

## Задача А. Палиндром

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

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

В первой строке входных данных содержится число  $n$  ( $1 \leq n \leq 100000$ ). Во второй строке задается последовательность из  $n$  больших латинских букв (буквы записаны без пробелов).

### Формат выходных данных

В единственной строке выходных данных выведите искомой палиндром.

### Примеры

palindrom.in	palindrom.out
AAB	ABA
QAZQAZ	AQZZQA

## Задача В. Проблема сапожника

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

В некоей воинской части есть сапожник. Рабочий день сапожника длится  $N$  минут. Заведующий складом оценивает работу сапожника по количеству починенной обуви, независимо от того, насколько сложный ремонт требовался в каждом случае. Дано  $k$  сапог, нуждающихся в починке. Определите, какое максимальное количество из них сапожник сможет починить за один рабочий день.

### Формат входных данных

В первой строке вводятся числа  $N$  (натуральное, не превышает 1000) и  $k$  (натуральное, не превышает 500). Затем идет  $k$  чисел — количество минут, которые требуются, чтобы починить  $i$ -й сапог (времена — натуральные числа, не превосходят 100).

### Формат выходных данных

Выведите одно число — максимальное количество сапог, которые можно починить за один рабочий день.

### Примеры

cobbler.in	cobbler.out
10 3 6 2 8	2
3 2 10 20	0

## Задача С. Лишнее число

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

В штаб секретной службы поступило сообщение от одного из агентов. Поступившее сообщение в зашифрованном виде представляет собой последовательность чисел, и лишь специальная программа способна расшифровать его и получить связный текст.

Обычно программа-расшифровщик быстро и бесшумно выдаёт связистам расшифрованный текст, но в этот раз вместо текста от программы поступил сигнал тревоги, свидетельствующий о том, что при пересылке сообщение было взломано или просто повреждено.

Корректное зашифрованное сообщение — это последовательность из  $4 \cdot k$  целых чисел, в котором  $k$  различных чисел присутствуют по 4 раза каждое; для расшифровки даже не важны значения этих чисел, а важен лишь их порядок.

Однако, изучив зашифрованное сообщение, связисты обнаружили, что в нём  $4 \cdot k + 1$  число. При этом ровно одно число является «лишним», то есть при его удалении зашифрованное сообщение становится корректным сообщением из  $4 \cdot k$  чисел (возможно, четыре из них равны удалённому числу).

Связисты решили, что на будущее им нужна программа, которая находит такое «лишнее» число автоматически. Помогите им написать такую программу.

### Формат входных данных

В первой строке входного файла задано число  $N = 4 \cdot k + 1$ , где  $N$  и  $k$  целые, и  $1 \leq k \leq 10\,000$ . В последующих  $N$  строках находятся числа  $A_1, A_2, \dots, A_N$ , по одному числу в каждой — зашифрованное сообщение. Известно, что  $0 \leq A_i \leq 1\,000\,000$ .

### Формат выходных данных

В первую строку выходного файла выведите «лишнее» число из набора  $A_i$ .

### Пример

<code>excess.in</code>	<code>excess.out</code>
5	1
4	
1	
4	
4	
4	

## Задача D. Простая сортировка

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 входных данных

В первой строке входного файла содержится число  $N$  ( $1 \leq N \leq 100\,000$ ) — количество элементов в массиве. Во второй строке находятся  $N$  целых чисел, по модулю не превосходящих  $10^9$ .

### Формат выходных данных

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

### Замечание

Запрещается использовать стандартные сортировки.

## Задача Е. Тестирующая система

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

Юный программист Саша написал свою первую тестирующую систему. Он так обрадовался тому, что она скомпилировалась, что решил пригласить школьных друзей на свой собственный контест.

Но в конце тура выяснилось, что система не умеет сортировать команды в таблице результатов. Помогите Саше реализовать эту сортировку.

Команды упорядочиваются по правилам ACM:

- по количеству решённых задач в порядке убывания;
- при равенстве количества решённых задач — по штрафному времени в порядке возрастания;
- при прочих равных — по номеру команды в порядке возрастания.

### Формат входных данных

Первая строка содержит натуральное число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество команд, участвующих в контесте. В  $i$ -й из следующих  $n$  строк записано количество решённых задач  $S$  ( $0 \leq S \leq 100$ ) и штрафное время  $T$  ( $0 \leq T \leq 100\,000$ ) команды с номером  $i$ .

### Формат выходных данных

В выходной файл выведите  $n$  чисел — номера команд в отсортированном порядке.

