

## Задача А. 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 В. Четвертый этаж

Имя входного файла: floor4.in  
Имя выходного файла: floor4.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Знаете ли вы, почему четвертый этаж заперт и там не останавливается лифт? Потому что на самом деле четвертый, запертый, этаж, где не останавливается лифт, содержит бесконечное количество комнат, пронумерованных натуральными числами. На этот этаж регулярно приезжают дети, каждый из которых заранее выбрал, в какую комнату он хочет заселиться. Если выбранная комната оказывается свободна, то ребенок занимает ее, в противном случае он занимает первую свободную комнату с большим номером.

Кроме того, некоторые дети уезжают в середине смены. Сразу после отъезда ребенка его комната становится доступна для заселения следующего.

Промоделируйте работу преподавателей, ответственных за четвертый этаж и научитесь быстро сообщать приезжающим детям, какую комнату им следует занимать.

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  — количество прибытий и отъездов, происходящих в течение смены ( $1 \leq n \leq 100\,000$ ).

Следующие  $n$  строк содержат информацию об ЛКШатах. Число  $a > 0$  обозначает, что приехал школьник, желающий занять комнату номер  $a$  ( $1 \leq a \leq 100\,000$ ). Число  $a < 0$  обозначает, что из комнаты номер  $|a|$  уехал школьник. (Гарантируется, что эта комната не была пуста).

### Формат выходных данных

Для каждого приезжающего школьника выведите одно натуральное число — номер комнаты, в которую он поселится.

### Примеры

floor4.in	floor4.out
6	5
5	6
5	7
5	6
-6	8
5	
5	

## Задача С. 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 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	