

## Задача А. Наивный RMQ

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

Вам дан массив  $a[]$  из  $n$  элементов и  $m$  – количество запросов, состоящих из двух чисел  $l$  и  $r$ , выведите минимальный элемент в массиве  $a[]$  с индексами от  $l$  до  $r$ .

### Формат входного файла

В первой строке входного файла содержится число  $n$  – длина массива  $a[]$  ( $1 \leq n \leq 1000$ ). Во второй строке содержится  $n$  чисел  $a_i$  – элементы массива ( $-10^9 \leq a_i \leq 10^9$ ). В следующей строке содержится число  $m$  – количество запросов ( $1 \leq m \leq 1000$ ). В последних  $m$  строках содержатся запросы  $l, r$  ( $1 \leq l, r \leq n$ ).

### Формат выходного файла

Для каждого запроса выведите ответ в новой строке.

### Примеры

stupid_rmq.in	stupid_rmq.out
5	-2
1 -2 3 4 1	3
3	-2
1 3	
3 4	
1 5	
2	-100000
-100000 -1111111	-1111111
2	
1 1	
1 2	

## Задача В. Запрос изменения на отрезке

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

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 значением среди элементов  $a_i, a_{i+1}, \dots, a_j$ ;
- присвоить элементу  $a_i$  значение  $j$ .

### Формат входного файла

Первая строка входного файла содержит натуральное число  $k$  — количество запросов ( $k \leq 100\,000$ ). 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 значением 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  $y_i$ . При этом  $-100\,000 \leq x_i \leq -1$  и  $|y_i| \leq 100\,000$ .

### Формат выходного файла

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

### Пример

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. Сумма

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

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

### Формат входного файла

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

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

Изначально массив заполнен нулями.

### Формат выходного файла

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

### Примеры

sum.in	sum.out
5 9	3
A 2 3 2	2
A 3 5 1	3
A 4 5 2	4
Q 1 3	2
Q 2 2	7
Q 3 4	
Q 4 5	
Q 5 5	
Q 1 5	

## Задача D. Сумма 2

Имя входного файла: `sum2.in`  
Имя выходного файла: `sum2.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вам дан массив  $a[]$  из  $n$  элементов и  $m$  – количество запросов, состоящих из двух чисел  $l$  и  $r$ , выведите сумму элементов в массиве  $a[]$  с индексами от  $l$  до  $r$ .

### Формат входного файла

В первой строке входного файла содержится число  $n$  – длина массива  $a[]$  ( $1 \leq n \leq 100000$ ). Во второй строке содержится  $n$  чисел  $a_i$  – элементы массива ( $-10^9 \leq a_i \leq 10^9$ ). В следующей строке содержится число  $m$  – количество запросов ( $1 \leq m \leq 100000$ ). В последних  $m$  строках содержатся запросы  $l, r$  ( $1 \leq l, r \leq n$ ).

### Формат выходного файла

Для каждого запроса выведите ответ в новой строке.

### Примеры

sum2.in	sum2.out
5 1 2 3 4 5 3 1 5 1 1 4 5	15 1 9
6 -1 1 -1 1 -1 1 2 1 6 3 3	0 -1

## Задача Е. Разрезанные таблицы

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

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 входного файла

В первой строке входного файла даны три целых числа  $n$ ,  $m$  ( $1 \leq n \leq 10^5, 1 \leq m \leq 10^7$ ) и  $a_1$  ( $0 \leq a_1 < 16714589$ ) — 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  $i$ .

### Формат выходного файла

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

### Пример

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

## Задача RMQ. Range Minimum Query

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

Компания *Giggle* открывает свой новый офис в Судиславле, и вы приглашены на собеседование. Ваша задача — решить поставленную задачу<sup>1</sup>.

Вам нужно создать структуру данных, которая представляет из себя массив целых чисел. Изначально массив пуст. Вам нужно поддерживать две операции:

- запрос: «?  $i$   $j$ » — возвращает минимальный элемент между  $i$ -ым и  $j$ -м, включительно;
- изменение: «+  $i$   $x$ » — добавить элемент  $x$  после  $i$ -го элемента списка. Если  $i = 0$ , то элемент добавляется в начало массива.

Конечно, эта структура должна быть достаточно хорошей.

### Формат входного файла

Первая строка входного файла содержит единственное целое число  $n$  — число операций над массивом ( $1 \leq n \leq 200\,000$ ). Следующие  $n$  строк описывают сами операции. Все операции добавления являются корректными. Все числа, хранящиеся в массиве, по модулю не превосходят  $10^9$ .

### Формат выходного файла

Для каждой операции в отдельной строке выведите её результат.

### Пример

<code>rmq.in</code>	<code>rmq.out</code>
8	4
+ 0 5	3
+ 1 3	1
+ 1 4	
? 1 2	
+ 0 2	
? 2 4	
+ 4 1	
? 3 5	

Нижеследующая таблица показывает процесс изменения массива из примера.

Операция	Массив после её выполнения
<i>изначально</i>	<i>пуст</i>
+ 0 5	5
+ 1 3	5, 3
+ 1 4	5, 4, 3
+ 0 2	2, 5, 4, 3
+ 4 1	2, 5, 4, 3, 1

---

<sup>1</sup>Капитан Очевидность намекает