

## Задача А. Сумма на отрезке

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

Дан массив из  $N$  элементов, нужно научиться находить сумму чисел на отрезке.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $N$  и  $K$  — количество чисел в массиве и количество запросов ( $1 \leq N \leq 100\,000$ ,  $0 \leq K \leq 100\,000$ ). Следующие  $K$  строк содержат следующие запросы:

1. `A i x` — присвоить  $i$ -му элементу массива значение  $x$  ( $1 \leq i \leq n$ ,  $0 \leq x \leq 10^9$ );
2. `Q l r` — найти сумму чисел в массиве на позициях от  $l$  до  $r$  ( $1 \leq l \leq r \leq n$ ).

Изначально в массиве живут нули.

### Формат выходных данных

На каждый запрос вида `Q l r` нужно вывести единственное число — сумму на отрезке.

### Примеры

sum.in	sum.out
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

### Замечание

TL для Python 4 секунды

## Задача В. Поиск максимума

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

Реализуйте структуру данных для эффективного вычисления номера максимального из нескольких подряд идущих элементов массива.

### Формат входных данных

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

Во второй строке вводятся  $N$  чисел от 1 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число  $K$  ( $1 \leq K \leq 30\,000$ ) — количество запросов на вычисление максимума.

В следующих  $K$  строках вводится по два числа — номера левого и правого элементов отрезка массива (считается, что элементы массива нумеруются с единицы).

### Формат выходных данных

Для каждого запроса выведите индекс максимального элемента на указанном отрезке массива. Если максимальных элементов несколько, выведите любой их них.

Числа выводите в одну строку через пробел.

### Примеры

<code>index-max.in</code>	<code>index-max.out</code>
5 2 2 2 1 5 2 2 3 2 5	3 5

### Замечание

TL для Python 2 секунды

## Задача С. 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  $N$  ( $1 \leq N \leq 200\,000$ ) — 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

## Задача D. Знакопереживание

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

Реализуйте структуру данных из  $n$  элементов  $a_1, a_2, \dots, a_n$ , поддерживающую следующие операции:

- присвоить элементу  $a_i$  значение  $j$ ;
- найти знакопереживающую сумму на отрезке от  $l$  до  $r$  включительно, т. е.  $(a_l - a_{l+1} + a_{l+2} - \dots a_r)$ .

### Формат входных данных

В первой строке входного файла содержится натуральное число  $n$  ( $1 \leq n \leq 10^5$ ) — длина массива. Во второй строке записаны начальные значения элементов — неотрицательные целые числа, не превосходящие  $10^4$ .

В третьей строке находится натуральное число  $m$  ( $1 \leq m \leq 10^5$ ) — количество операций. В последующих  $m$  строках записаны операции:

- операция первого типа задаётся тремя числами  $0 \ i \ j$  ( $1 \leq i \leq n, 1 \leq j \leq 10^4$ ).
- операция второго типа задаётся тремя числами  $1 \ l \ r$  ( $1 \leq l \leq r \leq n$ ).

### Формат выходных данных

Для каждой операции второго типа выведите на отдельной строке соответствующую знакопереживающую сумму.

### Пример

signchange.in	signchange.out
3	-1
1 2 3	2
5	-1
1 1 2	3
1 1 3	
1 2 3	
0 2 1	
1 1 3	

### Замечание

TL для Python 4 секунды

## Задача Е. 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 8 секунд

## Задача F. Марио и трубы

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

Марио собирается проходить уровень, состоящий из  $N$  последовательно расположенных труб, высота  $i$ -й трубы —  $a_i$ . Он еще не знает, где он будет располагаться изначально, и куда ему надо добраться, поэтому хочет рассмотреть несколько вариантов.

Находясь на трубе, Марио может переместиться только на соседние трубы слева и справа (если они существуют). Спускаться он может с любой высоты, также он может перемещаться между одинаковыми трубами. Подниматься Марио может только на трубу, высота которой больше высоты текущей на 1. Более формально, Марио может переместиться с трубы  $i$  на трубу  $j$ , если  $|i - j| = 1$  и  $a_j - a_i \leq 1$ .

Однако злой динозавр Боузер хочет помешать Марио пройти уровень, для чего иногда увеличивает высоту нескольких подряд идущих труб на одно число  $k$ . Теперь Марио не может понять, удастся ли ему пройти уровень и поэтому просит вас обрабатывать два типа запросов — Боузер изменяет высоту некоторых труб, и Марио пытается пройти от одной трубы до другой.

