

Задача А. Различные числа

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

Сколько различных чисел на отрезке массива?

Формат входных данных

На первой строке длина массива n ($1 \leq n \leq 300\,000$). На второй строке n целых чисел от 0 до 10^9-1 . На третьей строке количество запросов q ($1 \leq q \leq 300\,000$). Следующие q строк содержат описание запросов, по одному на строке. Каждый запрос задаётся парой целых чисел l, r ($1 \leq l \leq r \leq n$).

Формат выходных данных

Выведите ответы на запросы по одному в строке.

Примеры

<code>distinctnum.in</code>	<code>distinctnum.out</code>
5	3
1 1 2 1 3	2
3	3
1 5	
2 4	
3 5	

Задача В. 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, \dots, p_N называется пара (i, j) такая, что $i < j$ и $p_i > p_j$. Назовём мега-инверсией в перестановке p_1, p_2, \dots, 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 ($1 \leq N \leq 100\,000$). Следующие N чисел описывают перестановку: p_1, p_2, \dots, p_N ($1 \leq p_i \leq N$), 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, \dots, p_N .

Примеры

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

Задача С. Вложенные отрезки

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

Дан массив из $2n$ чисел, каждое число от 1 до n в нем встречается ровно два раза. Будем говорить, что отрезок y вложен в отрезок x , если оба вхождения числа y находятся между вхождениями числа x . Найдите для каждого отрезка i , сколько есть отрезков, которые в него вложены.

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 10^5$), вторая строка содержит $2n$ чисел. Гарантируется, что каждое число от 1 до n встречается ровно два раза.

Формат выходных данных

Выведите n чисел, i -е число равно числу отрезков, вложенных в отрезок i .

Пример

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

Задача D. Битовые OR и AND

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

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

- применить ко всем элементам от l до $r - 1$ операцию $a_i = a_i | v$ (битовый OR),
- найти битовый AND элементов 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:

- $1 \ l \ r \ v$ — применить ко всем элементам от l до $r - 1$ операцию $a_i = a_i | v$ ($0 \leq l < r \leq n$, $0 \leq v < 2^{30}$).
- $2 \ l \ r$ — найти битовый AND элементов на отрезке от l до $r - 1$ ($0 \leq l < r \leq n$).

Формат выходных данных

Для каждой операции второго типа выведите соответствующее значение.

Примеры

стандартный ввод	стандартный вывод
5 6 1 0 3 3 2 1 2 1 1 4 4 2 1 3 2 1 4 2 3 5	3 7 4 0
2 3 1 0 1 3 1 1 2 9 2 0 2	1

Задача Е. Атомы

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

В лаборатории аномальных материалов антинауочно-исследовательского комплекса «Black Mesa» проводят эксперименты с недавно разработанным графитовым наностержнем. Графитовый наностержень представляет собой n последовательно соединенных атомов углерода, находящихся на одной прямой. Каждый атом имеет определенный заряд.

Для проведения эксперимента, стержень располагают вертикально. Пронумеруем атомы от 1 до n снизу вверх. Между двумя атомами образуется сильная связь, если это соседние атомы и верхний из них имеет заряд ровно на один больше, чем нижний. Иными словами, атомы a и b соединены сильной связью, если $a = b + 1$ и $q_a = q_b + 1$, где q_i — заряд i -го атома. Цепочкой атомов назовем несколько последовательных атомов, соединенных сильными связями.

Вчера был проведен очередной эксперимент. Перед началом эксперимента каждому атому установили определенный заряд: i -му атому установили заряд q_i .

Во время эксперимента ученые проводили действия двух типов:

- у всех атомов с номерами от l_i до r_i , включительно, заряд изменяли на величину d_i ;
- временно разрушали все сильные связи атомов, кроме тех, которые соединяют атомы с номерами от l_i до r_i , включительно, и измеряли длину самой длинной цепочки атомов среди оставшихся сильных связей. Затем восстанавливали все временно разрушенные связи.

Было произведено m действий, однако выяснилось, что в результате побочного эффекта эксперимента запись результатов измерений оказалась утеряна. Для продолжения работы с графитовым наностержнем необходимо восстановить результаты вчерашних измерений. К счастью, сохранился план действий, произведенных во время эксперимента. Помогите ученым продолжить исследования, восстановите результаты измерений.

Формат входных данных

В первой строке находится одно целое число n ($1 \leq n \leq 100\,000$) — количество атомов в наностержне. Во второй строке находятся n чисел q_i ($|q_i| \leq 10^9$) — начальный заряд i -го атома. В третьей строке находится одно целое число m ($0 \leq m \leq 100\,000$) — количество действий в эксперименте. В следующих m строках содержится описание эксперимента.

