

## Задача А. Сортировка (!) (1 балл)

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

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания.

### Формат входного файла

В первой строке входного файла содержится число  $n$  ( $1 \leq n \leq 300000$ ) — количество элементов в массиве. Во второй строке находятся  $n$  целых чисел, по модулю не превосходящих  $10^9$ .

### Формат выходного файла

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

### Пример

<code>sort.in</code>	<code>sort.out</code>
10 1 8 2 1 4 7 3 2 3 6	1 1 2 2 3 3 4 6 7 8

### Примечание

Необходимо написать свою сортировку.

## Задача В. Соревнования по бегу (1 балл)

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

В Рио-де-Жанейро перед Олимпиадой-2016 проводятся пробные соревнования по бегу. Одной из важных задач по окончании соревнования является отображение результатов для каждой страны в отдельности. За день до соревнования стало известно, что программное обеспечение для выполнения этой задачи еще не готово. Ваша задача — помочь в его разработке.

Вам дана информация о том, в каком порядке участники приходили к финишу. Про каждого участника известно, какую он представляет страну, а также его фамилия. Составьте для каждой страны, участвовавшей в соревновании, список участников из этой страны в порядке прихода их к финишу.

### Формат входного файла

В первой строке входного файла находится число  $n$  ( $1 \leq n \leq 100\,000$ ) — число участников соревнования. В каждой из последующих  $n$  строк находятся название страны, которую представляет участник, и фамилия участника, разделенные ровно одним пробелом. Первым к финишу пришел участник, приведенный в первой после числа  $n$  строке входного файла, вторым — во второй строке, и так далее. Название страны и фамилия участника — строки длиной от одного до 10 символов, состоящие из заглавных и строчных латинских букв.

### Формат выходного файла

Для каждой страны, участвовавшей в соревновании, выведите результаты соревнования для этой страны в следующем формате. В первой строке выведите три знака равенства, пробел, название страны, пробел и три знака равенства. В последующих строках выведите фамилии участников, представляющих эту страну, в порядке их прихода к финишу, по одной фамилии на строке. Страны следует выводить в алфавитном порядке. При возникновении вопросов к формату выходного файла в первую очередь обращайтесь к примеру выходного файла, приведенному в условии.

### Пример

<code>race.in</code>	<code>race.out</code>
3	=== Russia ===
Russia Ivanov	Ivanov
USA Silver	Petrov
Russia Petrov	=== USA ===
	Silver

### Примечание

Необходимо написать свою сортировку.

## Задача С. Число инверсий (2 балла)

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

*Инверсией* в последовательности чисел  $A$  называется такая ситуация, когда  $i < j$ , а  $A_i > A_j$ .

Дан массив целых чисел. Ваша задача — подсчитать число инверсий в нем.

Подсказка: чтобы сделать это быстрее, можно воспользоваться модификацией сортировки слиянием.

### Формат входного файла

В первой строке входного файла содержится число  $n$  ( $1 \leq n \leq 100000$ ) — количество элементов в массиве. Во второй строке находятся  $n$  целых чисел, по модулю не превосходящих  $10^9$ .

### Формат выходного файла

В выходной файл надо вывести число инверсий в массиве.

### Пример

<code>inversions.in</code>	<code>inversions.out</code>
10 1 8 2 1 4 7 3 2 3 6	17

## Задача D. Анти-QuickSort (2 балла)

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

Для сортировки последовательности чисел широко используется быстрая сортировка — QuickSort. Далее приведена программа, которая сортирует массив **a**, используя этот алгоритм.

```
var a : array [1..N] of integer ;

procedure QSort(left , right : integer);
var i, j, key, buf : integer;
begin
  key := a[(left + right) div 2];
  i := left;
  j := right;
  repeat
    while a[i] < key do        {первый while}
      inc(i);
    while key < a[j] do        {второй while}
      dec(j);
    if i <= j then begin
      buf := a[i];
      a[i] := a[j];
      a[j] := buf;
      inc(i);
      dec(j);
    end;
  until i > j;
  if left < j then QSort(left , j);
  if i < right then QSort(i , right);
end;

begin
  ...
  QSort(1 , N);
end.
```

Хотя QuickSort является самой быстрой сортировкой в среднем, существуют тесты, на которых она работает очень долго. Оценивать время работы алгоритма будем количеством сравнений с элементами массива (то есть суммарным количеством сравнений в первом и втором while). Требуется написать программу, генерирующую тест, на котором быстрая сортировка сделает наибольшее число таких сравнений.

### Формат входного файла

В первой строке находится единственное число  $n$  ( $1 \leq n \leq 70000$ ).

### Формат выходного файла

Вывести перестановку чисел от 1 до  $n$ , на которой быстрая сортировка выполнит максимальное число сравнений. Если таких перестановок несколько, вывести любую из них.

### Пример

antiqs.in	antiqs.out
3	1 3 2

## Задача Е. К-ая порядковая статистика (2 балла)

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

Дан массив из  $n$  элементов. Какое число  $k$ -ое в порядке возрастания в этом массиве?

### Формат входного файла

В первой строке входного файла содержатся два числа  $n$  — размер массива и  $k$ . ( $1 \leq k \leq n \leq 3 \cdot 10^7$ ). Во второй строке находятся числа  $A$ ,  $B$ ,  $C$ ,  $a_1$ ,  $a_2$  по модулю не превосходящие  $10^9$ . Вы должны получить элементы массива начиная с третьего по формуле:  $a_i = A * a_{i-2} + B * a_{i-1} + C$ . Все вычисления должны производиться в 32 битном знаковом типе, переполнения должны игнорироваться.

### Формат выходного файла

Выведите  $k$ -ое в порядке возрастания число в массиве  $a$ .

### Пример

<code>kth.in</code>	<code>kth.out</code>
5 3 2 3 5 1 2	13
5 3 200000 300000 5 1 2	2

Во втором примере элементы массива  $a$  равны: (1, 2, 800005, -516268571, 1331571109).

### Примечание

Нельзя использовать встроенные сортировки и функции нахождения  $k$ -й порядковой статистики.