### Формат входных данных

В первой строке заданы два целых числа  $N$  и  $M$  — число труб и число запросов соответственно ( $2 \leq N \leq 3 \cdot 10^5, 1 \leq M \leq 10^6$ ).

Следующая строка содержит  $N$  целых чисел  $a_i$  — высоты труб на уровне ( $1 \leq a_i \leq 10^9$ ).

Далее идут  $M$  строк, содержащие описание запросов. Каждая строка имеет вид:

- $1 \ x \ y$  — может ли Марио пройти от трубы с номером  $x$  до трубы с номером  $y$  ( $1 \leq x, y \leq N$ ). Гарантируется, что номера  $x$  и  $y$  не совпадают.
- $2 \ l \ r \ d$  — Боузер увеличивает высоты труб с  $l$ -й до  $r$ -й на величину  $d$  ( $1 \leq l \leq r \leq N, -10^9 \leq d \leq 10^9$ ).

### Формат выходных данных

Для каждого запроса первого типа нужно на отдельной строке вывести «YES», если Марио может дойти от одной трубы до другой и «NO» в противном случае (без кавычек).

### Примеры

стандартный ввод	стандартный вывод
5 7	YES
1 2 3 4 5	NO
1 5 1	NO
2 2 4 3	YES
1 5 4	NO
1 1 3	
2 2 3 3	
1 2 4	
1 1 3	

## Задача G. 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 Н. 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 I. Ксюша и экзамен по алгебре

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Ксюша поступила в престижный университет на Кипре. Программа первого семестра по алгебре содержала только лишь темы «Умножение целых чисел» и «Деление целых чисел». Ксюша училась в лучшей школе страны и уже проходила эти темы, поэтому она решила прогуливать лекции и семинары по алгебре.

Когда выпал снег и наступила сессия, Ксюша пришла на экзамен. Экзаменатор был возмущён пропусками Ксюши и решил её завалить, дав самую сложную задачу.

Дан массив из  $n$  целых чисел  $a_1, a_2, \dots, a_n$ . Необходимо ответить на  $m$  запросов. Запросы бывают трёх типов:

1. посчитать  $a_l \cdot a_{l+1} \cdot \dots \cdot a_r$  по модулю  $10^9 + 7$ ;
2. поделить каждое из чисел  $a_l, a_{l+1}, \dots, a_r$  на его минимальный **нечётный** делитель, больший 1; если такого нет, число не меняется;
3. присвоить  $a_i = x$ .

Ксюша не хочет отчисляться и уезжать с Кипра, помогите ей сдать экзамен.

### Формат входных данных

В первой строке дано целое число  $t$  ( $1 \leq t \leq 10^4$ ) — количество наборов входных данных. Далее следует описание наборов.

В первой строке дано целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — количество элементов в массиве.

Во второй строке даны  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 5 \cdot 10^5$ ) — элементы массива.

В третьей строке дано целое число  $m$  ( $1 \leq m \leq 2 \cdot 10^5$ ) — количество запросов.

В следующих  $m$  строках даны запросы:

- $? \ l \ r$  ( $1 \leq l \leq r \leq n$ ) — запрос произведения по модулю  $10^9 + 7$  на отрезке;
- $/ \ l \ r$  ( $1 \leq l \leq r \leq n$ ) — запрос массового деления на отрезке;
- $= \ i \ x$  ( $1 \leq i \leq n, 1 \leq x \leq 5 \cdot 10^5$ ) — запрос изменения элемента.

Гарантируется, что сумма  $n$  по всем наборам входных данных не превосходит  $2 \cdot 10^5$ . То же самое гарантируется для  $m$ .

### Формат выходных данных

Для каждого набора входных данных выведите ответы на запросы типа «?» в отдельных строках.

## Пример

стандартный ввод	стандартный вывод
2	610080
4	160
15 62 41 16	32
8	3648
? 1 4	950998698
/ 1 4	61732212
? 1 4	26
/ 1 4	
? 1 4	
/ 1 4	
= 2 228	
? 1 4	
6	
1 500000 100500 1234 777 101	
7	
= 3 303	
? 2 4	
/ 1 3	
? 1 5	
/ 2 6	
= 5 13	
? 4 6	

## Замечание

В первом наборе входных данных после применения первой операции деления массив трансформируется так:

$$15, 62, 41, 16 \rightarrow \frac{15}{3}, \frac{62}{31}, \frac{41}{41}, 16 = 5, 2, 1, 16$$