

Задача А. Катый ноль

Имя входного файла: `kthzero.in`
Имя выходного файла: `kthzero.out`
Ограничение по времени: 2 секунды
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 ($1 \leq N \leq 200\,000$) — 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 ($1 \leq M \leq 200\,000$) — 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 ($1 \leq k \leq N$). 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

<code>kthzero.in</code>	<code>kthzero.out</code>
5 0 0 3 0 2 3 u 1 5 u 1 0 s 1 5 3	4

Замечание

TL для Python 8 секунд

Задача В. 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 $\min k: k \geq i$ и $a_k \geq x$.

Формат входных данных

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

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

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`, $1 \leq i \leq n$, $0 \leq x \leq 200\,000$. Элементы массива нумеруются с единицы.

Формат выходных данных

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

Примеры

<code>nearandmore.in</code>	<code>nearandmore.out</code>
4 5	1
1 2 3 4	3
1 1 1	-1
1 1 3	2
1 1 5	
0 2 3	
1 1 3	

Замечание

TL для Python 10 секунд

Задача С. Прибавление и максимум

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

Реализуйте эффективную структуру данных для хранения массива и выполнения следующих операций: увеличение всех элементов данного интервала на одно и то же число; поиск максимума на интервале.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100000$) – количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100000 – элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leq M \leq 30000$) – количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (m – найти максимум, a – увеличить все элементы на отрезке).

Следом за m вводятся два числа – левая и правая граница интервала.

Следом за a вводятся три числа – левый и правый концы отрезка и число add , на которое нужно увеличить все элементы данного отрезка массива ($0 \leq add \leq 100000$).

Формат выходных данных

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

Примеры

addandmax.in	addandmax.out
5	4
2 4 3 1 5	104
5	104
m 1 3	
a 2 4 100	
m 1 3	
a 5 5 10	
m 1 5	

Задача D. Присваивание, прибавление и сумма

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Есть массив из n элементов, изначально заполненный нулями. Вам нужно написать структуру данных, которая обрабатывает три вида запросов:

- присвоить всем элементам на отрезке от l до $r - 1$ значение v ,
- прибавить ко всем элементам на отрезке от l до $r - 1$ число v ,
- узнать сумму на отрезке от l до $r - 1$.

Формат входных данных

Первая строка содержит два числа n и m ($1 \leq n, m \leq 100000$) — размер массива и число операций. Далее следует описание операций. Описание каждой операции имеет следующий вид:

- $1 \ l \ r \ v$ — присвоить всем элементам на отрезке от l до $r - 1$ значение v ($0 \leq l < r \leq n$, $0 \leq v \leq 10^5$).
- $2 \ l \ r \ v$ — прибавить ко всем элементам на отрезке от l до $r - 1$ число v ($0 \leq l < r \leq n$, $0 \leq v \leq 10^5$).
- $3 \ l \ r$ — узнать сумму на отрезке от l до $r - 1$ ($0 \leq l < r \leq n$).

Формат выходных данных

Для каждой операции третьего типа выведите соответствующее значение.

Примеры

стандартный ввод	стандартный вывод
5 7	8
1 0 3 3	10
2 2 4 2	4
3 1 3	
2 1 5 1	
1 0 2 2	
3 0 3	
3 3 5	

Задача Е. Блин, че-то я тупанул, конечно

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

Письмо на Балабановскую спичечную фабрику: «Я 11 лет считаю спички у вас в коробках — их то 59, то 60, а иногда и 58. Вы там сумасшедшие что ли все???»

Гриша очень педантичный препод, который любит спорт. Поэтому он записал количество голов, которые он, *по его словам*, забил за каждый футбольный матч (может даже отрицательное, если он пропускал голы).

Еще Гриша считает, что главное в спорте — прогресс. Но прогресс должен быть плавным. Поэтому он хочет для разных последовательностей дней узнавать, насколько же прогресс был плавным. Мера плавности — длина максимального подотрезка вида $(x, x + 1, \dots, x + l)$, где l — длина такого отрезка.

Сокомандник Гриши, Женя, завидует результативности Гриши. Поэтому иногда он убавляет количество голов в блокнотике Гриши за некоторые матчи (иногда даже добавляет, чтобы ему не так обидно было).

А еще Гриша старший преподаватель, поэтому у него нет времени че-то там считать, поэтому вам дали данную задачу.

Формат входных данных

В первой строке находится одно целое число n — количество матчей по футболу за смену, в которых Гриша принял участие. $1 \leq n \leq 10^5$ Во второй строке находятся n чисел q_i ($|q_i| \leq 10^9$) — количество голов, который Гриша забил в каждом матче. В третьей строке находится одно целое число m ($0 \leq m \leq 100\,000$) — количество действий с блокнотиком Гриши. В следующих m строках содержится описание эксперимента.

- $+ l_i r_i d_i$ — Женя добавил d_i голов в матчах с l_i -го по r_i -й. ($1 \leq l_i \leq r_i \leq n$, $|d_i| \leq 10^9$)
- $? l_i r_i$ — Гриша попросил Вас узнать насколько у него был плавный прогресс в матчах с l_i -го по r_i -й. ($1 \leq l_i \leq r_i \leq n$)

Формат выходных данных

Для каждого действия второго типа выведите в новой строке одно число — меру плавности.

Примеры

<code>atoms.in</code>	<code>atoms.out</code>
6	3
2 3 4 3 4 4	3
5	5
? 1 6	
+ 6 6 1	
? 2 6	
+ 4 6 2	
? 1 5	

Задача F. 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 ($5 \leq n, m \leq 10^5$). Следующие m строк содержат запросы. Каждый запрос имеет вид «`change  $i$   $a$` » или «`palindrome?  $j$   $k$` », где i, j, k — целые числа ($1 \leq i \leq n; 1 \leq j \leq k \leq n$), 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

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

Задача G. 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 — $1 \leq N \leq 100\,000$ — 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	