

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 выбором

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

Отсортируйте массив целых чисел в порядке неубывания, используя сортировку выбором минимума.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 2500$), вторая строка — N целых чисел, по модулю не превышающих $2 \cdot 10^9$.

Формат выходных данных

Данные числа следует вывести в порядке неубывания.

Примеры

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

Задача С. Сортировка вставками

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

Отсортируйте массив целых чисел в порядке невозрастания, **используя сортировку вставками**.

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

Данные числа следует вывести в порядке невозрастания.

Примеры

<code>sort.in</code>	<code>sort.out</code>
5 9 2 7 1 2	9 7 2 2 1
5 5 4 3 2 1	5 4 3 2 1

Замечание

В этой задаче нельзя пользоваться методом `.sort()` и функцией `sorted()`, а также использовать метод `.reverse()`

Задача D. 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 Е. Анаграммы

Имя входного файла: `anagram.in`
Имя выходного файла: `anagram.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Слово называется анаграммой другого слова, если оно может быть получено перестановкой его букв.

Формат входных данных

Даны два слова на отдельных строках. Слова состоят из строчных латинских букв и цифр. Длины слов не превышают 255.

Формат выходных данных

Требуется вывести `YES` — если введенные слова являются анаграммами друг друга, `NO` — если нет.

Примеры

<code>anagram.in</code>	<code>anagram.out</code>
<code>sharm</code> <code>marsh</code>	<code>YES</code>
<code>test</code> <code>set</code>	<code>NO</code>

Задача F. Палиндром

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Палиндром — это строка, которая читается одинаково как справа налево, так и слева направо.

На вход программы поступает набор больших латинских букв (не обязательно различных). Решается переставлять буквы, а также удалять некоторые буквы. Требуется из данных букв по указанным правилам составить палиндром наибольшей длины, а если таких палиндромов несколько, то выбрать первый из них в алфавитном порядке.

Формат входных данных

Входные данные содержат одну непустую строку, состоящую лишь из не более чем 10^5 заглавных латинских символов, без пробелов.

Формат выходных данных

В единственной строке выходных данных выведите искомый палиндром.

Примеры

стандартный ввод	стандартный вывод
AAB	ABA
QAZQAZ	AQZZQA

Задача G. 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	