

## Задача А. Сортировка пузырьком - 1

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

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания, используя сортировку пузырьком.

### Формат входных данных

Первая строка входного файла содержит целое число  $N$  ( $1 \leq N \leq 1000$ ), вторая строка —  $N$  целых чисел, по модулю не превышающих  $2 \cdot 10^9$ .

### Формат выходных данных

Выведите отсортированный массив

### Примеры

<code>bubble.in</code>	<code>bubble.out</code>
5 9 2 7 1 2	1 2 2 7 9

### Замечание

При решении задачи нельзя использовать встроенные функции `sort` и `sorted`.

## Задача В. Сортировка подсчетом

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

Вам дан массив. Требуется его отсортировать.

### Формат входных данных

В первой строке входного файла находится одно целое число  $n$  — количество элементов массива ( $1 \leq n \leq 200000$ ). Во второй строчке находятся  $n$  натуральных чисел — элементы массива. Все элементы массива не превосходят  $10^4$ .

### Формат выходных данных

В единственную строку выходного файла выведите отсортированный массив.

### Примеры

<code>countsort.in</code>	<code>countsort.out</code>
3 1 2 3	1 2 3
3 3 2 1	1 2 3

## Задача С. Дареный конь

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Никита обожает стабильность. Вчера ночью Гриша подарил ему массив пар целых чисел.

Гриша крайне щедрый молодой человек, поэтому массив оказался несказанно огромным. Никита сделал вид, что доволен подарком, однако, как можно быть довольным неотсортированным массивом?!

Никита хочет отсортировать массив. А вы же помните, что Никита обожает стабильность? Поэтому он хочет, чтобы его массив был стабильно отсортирован по первому числу из пары (Пара  $(a_i, b_i)$  считается меньше пары  $(a_j, b_j)$ , если  $a_i < a_j$  или  $a_i = a_j$  и при этом  $i < j$ ). Но еще больше Никита любит полдник. А еще когда ему делают приятно. Пока он сходит до столовой и обратно, сделайте ему приятно, стабильно отсортируйте его массив.

### Формат входных данных

В первой строке находится одно натуральное число  $n$  — размер подаренного массива ( $1 \leq n \leq 50\,000$ ).

В следующих  $n$  строках заданы по два натуральных числа  $a_i, b_i$  — пары чисел в массиве ( $1 \leq a_i \leq 10, 1 \leq b_i \leq 10\,000$ ).

### Формат выходных данных

Выведите в  $n$  строках стабильно отсортированный массив.

### Примеры

стандартный ввод	стандартный вывод
6	1 6
3 1	2 3
2 3	2 4
1 6	2 2
2 4	3 1
3 3	3 3
2 2	

## Задача D. Хитрая сортировка

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

Дана последовательность чисел. Нужно упорядочить эти числа по возрастанию последней цифры, а при равенстве последних цифр — по возрастанию (точнее, по неубыванию) самих чисел.

### Формат входных данных

Вводится сначала число  $N$  ( $1 \leq N \leq 100$ ), а затем сами числа — натуральные и не превышающие 32000, по одному в каждой строке.

### Формат выходных данных

Выведите последовательность чисел в одной строке, упорядоченную так, как указано в условии.

### Примеры

<code>sort.in</code>	<code>sort.out</code>
3 1280 382 930	930 1280 382

### Замечание

Запрещено использовать встроенные структуры.

## Задача Е. Имперский марш

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

На этот раз Император нагрязнул с ревизией не в какой-то там ангар, а в казармы 501-го легиона имперских штурмовиков. В связи с этим каждого штурмовика постригли «под ежика». Несмотря на развитие нанотехнологий, постригли плохо — в результате из-за различной длины волос штурмовики могут отличаться друг от друга по росту, но незначительно — разница не превышает 137 нанометров. Ваша задача — выстроить штурмовиков по росту.

### Формат входных данных

Первая строка входного файла содержит целое число  $N$  — количество штурмовиков ( $1 \leq N \leq 100\,000$ ), вторая строка  $N$  — натуральных чисел, не превышающих  $2 \cdot 10^9$  каждое — рост штурмовика в нанометрах. Никакие два роста не различаются более, чем на 137 нм.

### Формат выходных данных

Выведите роста штурмовиков в порядке убывания.

### Примеры

<code>march.in</code>	<code>march.out</code>
5 12 1 2 1 13	1 1 2 12 13
1 1000000000	1000000000

### Замечание

При решении этой задачи нельзя пользоваться стандартными функциями и методами `min`, `index`, `sort`, `sorted` и т. д.

Естественно, можно пользоваться функциями `min`, `max`, которые принимают два числа.

## Задача F. Результаты олимпиады

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

$N$  участников олимпиады получили уникальные номера от 1 до  $N$ . В результате решения задач на олимпиаде каждый участник получил некоторое количество баллов (целое число от 0 до 600). Известно, кто сколько баллов набрал.

Требуется перечислить участников олимпиады в порядке невозрастания набранных ими баллов.

### Формат входных данных

В первой строке входного файла содержится число  $N$  ( $1 \leq N \leq 100$ ). Далее записаны через пробел  $N$  чисел — количество набранных участниками баллов (1-е число — это баллы, набранные участником номер 1, 2-е — участником номер 2 и т.д.).

### Формат выходных данных

В выходной файл следует вывести  $N$  чисел — номера участников в порядке невозрастания набранных ими баллов (номера участников, набравших одинаковое количество баллов, выводятся в порядке возрастания).

### Примеры

<code>olymp.in</code>	<code>olymp.out</code>
5 100 312 0 312 500	5 2 4 1 3

## Задача G. Инвентаризация

Имя входного файла: `robots.in`  
Имя выходного файла: `robots.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 128 мегабайт

В связи с модернизацией производства на заводе зубных щеток в Тау Кита было решено переписать список роботов, обслуживающих завод. Каждый робот имеет 2 номера: основной и вспомогательный. Новый список должен удовлетворять следующим правилам:

1. Если один робот в новом списке находится раньше другого, то основной номер первого меньше или равен основному номеру второго.
2. Если основные номера роботов равны, то они расположены в таком же порядке, как и в исходном списке.

Тау Китяне обратились к Вам с просьбой переписать список. Помогите модернизации организаций!

### Формат входных данных

В первой строке входного файла содержится число  $N$  ( $1 \leq N \leq 100\,000$ ) — количество роботов на заводе. На каждой следующей строке находятся 2 числа — основной и вспомогательный номера очередного робота. Оба номера неотрицательны и не превосходят  $10^9$ .

### Формат выходных данных

Выведите  $N$  строчек,  $i$ -ая содержит 2 числа — основной и вспомогательный номер  $i$ -го робота в новом списке.

### Примеры

<code>robots.in</code>	<code>robots.out</code>
10	1 8
1 8	1 11
8 9	2 10
2 10	2 23
1 11	3 11
4 2	3 3
7 2	4 2
3 11	6 7
2 23	7 2
3 3	8 9
6 7	

## Задача N. Число

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрежала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием.

Помогите ему!

### Формат входных данных

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

### Формат выходных данных

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

### Примеры

<code>stdin</code>	<code>stdout</code>
2 20 004 66	66220004
3	3