

## Задача А. А+В

Имя входного файла: `aplusb.in`  
Имя выходного файла: `aplusb.out`  
Ограничение по времени: 1 second  
Ограничение по памяти: 64 megabytes

Для начала найдите сумму двух целых чисел.

### Формат входного файла

В первой и единственной строке входного файла два числа —  $A$  и  $B$ , по модулю не превышающие  $10^9$ .

### Формат выходного файла

В первой и единственной строке выходного файла одно число, равное сумме  $A$  и  $B$ .

### Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
7 4	11

## Задача В. 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 входного файла

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

### Формат выходного файла

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

## Задача С. Поход в театр

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

Однажды под Новый Год преподаватели решили сводить ЛКШат в театр. Все школьники были поделены ровно на столько групп, сколько есть преподавателей в Школе. После этого было необходимо решить, какая группа с каким преподавателем отправится в театр.

Для каждой группы директор определил *желание* идти в театр, которое представляет из себя целое неотрицательное число и каждую минуту увеличивается на единицу. Аналогично завуч определил для каждого преподавателя *готовность* идти куда-то — так же целое неотрицательное число, увеличивающееся каждую минуту на единицу. Как только появляется группа школьников, желание идти которой в театр не меньше 100, а также преподаватель, готовность которого идти в театр не меньше 200, они вместе уходят. Если к этому моменту уже у нескольких групп школьников желание стало не меньше 100 или у нескольких преподавателей готовность стала не меньше 200, то отправляются в театр в первую очередь те, для которых это событие произошло раньше.

Гарантируется, что изначально желания всех групп различны, как и готовности всех преподавателей.

Узнайте, какая группа с каким преподавателем отправится.

### Формат входного файла

В первой строке содержится одно число  $N$  ( $1 \leq N \leq 1000$ ) — число групп школьников и преподавателей. Во второй и третьей строках содержатся по  $N$  целых чисел — желания групп идти в театр (вторая строка) и готовности преподавателей (третья строка).

### Формат выходного файла

Выведите  $N$  строк, в  $i$ -ой — номер преподавателя, с которым пойдет в театр  $i$ -ая группа школьников. Группы и преподаватели нумеруются с единицы.

### Примеры

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

## Задача D. АСМ Марафон

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

Школьник Вася Иванов так сильно боялся идти на командную олимпиаду, что ему приснился кошмар: В ЛКШ вместо обычной олимпиады устраивали обязательный АСМ марафон. Это почти обычная командная олимпиада, отличается она только продолжительностью. Контест длится ровно 24 часа, то есть если он начался в 00:00:00 то в 23:59:59 команда еще может сдать решение, а в 00:00:00 следующего дня — уже нет.

Как и в обычном турнире АСМ, побеждает команда, решившая наибольшее число задач, а при равном количестве решенных задач лучше результат у той команды, у которой меньше штрафное время. Изначально штрафное время каждой команды равно нулю. За каждую правильно сданную задачу к штрафному времени команды прибавляют время в минутах, округленное вниз, прошедшее с начала соревнования до момента сдачи задачи. Кроме того, если зачтённой попытке предшествовало несколько неудачных попыток сдать ту же задачу, то за каждую из них к штрафному времени прибавляют двадцать минут. За неудачные попытки сдать задачу, которую команде в итоге так и не удалось решить, штрафного времени не начисляется. Так же посылки с результатом "Compilation error" и "Code style violation" не считаются неудачными, то есть за них не начисляются штрафные минуты.

Вам требуется написать программу, которая подсчитает результаты марафона.

### Формат входного файла

В первой строке входного файла находится время начала олимпиады в формате  $hh : mm : ss$ , где двухразрядное целое число  $hh$  ( $00 \leq h \leq 23$ ) означает час, а двухразрядные целые числа  $mm$  и  $ss$  ( $00 \leq mm, ss \leq 59$ ) — минуты и секунды соответственно.

Во второй строке находится единственное целое число  $n$  ( $1 \leq n \leq 1000$ ) — количество посылок за олимпиаду.

Далее следуют  $n$  строк с описаниями посылок. В начале каждой из них в двойных кавычках записано название команды, сделавшей посылку. Название может состоять из строчных и заглавных латинских букв, пробелов и цифр от 1 до 9. Длина названия — не меньше одного символа и не больше 255. После названия команды написано время посылки в том же формате, что и время начала контеста.

Далее через пробел идет заглавная латинская буква - номер задачи. Последние два символа в строке - результат посылки. Результат посылки может быть следующим:

OK — OK  
WA — Wrong answer  
PE — Presentation error  
TL — Time limit  
ML — Memory limit  
CE — Compillation error

CS — Code style violation

### Формат выходного файла

Выходной файл должен содержать итоговую таблицу результатов — по строке на каждую команду. Строки должны идти в порядке уменьшения результата, если у нескольких команд результаты равны, то порядок команд определяется названием — раньше идет та, название которой лексикографически меньше.

Каждая строка должна начинаться с места команды в итоговом зачете. Место команды - это  $k + 1$ , где  $k$  - это число команд, имеющих строго лучший результат. Далее через пробел идет название команды в двойных кавычках, а за ним через пробел два числа - количество решенных задач и штрафное время.

### Примеры

contest.in	contest.out
00:00:00 5 "Super team" 00:00:23 A WA "Mega team" 00:10:21 A WA "Super team" 00:20:23 A OK "Mega team" 00:30:23 A OK "Mega team" 00:40:23 B OK	1 "Mega team" 2 90 2 "Super team" 1 40
01:00:00 3 "Team1" 01:10:00 A WA "Team1" 01:20:00 A OK "Team2" 01:40:00 B OK	1 "Team1" 1 40 1 "Team2" 1 40

## Задача E. Сортировка подсчетом

Имя входного файла: `countsort.in`  
Имя выходного файла: `countsort.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 4 мегабайта

Вам дан массив. Требуется его отсортировать.

### Формат входного файла

В первой строке входного файла находится одно целое число  $n$  — количество элементов массива ( $1 \leq n \leq 2000000$ ). Во второй строчке находятся  $n$  натуральных чисел — элементы массива. Все элементы массива не превосходят  $10^4$ .

### Формат выходного файла

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

**Note**

Используйте силу, силу сортировки подсчетом.