

## Задача А. Сумма на отрезке

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
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:

- A  $i$   $x$  — присвоить  $i$ -му элементу массива значение  $x$  ( $1 \leq i \leq n$ ,  $0 \leq x \leq 10^9$ );
- 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.

### Пример

стандартный ввод	стандартный вывод
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	

## Задача В. RMQ

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

Реализуйте структуру данных, которая на данном массиве из  $N$  целых чисел позволяет узнать максимальное значение на этом массиве и индекс элемента, на котором достигается это максимальное значение.

### Формат входных данных

В первой строке вводится натуральное число  $N$  ( $1 \leq N \leq 10^5$ ) – количество элементов в массиве. В следующей строке содержатся  $N$  целых чисел, не превосходящих по модулю  $10^9$  – элементы массива. Гарантируется, что в массиве нет одинаковых элементов. Далее идет число  $K$  ( $0 \leq K \leq 10^5$ ) – количество запросов к структуре данных. Каждая из следующих  $K$  строк содержит два целых числа  $l$  и  $r$  ( $1 \leq l \leq r \leq N$ ) – левую и правую границы отрезка в массиве для данного запроса.

### Формат выходных данных

Для каждого из запросов выведите два числа: наибольшее значение среди элементов массива на отрезке от  $l$  до  $r$  и индекс одного из элементов массива, принадлежащий отрезку от  $l$  до  $r$ , на котором достигается этот максимум.

### Пример

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

## Задача C. НОД на подотрезках с изменением элемента

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

Реализуйте эффективную структуру данных, позволяющую изменять элементы массивы и вычислять НОД нескольких подряд идущих элементов.

### 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 ( $s$  – вычислить НОД,  $u$  – обновить значение элемента).

Следом за  $s$  вводятся два числа – номера левой и правой границы отрезка.

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.

### Пример

стандартный ввод	стандартный вывод
5	2 4 4 32
2 8 4 16 12	
5	
s 1 5	
s 4 5	
u 3 32	
s 2 5	
s 3 3	

## Задача D. Количество максимумов на отрезке

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

Дан массив из  $N$  элементов, нужно научиться находить количество элементов на отрезке  $[l, r]$ , равных максимуму на этом отрезке.

### 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  $10^9$  — элементы массива.

В третьей строке вводится одно натуральное число  $K$  ( $1 \leq K \leq 100\,000$ ) — количество запросов.

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 количество максимумов на позициях от  $l$  до  $r$  ( $1 \leq l \leq r \leq n$ ).

### Формат выходных данных

На каждый запрос вида Q  $l$   $r$  нужно вывести единственное число — количество чисел на позициях от  $l$  до  $r$ , равных максимуму из чисел этого отрезка.

### Пример

стандартный ввод	стандартный вывод
5	2 1 3 2 1 1 1
0 1 1 3 4	
9	
Q 1 3	
Q 3 5	
A 4 1	
Q 1 4	
Q 3 4	
A 2 10	
Q 1 2	
Q 1 5	
Q 4 5	

## Задача E. Количество различных на отрезке

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

Дан массив  $a$ , состоящий из небольших целых чисел ( $1 \leq a_i \leq 40$ ). Нужно научиться находить количество различных элементов на отрезке.

### 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 40 — элементы массива.

В третьей строке вводится одно натуральное число  $K$  ( $1 \leq K \leq 100\,000$ ) — количество запросов.

Следующие  $K$  строк содержат следующие запросы:

- A  $i$   $x$  — присвоить  $i$ -му элементу массива значение  $x$  ( $1 \leq i \leq n$ ,  $1 \leq x \leq 40$ );
- Q  $l$   $r$  — найти количество различных чисел на позициях от  $l$  до  $r$  ( $1 \leq l \leq r \leq n$ ).

### Формат выходных данных

На каждый запрос вида Q  $l$   $r$  нужно вывести единственное число — количество различных чисел на позициях от  $l$  до  $r$ .

### Пример

стандартный ввод	стандартный вывод
7	3
1 2 3 6 5 4 19	4
6	3
Q 1 3	5
Q 2 5	2
Q 2 4	
A 2 3	
Q 1 6	
Q 1 3	

## Задача F. Катый ноль

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
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.

### Пример

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

## Задача G. Количество инверсий

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Напишите программу, которая для заданного массива  $A = [a_1, a_2, \dots, a_n]$  находит количество пар  $(i, j)$  таких, что  $i < j$  и  $a_i > a_j$ .

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество элементов массива. Вторая строка содержит  $n$  элементов массива  $A$  — целых неотрицательных чисел, не превосходящих  $10^9$ .

Гарантируется, что все числа  $a_i$  попарно различны.

### Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
5 6 11 18 28 31	0
7 3 1 4 15 9 2 6	8

## Задача Н. Противник слаб

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Римляне снова наступают. На этот раз их гораздо больше чем персов, но Шапур готов победить их. Он говорит: «Лев никогда не испугается сотни овец».

Не смотря на это, Шапур должен найти слабость римской армии чтобы победить ее. Как вы помните, Шапур — математик, поэтому он определяет насколько слаба армии как число — степень слабости.

Шапур считает, что степень слабости армии равна количеству таких троек  $i, j, k$ , что  $i < j < k$  и  $a_i > a_j > a_k$ , где  $a_x$  — сила человека, стоящего в строю на месте с номером  $x$ .

Помогите Шапуру узнать, насколько слаба армия римлян.

### Формат входных данных

В первой строке записано одно целое число  $n$  ( $3 \leq n \leq 10^6$ ) — количество солдат в римской армии. Следующая строка содержит  $n$  целых чисел  $a_i$  ( $1 \leq i \leq n, 1 \leq a_i \leq 10^9$ ) — силы людей в римской армии.

### Формат выходных данных

Выведите одно число — степень слабости римской армии.

### Примеры

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



## Задача I. Марио и трубы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
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 ( $2 \leq n \leq 300\,000$ ,  $1 \leq m \leq 10^6$ ).

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 пройти от одной трубы до другой и «No» в противном случае (без кавычек).

### Пример

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 J. Знакочередование

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
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 знакопередающуюся 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 знакопередающуюся сумму.

### Пример

stdin	stdout
3	-1
1 2 3	2
5	-1
1 1 2	3
1 1 3	
1 2 3	
0 2 1	
1 1 3	

## Задача К. Число возрастающих подпоследовательностей

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

Задана последовательность из  $n$  чисел  $a_1, a_2, \dots, a_n$ . Подпоследовательностью длины  $k$  этой последовательности называется набор индексов  $i_1, i_2, \dots, i_k$ , удовлетворяющий неравенствам  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ . Подпоследовательность называется возрастающей, если выполняются неравенства  $a_{i_1} < a_{i_2} < \dots < a_{i_k}$ .

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности  $a_1, \dots, a_n$ . Так как это число может быть достаточно большим, необходимо найти остаток от его деления на  $10^9 + 7$ .

### Формат входных данных

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ). Вторая строка входного файла содержит  $n$  целых чисел:  $a_1, a_2, \dots, a_n$ . Все  $a_i$  не превосходят  $10^9$  по абсолютной величине.

### Формат выходных данных

В выходной файл выведите ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8