

## Задача А. Сортировка небольшой последовательности

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

В этой задаче нужно отсортировать числа, заданные во входном файле.

### Формат входного файла

В первой строке входного файла задано целое число  $n$  ( $1 \leq n \leq 5000$ ). Во второй строке заданы через пробел  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

### Формат выходного файла

В первой строке выходного файла выведите через пробел  $n$  чисел — числа  $a_i$  в неубывающем порядке.

### Примеры

<code>bubble.in</code>	<code>bubble.out</code>
3 1 2 3	1 2 3
4 3 2 2 1	1 2 2 3
5 10 100 10 1000 10	10 10 10 100 1000

## Задача В. Медиана последовательности

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

В теории вероятностей и статистике часто рассматриваются конечные последовательности чисел. Иногда для такой последовательности требуется оценить среднее значение ее членов. В разных случаях под средним значением понимают разные числа: мы можем вычислить среднее арифметическое, а можем посчитать, какое число в отсортированной последовательности будет посередине.

*Медианой* последовательности называется число, которое может разбить все числа последовательности на два множества: числа в первом множестве не больше медианы, а во втором — не меньше, и при этом мощности множеств равны (то есть в них одинаковое количество элементов). Например, медиана последовательности  $\{1, 2, 7, 5, 3\}$  — это 3.

Когда в последовательности нечетное количество членов, медиана определяется однозначно — это тот член последовательности, который находится на равном расстоянии от концов последовательности.

В случае, когда в последовательности четное количество членов, медианой могло бы служить любое число между двумя средними значениями в последовательности. Например, в последовательности  $\{2, 2, 3, 10\,000\}$  медианой могло бы быть любое число из интервала  $(2, 3)$ . Для определенности в случае четного количества членов медианой считается среднее арифметическое двух средних значений.

Ваша задача — для данной последовательности целых чисел вывести ее медиану.

### Формат входного файла

В первой строке входного файла задано целое число  $n$  ( $1 \leq n \leq 5000$ ). Во второй строке заданы через пробел  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

### Формат выходного файла

В первой строке выходного файла выведите единственное вещественное число — медиану последовательности с точностью до шести знаков после запятой.

### Примеры

<code>median.in</code>	<code>median.out</code>
5 1 2 5 7 3	3
4 2 2 3 10000	2.5
5 10 100 10 1000 10	10

## Задача С. Устойчивая сортировка пар

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

В этой задаче нужно отсортировать пары чисел, заданных во входном файле. Зададим *устойчивый порядок* на парах следующим образом. Пусть  $(a_i, b_i)$  и  $(a_j, b_j)$  — две пары из исходной последовательности,  $i, j$  — номера этих пар.

Пара  $(a_i, b_i)$  считается меньше пары  $(a_j, b_j)$  в двух случаях:

- если  $a_i < a_j$ ,
- если  $a_i = a_j$  и  $i < j$ .

Таким образом, требуется отсортировать последовательность пар так, чтобы первые числа пар были расположены в неубывающем порядке, и при этом сортировка была устойчивой.

### Формат входного файла

В первой строке входного файла задано целое число  $n$  ( $1 \leq n \leq 5000$ ). В следующих  $n$  строках заданы через пробел по два целых числа,  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq 10^9$ ).

### Формат выходного файла

В первых  $n$  строках выходного файла выведите пары чисел  $a_i b_i$ , разделенных пробелом, в требуемом порядке.

### Примеры

<code>pairs.in</code>	<code>pairs.out</code>
3 1 2 3 1 1 1	1 2 1 1 3 1
4 4 1 3 2 2 2 3 3	2 2 3 2 3 3 4 1

## Задача D. Сортировка различных чисел

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

В этой задаче нужно отсортировать последовательность натуральных чисел, заданных во входном файле, при условии, что числа в последовательности не повторяются, и максимальное из этих чисел не превосходит длины последовательности.

### Формат входного файла

В первой строке входного файла задано натуральное число  $n$  ( $1 \leq n \leq 1\,000\,000$ ). Во второй строке заданы через пробел  $n$  натуральных чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ).

### Формат выходного файла

В первой строке выходного файла выведите через пробел  $n$  чисел — числа  $a_i$  в неубывающем порядке.

### Пример

<code>permut.in</code>	<code>permut.out</code>
2	1 2
2 1	

## Задача E. Сортировка большой последовательности

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

В этой задаче нужно отсортировать числа, заданные во входном файле.

### Формат входного файла

В первой строке входного файла задано целое число  $n$  ( $1 \leq n \leq 300\,000$ ). Во второй строке заданы через пробел  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 1000$ ).

### Формат выходного файла

В первой строке выходного файла выведите через пробел  $n$  чисел — числа  $a_i$  в неубывающем порядке.

### Примеры

<code>sort.in</code>	<code>sort.out</code>
3	1 2 3
1 2 3	
4	1 2 2 3
3 2 2 1	
5	10 10 10 100 1000
10 100 10 1000 10	