

## Задача А. Правое вхождение

Имя входного файла: `right.in`  
Имя выходного файла: `right.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дан массив целых чисел, отсортированных в неубывающем порядке. Напишите программу, которая обрабатывает запросы следующего вида:

- для заданного числа  $x_i$  найти позицию его самого правого вхождения в массив.

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  ( $1 \leq n, m \leq 100\,000$ ). Вторая строка содержит  $n$  элементов массива  $A$ . Оставшиеся  $m$  строк содержат запросы — числа  $x_i$ . Как элементы массива, так и числа в запросе не превосходят по модулю  $10^9$ .

### Формат выходного файла

В выходной файл выведите  $m$  чисел — правые позиции соответствующих чисел в массиве. Если элемент не найден, то выведите ноль.

### Примеры

<code>right.in</code>	<code>right.out</code>
3 3	1
1 3 5	3
1	0
5	
7	

## Задача В. $K$ -й минимум — 2

Имя входного файла: `kth2.in`  
Имя выходного файла: `kth2.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Напишите программу, которая находит  $k$ -е в возрастающем порядке число в массиве  $A = \langle a_1, a_2, \dots, a_n \rangle$ .

Массив  $A$  задается с помощью полинома  $P(x) = 131x^3 + 77x^2 + 1345x + 1577$ :  
 $a_i = P(i) \bmod 2^{32}$ .

### Формат входного файла

Входной файл содержит два натуральных числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 10^7$ ).

### Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

## Примеры

<code>kth2.in</code>	<code>kth2.out</code>
1 1	3130
10 1	3130

### Note

Для вычислений по модулю  $2^{32}$  используйте тип `unsigned int` в C/C++ или `LongWord` в Delphi, игнорируя переполнения.

## Задача С. Количество инверсий

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

Напишите программу, которая для заданного массива  $A = \langle a_1, a_2, \dots, a_n \rangle$  находит количество пар  $(i, j)$  таких, что  $i < j$  и  $a_i > a_j$ .

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 50\,000$ ) — количество элементов массива. Вторая строка содержит  $n$  попарно различных элементов массива  $A$  — целых неотрицательных чисел, не превосходящих  $10^6$ .

### Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

### Примеры

<code>inverse.in</code>	<code>inverse.out</code>
5	0
6 11 18 28 31	
8	28
999994 999989 999982 999972	
999969 999961 999954 999950	

## Задача D. Деки на 6 мегабайтах

Имя входного файла: `deques.in`  
Имя выходного файла: `deques.out`  
Ограничение по времени: 1 second  
Ограничение по памяти: 6 megabytes

Напишите программу, которая умеет оперировать большим количеством деков. Дек — это «очередь с двумя концами».

### Формат входного файла

Первая строка входного файла содержит общее количество команд  $n$  ( $0 \leq n \leq 150\,000$ ). Каждая из следующих  $n$  строк содержит описание команды:

- «pushfront  $A$   $B$ » — вставить число  $B$  в начало дека  $A$ ;
- «pushback  $A$   $B$ » — вставить число  $B$  в конец дека  $A$ ;
- «popfront  $A$ » — удалить первый элемент дека  $A$ ;
- «popback  $A$ » — удалить последний элемент дека  $A$ .

Для каждой команды параметры  $A$  и  $B$  — целые числа от 1 до 150 000 включительно.

### Формат выходного файла

Для каждой команды `popfront` или `popback` выведите удаляемое число. Гарантируется, что перед выполнением команды удаления соответствующий дек не пуст.

### Примеры

deques.in	deques.out
9	71819
pushfront 1 71819	1
pushback 2 71820	11
pushback 1 1	71820
popfront 1	
popfront 1	
pushfront 2 10	
pushback 2 11	
popback 2	
popback 2	