

**Задача А. Сумма на отрезке**

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

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  $i$ -му элементу массива значение  $x$  ( $1 \leq i \leq n$ ,  $0 \leq x \leq 10^9$ )
2. `Q l r` — найти сумму чисел в массиве на позициях от  $l$  до  $r$ . ( $1 \leq l \leq r \leq n$ )

Изначально в массиве живут нули.

**Формат выходного файла**

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

**Примеры**

<code>sum.in</code>	<code>sum.out</code>
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. Окна**

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

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

**Формат входного файла**

В первой строке входного файла записано число окон  $n$  ( $1 \leq n \leq 50\,000$ ). Следующие  $n$  строк содержат координаты окон  $x_{(1,i)} y_{(1,i)} x_{(2,i)} y_{(2,i)}$ , где  $(x_{(1,i)}, y_{(1,i)})$  — координаты левого верхнего угла  $i$ -го окна, а  $(x_{(2,i)}, y_{(2,i)})$  — правого нижнего (на экране компьютера  $y$  растёт сверху вниз, а  $x$  — слева направо). Все координаты — целые числа, по модулю не превосходящие  $10^6$ .

**Формат выходного файла**

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенных пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т. е. покрывающими свои граничные точки.

**Примеры**

<code>windows.in</code>	<code>windows.out</code>
2 0 0 3 3 1 1 4 4	2 1 3
1 0 0 1 1	1 0 1
4 0 0 1 1 0 1 1 2 1 0 2 1 1 1 2 2	4 1 1
5 0 0 1 1 0 1 1 2 0 0 2 2 1 0 2 1 1 1 2 2	5 1 1

**Задача С. Перестановки**

Имя входного файла: `permutation2.in`  
 Имя выходного файла: `permutation2.out`  
 Ограничение по времени: 10 секунды  
 Ограничение по памяти: 512 мегабайт

Вася выписал на доске в каком-то порядке все числа от 1 по  $N$ , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая отвечает на вопрос — сколько среди чисел,

стоящих на позициях с  $x$  по  $y$ , по величине лежат в интервале от  $k$  до  $l$ . Сделайте то же самое.

### Формат входного файла

В первой строке лежит два натуральных числа —  $1 \leq N \leq 500\,000$  — количество чисел, которые выписал Вася и  $1 \leq M \leq 500\,000$  — количество вопросов, которые Вася хочет задать программе. Во второй строке дано  $N$  чисел — последовательность чисел, выписанных Васей. Далее в  $M$  строках находятся описания вопросов. Каждая строка содержит четыре целых числа  $1 \leq x \leq y \leq N$  и  $1 \leq k \leq l \leq N$ .

### Формат выходного файла

Выведите  $M$  строк, каждая должна содержать единственное число — ответ на Васин вопрос.

### Примеры

permutation2.in	permutation2.out
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

### Задача D. Снеговики

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

Зима. 2012 год. На фоне грядущего Апокалипсиса и конца света незамеченной прошла новость об очередном прорыве в областях клонирования и снеговиков: клонирования снеговиков. Вы конечно знаете, но мы вам напомним, что снеговик состоит из нуля или более вертикально поставленных друг на друга шаров, а клонирование — это процесс создания идентичной копии (клона).

В местечке Местячково учитель Андрей Сергеевич Учитель купил через интернет-магазин «Интернет-магазин аппаратов клонирования» аппарат для клонирования снеговиков. Теперь дети могут играть и даже играют во дворе в следующую игру. Время от времени один из них выбирает понравившегося снеговика, клонирует его и:

- либо добавляет ему сверху один шар;
- либо удаляет из него верхний шар (если снеговик не пустой).

Учитель Андрей Сергеевич Учитель записал последовательность действий и теперь хочет узнать суммарную массу всех построенных снеговиков.

### Формат входного файла

Первая строка содержит количество действий  $n$  ( $1 \leq n \leq 200\,000$ ). В строке номер  $i + 1$  содержится описание действия  $i$ :

- $t\ m$  — клонировать снеговика номер  $t$  ( $0 \leq t < i$ ) и добавить сверху шар массой  $m$  ( $0 < m \leq 1000$ );
- $t\ \emptyset$  — клонировать снеговика номер  $t$  ( $0 \leq t < i$ ) и удалить верхний шар. Гарантируется, что снеговик  $t$  не пустой.

В результате действия  $i$ , описанного в строке  $i + 1$  создается снеговик номер  $i$ . Изначально имеется пустой снеговик с номером ноль.

Все числа во входном файле целые.

### Формат выходного файла

Выведите суммарную массу построенных снеговиков.

### Примеры

snowmen.in	snowmen.out
8	74
0 1	
1 5	
2 4	
3 2	
4 3	
5 0	
6 6	
1 0	