

Задача А. 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 на отрезке (оба конца включительно), a – увеличить все элементы на отрезке, то есть все элементы с l -го по r -й).

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

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

<code>addandmax.in</code>	<code>addandmax.out</code>
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 В. Невозрастающая подпоследовательность

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

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (т.е. такую последовательность чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длиной $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделенные друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке выходного файла число k — длину максимальной невозрастающей подпоследовательности. Во второй строке должны быть выведены все номера элементов исходной последовательности i_j , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

Примеры

<code>subseq.in</code>	<code>subseq.out</code>
5	3
5 8 10 4 1	1 4 5

Задача С. 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 D. RMQ Наоборот

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

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 Е. 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.

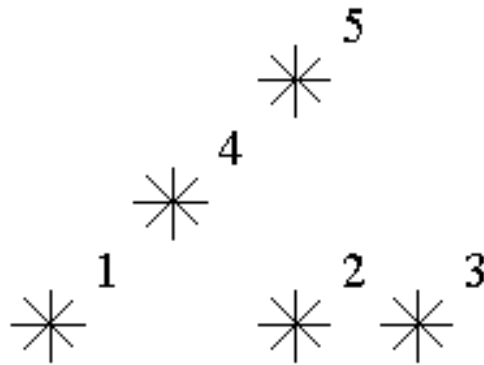
Примеры

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 F. 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 G. Кислотные дожди

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Для сборки лаборатории-поселения на Венеру доставлены n блоков. Блоки расположены в ряд, i -й блок имеет высоту h_i .

Сборку будет осуществлять специальный робот. В процессе сборки последовательные сегменты блоков будут постепенно объединяться. При этом порядок блоков в ряду не будет меняться.

Исходно каждый блок представляет собой отдельный сегмент, сегменты пронумерованы от 1 до n в том же порядке, что и блоки. Если есть два соседних сегмента, составленных из блоков: сегмент из блоков $A = [i, i + 1, \dots, i + p - 1]$ и сегмент из блоков $B = [i + p, i + p + 1, \dots, i + p + q - 1]$, то после их объединения в один получается сегмент $AB = [i, i + 1, \dots, i + p - 1, i + p, i + p + 1, \dots, i + p + q - 1]$.

Инструкция по сборке состоит из $n - 1$ инструкций. Каждая инструкция характеризуется одним числом, j -я инструкция характеризуется числом k_j . После выполнения этой инструкции сегменты с номерами k_j и $k_j + 1$ объединяются в один, получившийся сегмент занимает место в последовательности сегментов на месте двух объединенных сегментов, и вводится новая нумерация на сегментах в том порядке, в котором они расположены — номера сегментов, начиная с $k_j + 2$, уменьшаются на один. После выполнения всех инструкций все сегменты окажутся объединены в один общий сегмент.

На Венере постоянно идут кислотные дожди, поэтому в процессе сборки важно для каждого сегмента блоков понимать, сколько жидкости может скопиться в этом сегменте. Пусть сегмент состоит из блоков высотой h_l, h_{l+1}, \dots, h_r . Для p , где $l \leq p \leq r$ определим *глубину* блока с высотой h_p в этом сегменте следующим образом. Посчитаем величины $l_p = \max\{h_l, \dots, h_p\}$, $r_p = \max\{h_p, \dots, h_r\}$. Это самые высокие блоки в сегменте слева и справа от p -го. Тогда глубина блока p в его сегменте равна $d_p = \min(l_p, r_p) - h_p$, заметим, что $d_p \geq 0$. *Емкостью* сегмента будем называть сумму глубин блоков этого сегмента, то есть $w = d_l + d_{l+1} + \dots + d_r$.

Задана последовательность объединений сегментов. После каждого объединения выведите емкость получившегося сегмента.

Рисунок на следующей странице показывает процесс выполнения инструкции из примера, над каждым блоком указана его глубина, а для нового сегмента показана его емкость.

Формат входных данных

Первая строка содержит одно целое число n — количество блоков ($2 \leq n \leq 10^5$).

Во второй строке записано n чисел h_1, \dots, h_n ($1 \leq h_i \leq 10^9$).

