

## Задача А. $k$ -я порядковая статистика

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

Дан массив, содержащий  $n$  целых чисел. Вам нужно найти в этом массиве  $k$ -й по счету минимальный элемент ( $k = 0..n - 1$ ), то есть элемент, который после сортировки массива по неубыванию окажется на  $k$ -м месте от начала массива (индексация элементов начинается с нуля). Решение должно иметь сложность  $O(n)$  в среднем.

Элементы массива  $a_i$  задаются при помощи псевдослучайного генератора по формуле:  $a_i = (1103515245a_{i-1} + 12345) \bmod 2^{31}$ , то есть все элементы массива задаются одним начальным значением  $a_0$ . Для заполнения элементов массива начальными значениями следует использовать следующую функцию:

```
void fill(vector<int> & a, int n, int start)
{
    a.resize(n);
    a[0] = start;
    for (int i = 1; i < n; ++i)
        a[i] = (1103515245LL * a[i - 1] + 12345) % (1LL << 31);
}
```

### Формат входных данных

Программа получает на вход три целых числа  $n$ ,  $a_0$  и  $k$ .

$n$  — количество элементов в массиве,  $1 \leq n \leq 2 \cdot 10^7$ .

$a_0$  — значение первого элемента в массиве,  $0 \leq a_0 < 2^{31}$ .

$k$  — индекс искомого элемента,  $0 \leq k < n$ .

### Формат выходных данных

Программа должна вывести одно целое число —  $k$ -й минимум в данной последовательности.

### Примеры

<code>kth.in</code>	<code>kth.out</code>
5 123456789 2	850994577

### Замечание

В примере из условия массив (до сортировки) имеет вид  $\{123456789, 231794730, 1126946331, 1757975480, 850994577\}$ .

## Задача В. 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 входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 80\,000$ ) — количество элементов массива. Вторая строка содержит  $n$  попарно различных элементов массива  $A$  — целых неотрицательных чисел, не превосходящих  $10^6$ .

### Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

### Примеры

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

## Задача С. Anti-qsort test

Имя входного файла: anti-qsort.in  
Имя выходного файла: anti-qsort.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Рассмотрим алгоритм быстрой сортировки Хоара, с выбором в качестве барьерного элемента среднего элемента на отрезке ( $q = A[(1 + r) / 2]$ ):

```
void qsort(vector<int> & a, int left, int right)
// Сортировка A[left...right] включительно
{
    if (right <= left)
        return;
    int q = A[(1 + r) / 2];
    int i = left;
    int j = right;
    while (i <= j) {
        while (a[i] < q)
            ++i;
        while (q < a[j])
            --j;
        if (i <= j) {
            swap(a[i], a[j]);
            ++i;
            --j;
        }
    }
    qsort(a, left, j);
    qsort(a, i, right);
}
```

По данному числу  $n$  составьте тест, являющийся перестановкой чисел от 1 до  $n$ , на котором этот алгоритм выполняет наибольшее число сравнений (подсчитываются сравнения  $a[i] < q$  и  $q < a[j]$ ).

### Формат входных данных

Программа получает на вход одно целое число  $n$ ,  $1 \leq n \leq 70\,000$ .

### Формат выходных данных

Программа должна вывести перестановку чисел от 1 до  $n$ , на которой данная реализация алгоритма быстрой сортировки Хоара будет выполнять наибольшее число сравнений.

Можно вывести любой из возможных ответов.

### Примеры

anti-qsort.in	anti-qsort.out
3	1 3 2

## Задача D. Свадьбы

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

Одна очень предприимчивая и симпатичная девушка решила собрать себе денег на роскошную жизнь. У нее есть  $N$  поклонников, про каждого из них она узнала размер его состояния  $P_i$ . Девушка намерена выйти замуж и сразу же развестись с некоторыми из своих поклонников. По законам страны в случае развода каждый из супругов получает ровно половину их общего состояния.

Девушка хочет путем заключения браков и разводов получить со своих поклонников как можно больше денег.

### Формат входных данных

Первая строка входных данных содержит число  $N$  — количество поклонников ( $1 \leq N \leq 10^5$ ). Вторая строка содержит  $N$  целых чисел  $P_1, \dots, P_N$  — размеры состояний поклонников ( $0 \leq P_i \leq 10^6$ ). Третья строка содержит одно число  $Y$  — состояние девушки ( $0 \leq Y \leq 10^6 + 1$ ).

### Формат выходных данных

Программа должна вывести одно действительное число — максимальную сумму денег, которая может оказаться у девушки в результате махинаций.

### Примеры

weddings.in	weddings.out
2 5 10 5	7.5
3 1 3 2 0	2.125
1 1 2	2.0

## Задача Е. Коньки

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

В ЛКШ-Зима школьники любят ходить на каток. В прокате коньков есть много коньков самых разных размеров. Школьник может надеть коньки любого размера, который не меньше размера его ноги. Известны размеры всех коньков и размеры ног школьников. Определите, какое наибольшее число школьников сможет одновременно пойти покататься.

### Формат входных данных

Первая строка входных данных содержит число  $N$  — количество коньков в прокате ( $1 \leq N \leq 10^5$ ). Во второй строке записано  $N$  чисел — размеры коньков. В третьей строке содержится число  $M$  — количество школьников в ЛКШ ( $1 \leq M \leq 10^5$ ), четвертая строка содержит размеры их ног. Размеры коньков и ног — натуральные числа, не превосходящие 100.

### Формат выходных данных

Выведите единственное число — наибольшее количество школьников, которое сможет пойти на каток.

### Примеры

<code>skates.in</code>	<code>skates.out</code>
4 41 40 39 42 3 42 41 42	2