Если строка начинается с символа «+», очередное действие — изменение заряда атомов. В таком случае, далее в этой строке находятся три целых числа: l_i , r_i и d_i ($1 \leq l_i \leq r_i \leq n$, $|d_i| \leq 10^9$), которые характеризуют это действие.

Если строка начинается с символа «?», очередное действие — второго типа. В таком случае, далее в этой строке находятся два целых числа: l_i и r_i ($1 \leq l_i \leq r_i \leq n$), которые характеризуют это действие.

Формат выходных данных

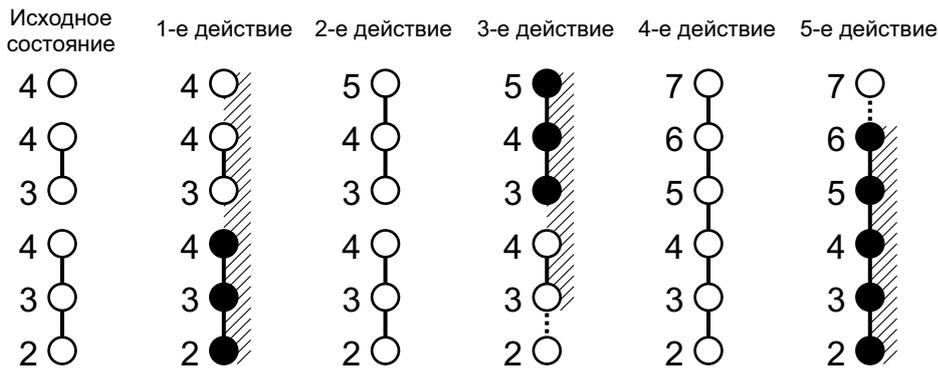
Для каждого действия второго типа выведите в новой строке одно число — длину наибольшей цепочки.

Примеры

atoms.in	atoms.out
6	3
2 3 4 3 4 4	3
5	5
? 1 6	
+ 6 6 1	
? 2 6	
+ 4 6 2	
? 1 5	

Замечание

Иллюстрация к примеру. Пунктиром выделены сильные связи, которые разрушаются на время действия второго типа. Для каждого действия второго типа выделены отрезок запроса и самая длинная цепочка.



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

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

Задача Н. K -я порядковая статистика на отрезке

Имя входного файла: `kth.in`
 Имя выходного файла: `kth.out`
 Ограничение по времени: 6 секунд
 Ограничение по памяти: 512 мегабайт

Дан массив из N неотрицательных чисел, строго меньших 10^9 . Вам необходимо ответить на несколько запросов о величине k -й порядковой статистики на отрезке $[l, r]$.

Формат входных данных

Первая строка содержит число N ($1 \leq N \leq 450\,000$) — размер массива.

Вторая строка может быть использована для генерации a_i — начальных значений элементов массива. Она содержит три числа a_1, l и m ($0 \leq a_1, l, m < 10^9$); для i от 2 до N

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9.$$

В частности, $0 \leq a_i < 10^9$.

Третья строка содержит одно целое число B ($1 \leq B \leq 1000$) — количество групп запросов.

Следующие B строк описывают одну группу запросов. Каждая группа запросов описывается 10 числами. Первое число G обозначает количество запросов в группе. Далее следуют числа x_1, l_x и m_x , затем y_1, l_y и m_y , затем, k_1, l_k и m_k ($1 \leq x_1 \leq y_1 \leq N, 1 \leq k_1 \leq y_1 - x_1 + 1, 0 \leq l_x, m_x, l_y, m_y, l_k, m_k < 10^9$). Эти числа используются для генерации вспомогательных последовательностей x_g и y_g , а также параметров запросов i_g, j_g и k_g ($1 \leq g \leq G$)

$$\begin{aligned} x_g &= ((i_{g-1} - 1) \cdot l_x + m_x) \bmod N + 1, & 2 \leq g \leq G \\ y_g &= ((j_{g-1} - 1) \cdot l_y + m_y) \bmod N + 1, & 2 \leq g \leq G \\ i_g &= \min(x_g, y_g), & 1 \leq g \leq G \\ j_g &= \max(x_g, y_g), & 1 \leq g \leq G \\ k_g &= (((k_{g-1} - 1) \cdot l_k + m_k) \bmod (j_g - i_g + 1)) + 1, & 2 \leq g \leq G \end{aligned}$$

Сгенерированные последовательности описывают запросы, g -й запрос состоит в поиске k_g -го по величине числа среди элементов отрезка $[i_g, j_g]$.

Суммарное количество запросов не превосходит 600 000.

Формат выходных данных

Выведите единственное число — сумму ответов на запросы.

