

## Содержание

<b>Больше AC!</b>	<b>2</b>
1 Задача А. Площадь и прямоугольники [1 секунда, 64 mb]	2
<b>Обязательные задачи</b>	<b>3</b>
2 Задача В. Неявный Ключ [3 секунда, 256 mb]	3
3 Задача С. И снова сумма... [2 секунды, 256 mb]	4
<b>Интересные задачи</b>	<b>5</b>
4 Задача D. Persistent Array [2 секунды, 256 mb]	5
5 Задача E. Приказы [5 секунд, 256 mb]	6

## Больше АС!

### 1 Задача А. Площадь и прямоугольники [1 секунда, 64 mb]

На плоскости задано  $N$  прямоугольников с вершинами в точках с целыми координатами и сторонами, параллельными осям координат. Необходимо посчитать площадь их пересечения.

#### Формат входных данных

В первой строке входного файла указано число  $N$  ( $1 \leq N \leq 1500$ ). В следующих  $N$  строках заданы по 4 целых числа  $x_1, y_1, x_2, y_2$  — сначала координаты левого нижнего угла прямоугольника, потом правого верхнего ( $-10^9 \leq x_1 \leq x_2 \leq 10^9, -10^9 \leq y_1 \leq y_2 \leq 10^9$ ). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

#### Формат выходных данных

Выведите требуемое число.

#### Примеры

стандартный ввод	стандартный вывод
2 0 0 2 2 1 1 3 3	1

## Обязательные задачи

### 2 Задача В. Неявный Ключ [3 секунда, 256 mb]

Научитесь быстро делать две операции с массивом:

- `add i x` — добавить после  $i$ -го элемента  $x$  ( $0 \leq i \leq n$ )
- `del i` — удалить  $i$ -й элемент ( $1 \leq i \leq n$ )

#### Формат входных данных

На первой строке  $n_0$  и  $m$  ( $1 \leq n_0, m \leq 10^5$ ) — длина исходного массива и количество запросов. На второй строке  $n_0$  целых чисел от 0 до  $10^9 - 1$  — исходный массив. Далее  $m$  строк, содержащие запросы. Гарантируется, что запросы корректны: например, если просят удалить  $i$ -й элемент, он точно есть.

#### Формат выходных данных

Выведите конечное состояние массива. На первой строке количество элементов, на второй строке сам массив.

#### Примеры

стандартный ввод	стандартный вывод
3 4	3
1 2 3	9 2 8
del 3	
add 0 9	
add 3 8	
del 2	

### 3 Задача С. И снова сумма... [2 секунды, 256 mb]

Реализуйте структуру данных, которая поддерживает множество  $S$  целых чисел, с которым разрешается производить следующие операции:

- $add(i)$  — добавить в множество  $S$  число  $i$  (если он там уже есть, то множество не меняется);
- $sum(l, r)$  — вывести сумму всех элементов  $x$  из  $S$ , которые удовлетворяют неравенству  $l \leq x \leq r$ .

#### Формат входных данных

Исходно множество  $S$  пусто. Первая строка входного файла содержит  $n$  — количество операций ( $1 \leq n \leq 300\,000$ ). Следующие  $n$  строк содержат операции. Каждая операция имеет вид либо «+  $i$ », либо «?  $l$   $r$ ». Операция «?  $l$   $r$ » задает запрос  $sum(l, r)$ .

Если операция «+  $i$ » идет во входном файле в начале или после другой операции «+», то она задает операцию  $add(i)$ . Если же она идет после запроса «?», и результат этого запроса был  $y$ , то выполняется операция  $add((i + y) \bmod 10^9)$ .

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до  $10^9$ .

#### Формат выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

#### Пример

стандартный ввод	стандартный вывод
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

## Интересные задачи

### 4 Задача D. Persistent Array [2 секунды, 256 mb]

Дан массив (вернее, первая, начальная его версия).

Нужно уметь отвечать на два запроса:

○  $a_i[j] = x$  — создать из  $i$ -й версии новую, в которой  $j$ -й элемент равен  $x$ , а остальные элементы такие же, как в  $i$ -й версии.

○  $\text{get } a_i[j]$  — сказать, чему равен  $j$ -й элемент в  $i$ -й версии.

#### Формат входных данных

Количество чисел в массиве  $N$  ( $1 \leq N \leq 10^5$ ) и  $N$  элементов массива. Далее количество запросов  $M$  ( $1 \leq M \leq 10^5$ ) и  $M$  запросов. Формат описания запросов можно посмотреть в примере. Если уже существует  $K$  версий, новая версия получает номер  $K + 1$ . И исходные, и новые элементы массива — целые числа от 0 до  $10^9$ . Элементы в массиве нумеруются числами от 1 до  $N$ .

#### Формат выходных данных

На каждый запрос типа  $\text{get}$  вывести соответствующий элемент нужного массива.

#### Пример

стандартный ввод	стандартный вывод
6	6
1 2 3 4 5 6	5
11	10
create 1 6 10	5
create 2 5 8	10
create 1 5 30	8
get 1 6	6
get 1 5	30
get 2 6	
get 2 5	
get 3 6	
get 3 5	
get 4 6	
get 4 5	

## 5 Задача E. Приказы [5 секунд, 256 mb]

Вася работает в НИИГСД (НИИ Государственных Структур Данных). Он изучает приказы правительства далёкого государства.

В том государстве все города расположены вдоль одной дороги. Они пронумерованы в порядке обхода. Изначально качество жизни в каждом из них равно нулю.

Далее последовательно издаются указы вида «уровень жизни в городах с  $i$  по  $j$  должен стать не меньше  $x$ ».

Также есть некоторые официальные заявления. Они имеют следующую форму: «средний уровень жизни в городах с  $i$  по  $j$  равен  $x$ ». Вася нуждается в помощи с проверкой этих утверждений: для каждого из них известны  $i$  и  $j$ , требуется подсчитать верное значение  $x$ .

Можете считать, что каждый приказ исполняется, а также в каждый момент времени каждый город имеет минимальный неотрицательный уровень жизни, удовлетворяющий всем приказам.

### Формат входных данных

Ввод состоит из одного или более тестов. Каждый тест начинается строкой с двумя целыми числами  $n$  и  $k$  — числом городов и событий, соответственно. Следующие  $k$  строк содержат по одному описанию события:

- $\wedge i j x$  означает приказ: после этого, все города с номерами от  $i$  до  $j$  включительно должны иметь уровень жизни не менее  $x$  ( $1 \leq x \leq 10^9$ ,  $1 \leq i \leq j \leq n$ ).
- $? i j$  означает официальное заявление: следует подсчитать средний уровень жизни в городах с  $i$  по  $j$  включительно ( $1 \leq i \leq j \leq n$ ).

В конце ввода будет помещён тест с  $n = k = 0$ , который не требуется обрабатывать.

Сумма  $n$  по всему вводу не превысит 100 000. Сумма  $k$  по всему вводу не превысит 100 000.

### Формат выходных данных

Для каждого официального заявления выведите на отдельной строке искомый средний уровень жизни в виде несократимой дроби с наименьшим возможным натуральным знаменателем. Если знаменатель равен 1, выведите вместо дроби целое число. Следуйте формату вывода, как это показано в примере.

### Пример

стандартный ввод	стандартный вывод
10 10	0
? 1 10	1
$\wedge$ 1 10 1	10
? 1 10	10
$\wedge$ 2 3 10	5
$\wedge$ 3 4 5	27/5
? 2 2	16/5
? 3 3	
? 4 4	
? 1 5	
? 1 10	
0 0	