

## Задача А. Дерево отрезков с операцией на отрезке

Имя входного файла: `segment-tree.in`  
Имя выходного файла: `segment-tree.out`  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 64 мегабайта

Реализуйте эффективную структуру данных для хранения элементов и увеличения нескольких подряд идущих элементов на одно и то же число.

### 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 0 до 100 000 — элементы массива.

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

Каждая из следующих  $M$  строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса ( $g$  — получить текущее значение элемента по его номеру,  $a$  — увеличить все элементы на отрезке).

Следом за  $g$  вводится одно число — номер элемента.

Следом за  $a$  вводится три числа — левый и правый концы отрезка и число  $add$ , на которое нужно увеличить все элементы данного отрезка массива ( $1 \leq add \leq 100\,000$ ).

### Формат выходных данных

Выведите в одну строку через пробел ответы на каждый запрос  $g$ .

### Примеры

<code>segment-tree.in</code>	<code>segment-tree.out</code>
5	4
2 4 3 5 2	2
5	14
g 2	5
g 5	
a 1 3 10	
g 2	
g 4	

## Задача В. Range Variation Query

Имя входного файла: `rvq.in`  
Имя выходного файла: `rvq.out`  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 64 мегабайта

В начальный момент времени последовательность  $a_n$  задана следующей формулой:  $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$ .

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значениями среди элементов  $a_i, a_{i+1}, \dots, a_j$ ;
- присвоить элементу  $a_i$  значение  $j$ .

### Формат входных данных

Первая строка входного файла содержит натуральное число  $k$  — количество запросов ( $1 \leq k \leq 100\,000$ ). Следующие  $k$  строк содержат запросы, по одному на строке. Запрос номер  $i$  описывается двумя целыми числами  $x_i, y_i$ .

Если  $x_i > 0$ , то требуется найти разность между максимальным и минимальным значениями среди элементов  $a_{x_i}, \dots, a_{y_i}$ . При этом  $1 \leq x_i \leq y_i \leq 100\,000$ .

Если  $x_i < 0$ , то требуется присвоить элементу  $a_{|x_i|}$  значение  $y_i$ . В этом случае  $-100\,000 \leq x_i \leq -1$  и  $|y_i| \leq 100\,000$ .

### Формат выходных данных

Для каждого запроса первого типа в выходной файл требуется вывести одну строку, содержащую разность между максимальным и минимальным значениями на соответствующем отрезке.

### Примеры

rvq.in	rvq.out
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	

## Задача С. Разреженные таблицы

Имя входного файла: `sparse.in`  
Имя выходного файла: `sparse.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дан массив из  $n$  чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между  $u$  и  $v$  включительно.

### Формат входных данных

В первой строке входного файла даны три натуральных числа  $n, m$  ( $1 \leq n \leq 10^5, 1 \leq m \leq 10^7$ ) и  $a_1$  ( $0 \leq a_1 < 16\,714\,589$ ) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа  $u_1$  и  $v_1$  ( $1 \leq u_1, v_1 \leq n$ ) — первый запрос.

Элементы  $a_2, a_3, \dots, a_n$  задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при  $n = 10, a_1 = 12345$  получается следующий массив:  $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$ .

Запросы генерируются следующим образом:

$$\begin{aligned} u_{i+1} &= ((17 \cdot u_i + 751 + ans_i + 2i) \bmod n) + 1, \\ v_{i+1} &= ((13 \cdot v_i + 593 + ans_i + 5i) \bmod n) + 1, \end{aligned}$$

где  $ans_i$  — ответ на запрос номер  $i$ .

Обратите внимание, что  $u_i$  может быть больше, чем  $v_i$ .

### Формат выходных данных

В выходной файл выведите  $u_m, v_m$  и  $ans_m$  (последний запрос и ответ на него).

### Примеры

<code>sparse.in</code>	<code>sparse.out</code>
10 8 12345 3 9	5 3 1565158

## Задача D. 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	