

---

## Задача А. Быстрая сортировка

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Вам необходимо реализовать `qsort` и отсортировать с его помощью список чисел.

В этой задаче будет проведена проверка код-стайла. Убедитесь, что у вас стоят пробелы в правильных местах, правильно стоят отступы, все названия переменных несут смысл и имеют больше одной буквы.

### Формат входных данных

В единственной строке входного файла содержится последовательность, содержащая не более чем 100000 целых чисел.

### Формат выходных данных

В единственной строке выходного файла выведите последовательность в неубывающем порядке.

### Пример

стандартный ввод	стандартный вывод
4 1 4 8 6 6 5	1 4 4 5 6 6 8

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

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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.

### Пример

стандартный ввод	стандартный вывод
5 123456789 2	850994577

### Замечание

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

Вам необходимо реализовать поиск  $k$ -ого минимума самому, пользоваться встроенной функцией `nth_element` нельзя.

В этой задаче будет проведена проверка код-стайла. Убедитесь, что у вас стоят пробелы в правильных местах, правильно стоят отступы, все названия переменных несут смысл и имеют больше одной буквы.

## Задача C. Anti-qsort test

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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 ( $q = A[(l + r) / 2]$ ):

```
void qsort(vector<int> & a, int left, int right)
// Сортировка A[left...right] включительно
{
    if (right <= left)
        return;
    int q = A[(l + 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.

### Пример

стандартный ввод	стандартный вывод
3	1 3 2