

## Задача А. А+В

Имя входного файла: `aplusb.in`  
Имя выходного файла: `aplusb.out`  
Ограничение по времени: 1 second  
Ограничение по памяти: 64 megabytes

Для начала найдите сумму двух целых чисел.

### Формат входного файла

В первой и единственной строке входного файла два числа —  $A$  и  $B$ , по модулю не превышающие  $10^9$ .

### Формат выходного файла

В первой и единственной строке выходного файла одно число, равное сумме  $A$  и  $B$ .

### Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
7 4	11

## 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, разделённых пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т. е. покрывающими свои граничные точки.

### Примеры

windows.in	windows.out
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. Снеговика

Имя входного файла: `snowmen.in`  
Имя выходного файла: `snowmen.out`  
Ограничение по времени: 4 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

<code>snowmen.in</code>	<code>snowmen.out</code>
8	74
0 1	
1 5	
2 4	
3 2	
4 3	
5 0	
6 6	
1 0	

## Задача D. $K$ -я порядковая статистика на отрезке

Имя входного файла: `kth.in`  
Имя выходного файла: `kth.out`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Дан массив из  $N$  неотрицательных чисел, строго меньших  $10^9$ . Вам необходимо ответить на несколько запросов о величине  $k$ -й порядковой статистики на отрезке  $[l, r]$ .

### Формат входного файла

Первая строка содержит число  $N$  ( $1 \leq N \leq 450\,000$ ) — размер массива.

Вторая строка может быть использована для генерации  $a_i$  — начальных значений элементов массива. Она содержит три числа  $a_1, l$  и  $m$  ( $0 \leq a_1, l, m < 10^9$ ); для  $i$  от 2 до  $N$

$$a_i = (a_{i-1} \cdot l + m) \bmod 10^9.$$

В частности,  $0 \leq a_i < 10^9$ .

Третья строка содержит одно целое число  $B$  ( $1 \leq B \leq 1000$ ) — количество групп запросов.

Следующие  $B$  строк описывают одну группу запросов. Каждая группа запросов описывается 10 числами. Первое число  $G$  обозначает количество запросов в группе. Далее следуют числа  $x_1, l_x$  и  $m_x$ , затем  $y_1, l_y$  и  $m_y$ , затем,  $k_1, l_k$  и  $m_k$  ( $1 \leq x_1 \leq y_1 \leq N$ ,  $1 \leq k_1 \leq y_1 - x_1 + 1$ ,  $0 \leq l_x, m_x, l_y, m_y, l_k, m_k < 10^9$ ). Эти числа используются для генерации вспомогательных последовательностей  $x_g$  и  $y_g$ , а также параметров запросов  $i_g, j_g$  и  $k_g$  ( $1 \leq g \leq G$ )

$$\begin{aligned}x_g &= ((i_{g-1} - 1) \cdot l_x + m_x) \bmod N + 1, & 2 \leq g \leq G \\y_g &= ((j_{g-1} - 1) \cdot l_y + m_y) \bmod N + 1, & 2 \leq g \leq G \\i_g &= \min(x_g, y_g), & 1 \leq g \leq G \\j_g &= \max(x_g, y_g), & 1 \leq g \leq G \\k_g &= (((k_{g-1} - 1) \cdot l_k + m_k) \bmod (j_g - i_g + 1)) + 1, & 2 \leq g \leq G\end{aligned}$$

Сгенерированные последовательности описывают запросы,  $g$ -й запрос состоит в поиске  $k_g$ -го по величине числа среди элементов отрезка  $[i_g, j_g]$ .

Суммарное количество запросов не превосходит 600 000.

### Формат выходного файла

Выведите единственное число — сумму ответов на запросы.

### Примеры

kth.in	kth.out
5	15
1 1 1	
5	
1	
1 0 0 3 0 0 2 0 0	
1	
2 0 0 5 0 0 3 0 0	
1	
1 0 0 5 0 0 5 0 0	
1	
3 0 0 3 0 0 1 0 0	
1	
1 0 0 4 0 0 1 0 0	

## Задача Е. Менеджер памяти

Имя входного файла: `memory.in`  
Имя выходного файла: `memory.out`  
Ограничение по времени: 6 секунды  
Ограничение по памяти: 512 мегабайт

Одно из главных нововведений новейшей операционной системы Indows 7 — новый менеджер памяти. Он работает с массивом длины  $N$  и позволяет выполнять три самые современные операции:

- `copy(a, b, l)` — скопировать отрезок длины  $[a, a + l - 1]$  в  $[b, b + l - 1]$
- `sum(l, r)` — посчитать сумму элементов массива на отрезке  $[l, r]$
- `print(l, r)` — напечатать элементы с  $l$  по  $r$ , включительно

Вы являетесь разработчиком своей операционной системы, и Вы, безусловно, не можете обойтись без инновационных технологий. Вам необходимо реализовать точно такой же менеджер памяти.

### Формат входного файла

Первая строка входного файла содержит целое число  $N$  ( $1 \leq N \leq 1\,000\,000$ ) — размер массива, с которым будет работать Ваш менеджер памяти.

Во второй строке содержатся четыре числа  $1 \leq X_1, A, B, M \leq 10^9 + 10$ . С помощью них можно сгенерировать исходный массив чисел  $X_1, X_2, \dots, X_N$ .  $X_{i+1} = (A * X_i + B) \bmod M$

Следующая строка входного файла содержит целое число  $K$  ( $1 \leq K \leq 200\,000$ ) — количество запросов, которые необходимо выполнить Вашему менеджеру памяти.

Далее в  $K$  строках содержится описание запросов. Запросы заданы в формате:

- `cpy a b l` — для операции `copy`
- `sum l r` — для операции `sum` ( $l \leq r$ )
- `out l r` — для операции `print` ( $l \leq r$ )

Гарантируется, что суммарная длина запросов `print` не превышает 3000. Также гарантируется, что все запросы корректны.

### Формат выходного файла

Для каждого запроса `sum` или `print` выведите в выходной файл на отдельной строке результат запроса.

### Примеры

<code>memory.in</code>	<code>memory.out</code>
6	1 2 6 1 2 6
1 4 5 7	1 2 1 2 2 6
7	6
out 1 6	1 1 2 1 2 6
cpy 1 3 2	13
out 1 6	
sum 1 4	
cpy 1 2 4	
out 1 6	
sum 1 6	