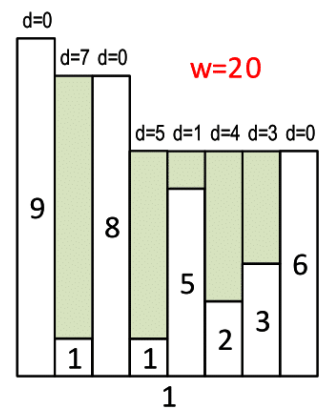
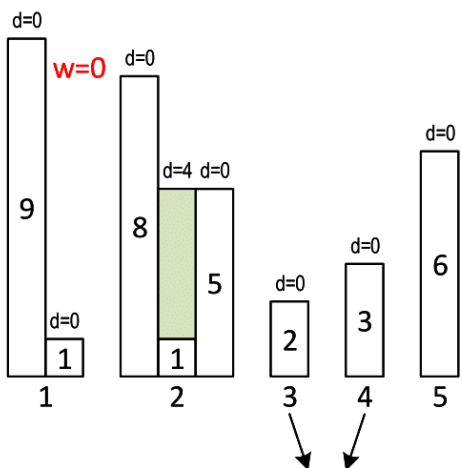
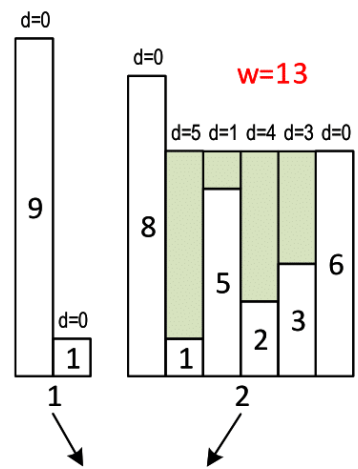
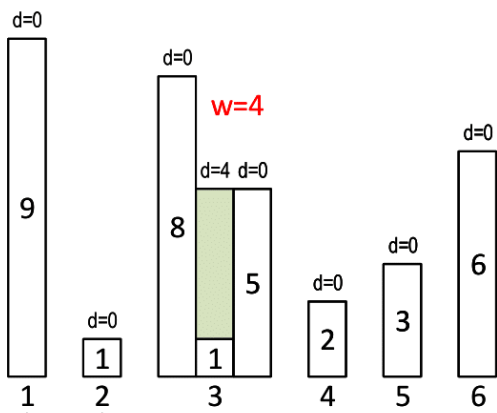
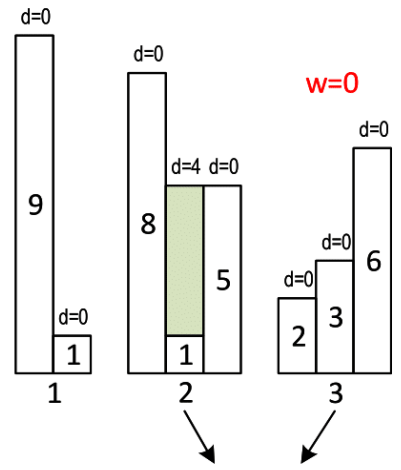
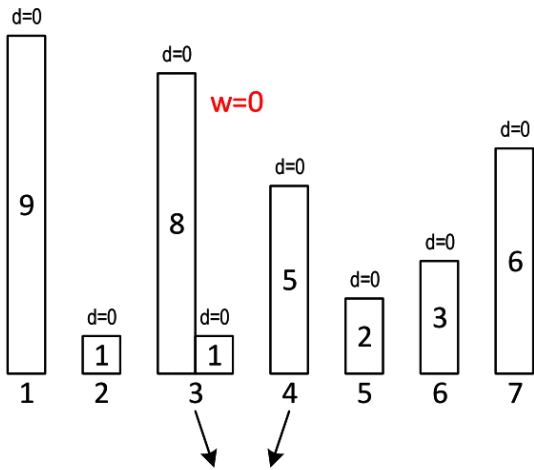
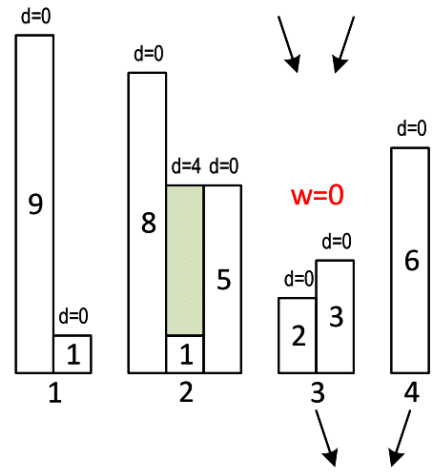
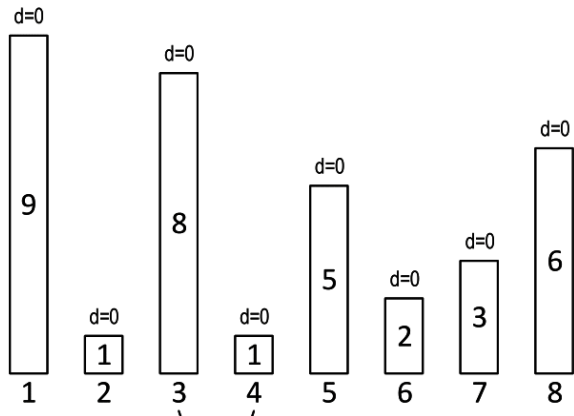
В третьей строке записаны $n - 1$ чисел — инструкции по объединению сегментов. Каждая инструкция характеризуется одним числом k_j ($1 \leq k_j \leq n - j$).

Формат выходных данных

Выведите $n - 1$ чисел — после каждого объединения сегментов выведите емкость получившегося объединенного сегмента.

Пример

стандартный ввод	стандартный вывод
8	0
9 1 8 1 5 2 3 6	4
3 3 1 3 3 2 1	0
	0
	0
	13
	20



Задача Н. 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 I. 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.

