примеры

стандартный ввод	стандартный вывод
5 5 2 1 5 3 1 1 5 1 3 1 1 2 4 2 5	13 4 1 2 11