### Пример

<code>ejudge.in</code>	<code>ejudge.out</code>
5	5 2 1 3 4
3 50	
5 720	
1 7	
0 0	
8 500	

## Задача F. Пакуем чемоданы!

Имя входного файла: `traveling.in`  
Имя выходного файла: `traveling.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Алёна собирает вещи в отпуск. С собой в самолёт она может взять ручную кладь и багаж. Для ручной клади у Алёны есть рюкзак, а для багажа – огромный чемодан. По правилам перевозки масса ручной клади не должна превосходить  $S$  кг, а багаж может быть любой массы (за сверхнормативный багаж Алёна готова доплатить). Разумеется, наиболее ценные вещи – ноутбук, фотоаппарат, документы и т. д. – Алёна хочет положить в ручную кладь.

Алёна знает для каждого предмета его ценность  $p_i$  и вес  $w_i$ . Она разложила предметы в порядке убывания ценности и начинает складывать наиболее ценные вещи в рюкзак. Она действует следующим образом – берёт самый ценный предмет, и если его масса не превосходит  $S$ , то кладёт его в рюкзак, иначе кладёт его в чемодан. Затем она берёт следующий по ценности предмет, если его можно положить в рюкзак, то есть если его масса вместе с массой уже положенных в рюкзак вещей не превосходит  $S$ , то кладёт его в рюкзак, иначе в чемодан, и таким же образом процесс продолжается для всех предметов в порядке убывания их ценности.

Определите вес рюкзака и чемодана после того, как Алёна сложит все вещи.

### Формат входных данных

Первая строка входных данных содержит число  $S$  — максимально разрешённый вес рюкзака ( $1 \leq S \leq 10^9$ ). Во второй строке входных данных записано число  $N$  — количество предметов ( $1 \leq N \leq 10^5$ ). В следующих  $N$  строках содержатся числа  $p_i$  и  $w_i$  — ценность и вес  $i$ -го предмета соответственно ( $1 \leq p_i, w_i \leq 10^9$ ). Все числа  $p_i$  различны.

### Формат выходных данных

Программа должна вывести два числа – вес рюкзака и вес чемодана (вес пустого рюкзака и чемодана не учитывается).

### Пример

<code>traveling.in</code>	<code>traveling.out</code>
10 3 10 10 20 5 30 7	7 15

## Задача G. Мекс

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

Пусть  $mex$  от последовательности неотрицательных целых чисел — наименьшее число, которое не встречается в этой последовательности. Например,  $mex(0, 1, 3) = 2$ . Найдите  $mex$  от заданной последовательности.

### Формат входных данных

В первой строке дано число  $n$  ( $1 \leq n \leq 10^5$ ) — длина последовательности. Во второй строке дано  $n$  чисел  $a_i$  ( $0 \leq a_i \leq 10^9$ ) — элементы заданной последовательности.

### Формат выходных данных

В единственной строке выведите одно целое число —  $mex$  заданной последовательности.

### Пример

<code>mex.in</code>	<code>mex.out</code>
3 0 1 3	2

## Задача Н. Количество инверсий

Имя входного файла: `inverse.in`  
Имя выходного файла: `inverse.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Напишите программу, которая для заданного массива  $A = \langle a_1, a_2, \dots, a_n \rangle$  находит количество пар  $(i, j)$  таких, что  $i < j$  и  $a_i > a_j$ .

### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 50\,000$ ) — количество элементов массива. Вторая строка содержит  $n$  попарно различных элементов массива  $A$ .

### Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

### Примеры

<code>inverse.in</code>	<code>inverse.out</code>
4 1 2 4 5	0
4 5 4 2 1	6



## Задача I. Выбор заявок

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

Вы прекрасно знаете, что в ЛКШ.Зима 2013 лекции читают лучшие преподаватели мира. К сожалению, лекционных аудиторий у нас не так уж и много, поэтому каждый преподаватель составил список лекций, которые он хочет прочитать ЛКШатам. Чтобы ЛКШата, утром идя на завтрак, увидели расписание лекций, необходимо его составить прямо сейчас. И без вас нам здесь не справиться.

У нас есть список заявок от преподавателей на лекции для одной из аудиторий. Каждая заявка представлена в виде временного интервала  $[s_i, f_i)$  — время начала и конца лекции. Лекция считается открытым интервалом, то есть какая-то лекция может начаться в момент окончания другой, без перерыва. Необходимо выбрать из этих заявок такое подмножество, чтобы суммарно выполнить максимальное количество заявок. Учтите, что одновременно в лекционной аудитории, конечно же, может читаться лишь одна лекция.

### Формат входных данных

В первой строке вводится натуральное число  $N$ , не более 1000 — общее количество заявок на лекции. Затем вводится  $N$  строк с описаниями заявок — по два числа в каждом  $s_i$  и  $f_i$ . Гарантируется, что  $s_i < f_i$ . Время начала и окончания лекции — натуральные числа, не превышают 1440 (в минутах с начала суток).

### Формат выходных данных

Выведите одно число — максимальное количество заявок, которые можно выполнить.

### Примеры

<code>request.in</code>	<code>request.out</code>
1 5 10	1
3 1 5 2 3 3 4	2