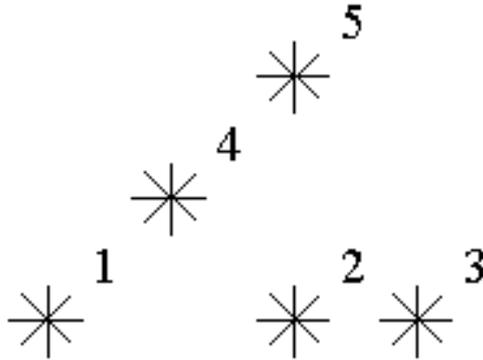
Примеры

kth.in	kth.out
5	15
1 1 1	
5	
1	
1 0 0 3 0 0 2 0 0	
1	
2 0 0 5 0 0 3 0 0	
1	
1 0 0 5 0 0 5 0 0	
1	
3 0 0 3 0 0 1 0 0	
1	
1 0 0 4 0 0 1 0 0	

Задача I. Звёзды

Имя входного файла: `stars.in`
Имя выходного файла: `stars.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Астрономы часто исследуют звёздные карты, на которых звёзды представлены точками на плоскости, каждая звезда имеет декартовы координаты. Пусть уровень звезды – количество звёзд, которые не выше и не правее данной звезды. Астрономы хотят найти распределение уровней звёзд.



Для примера посмотрим на карту звёзд на картинке выше. Уровень звезды номер 5 равен 3 (т.к. есть звёзды с номерами 1, 2, 4). Уровни звёзд 2 и 4 равны 1. На данной карте есть только одна звезда на уровне 0, две звезды на уровне 1, одна звезда на уровне 2 и одна звезда на уровне 3. Напишите программу, считающую количество звёзд на каждом уровне.

Формат входных данных

Вам дан один или несколько тестов. Каждый тест описывается следующим образом.

В первой строке количество звёзд N ($1 \leq N \leq 15\,000$). Следующие N строк описывают координаты звёзд (два целых числа X и Y , разделённые пробелом, $0 \leq X, Y \leq 32\,000$). В каждой точке плоскости находится не более одной звезды. Звёзды перечислены в порядке возрастания Y координаты, при равенстве в порядке возрастания X координаты.

Формат выходных данных

Выведите ответ для каждого теста. Ответ для теста описывается следующим образом. N строк, по одному числу в строке. i -я строка содержит количество звёзд на уровне i ($i = 0 \dots N-1$).

Примеры

stars.in	stars.out
5	1
1 1	2
5 1	1
7 1	1
3 3	0
5 5	1
5	2
1 1	1
5 1	1
7 1	0
3 3	
5 5	

Задача J. Откат

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

Сергей работает системным администратором в очень крупной компании. Естественно, в круг его обязанностей входит резервное копирование информации, хранящейся на различных серверах и «откат» к предыдущей версии в случае возникновения проблем.

В данный момент Сергей борется с проблемой недостатка места для хранения информации для восстановления. Он решил перенести часть информации на новые сервера. К сожалению, если что-то случится во время переноса, он не сможет произвести откат, поэтому процедура переноса должна быть тщательно спланирована.

На данный момент у Сергея хранятся n точек восстановления различных серверов, пронумерованных от 1 до n . Точка восстановления с номером i позволяет произвести откат для сервера a_i . Сергей решил разбить перенос на этапы, при этом на каждом этапе в случае возникновения проблем будут доступны точки восстановления с номерами $l, l + 1, \dots, r$ для некоторых l и r .

Для того, чтобы спланировать перенос данных оптимальным образом, Сергею необходимо научиться отвечать на запросы: для заданного l , при каком минимальном r в процессе переноса будут доступны точки восстановления не менее чем k различных серверов.

Помогите Сергею.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m , разделенные пробелами — количество точек восстановления и количество серверов ($1 \leq n, m \leq 100\,000$). Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n — номера серверов, которым соответствуют точки восстановления ($1 \leq a_i \leq m$).

Третья строка входного файла содержит q — количество запросов, которые необходимо обработать ($1 \leq q \leq 100\,000$). В процессе обработки запросов необходимо поддерживать число p , исходно оно равно 0. Каждый запрос задается парой чисел x_i и y_i , используйте их для получения данных запроса следующим образом: $l_i = ((x_i + p) \bmod n) + 1$, $k_i = ((y_i + p) \bmod m) + 1$ ($1 \leq l_i, x_i \leq n$, $1 \leq k_i, y_i \leq m$). Пусть ответ на i -й запрос равен r . После выполнения этого запроса, следует присвоить p значение r .

Формат выходных данных

На каждый запрос выведите одно число — искомое минимальное r , либо 0, если такого r не существует.

Примеры

<code>rollback.in</code>	<code>rollback.out</code>
7 3	1
1 2 1 3 1 2 1	4
4	0
7 3	6
7 1	
7 1	
2 2	