

Задача А. Сортировка дробей

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

В этой задаче вам нужно научиться сортировать не числа, а рациональные дроби.

Программа должна вывести список заданных дробей в порядке неубывания. Если в списке есть две равные дроби $a/b = c/d$, то раньше выводится дробь, у которой меньше числитель.

Формат входных данных

Программа получает на вход n дробей: сначала задается число n ($1 \leq n \leq 100000$), потом идет n строк, в каждой из которых записана одна дробь. Дробь записана в виде a/b ($1 \leq a, b \leq 1000$), где a и b — натуральные.

Формат выходных данных

Выведите список дробей по одной дроби в строчку в том же формате, как и во входном файле.

Примеры

<code>fractions.in</code>	<code>fractions.out</code>
3	2/5
2/5	2/3
2/3	3/3
3/3	

Задача В. Инвентаризация

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

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 С. Рейтинг кавалеров

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

Одна симпатичная и предприимчивая девушка ищет себе идеального партнера для танцевальной зарядки.

Идеальный кавалер по её представлениям должен иметь рост 180 см, поэтому прежде всего она хочет найти юношу, чей рост как можно ближе к 180 см. Будет ли кавалер выше или ниже указанной величины, не имеет значения (то есть юноши с ростом 179 и 181 для неё одинаково привлекательны).

Среди всех кандидатов одинаково подходящего роста ей нужен кто-то, чей вес насколько это возможно близок к 75 кг, но не превышает этой величины. Если же все кандидаты одного роста весят более 75 кг, то девушка выберет самого легкого из них.

Вам дан список кавалеров, содержащий имя партнера, его рост и вес.

Отсортируйте список по критерию привлекательности роста, при равной привлекательности роста — по привлекательности веса. При одинаковых параметрах роста и веса список должен быть отсортирован в лексикографическом порядке имен кавалеров.

Формат входных данных

В первой строке дано одно натуральное число N ($1 \leq N \leq 100\,000$). В последующих строках перечислены кавалеры по одному в строке в следующем формате: в начале идёт имя (строка из заглавных и строчных латинских букв, длиной не более 10), затем через пробел рост кавалера (натуральное число из диапазона $[1; 210]$), затем через пробел — вес (натуральное число из диапазона $[1; 120]$).

Формат выходных данных

Выведите отсортированный список имён кавалеров, по одному в строке.

Примеры

<code>cavaliers.in</code>	<code>cavaliers.out</code>
10	John
George 195 110	Thomas
Thomas 180 75	James
John 180 75	William
James 180 65	Martin
Andrew 165 110	Benjamin
Martin 170 70	Franklin
William 180 77	Theodore
Franklin 195 70	Andrew
Benjamin 165 70	George
Theodore 165 80	

Задача D. k -я порядковая статистика

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

Дан массив, содержащий n целых чисел. Вам нужно найти в этой массиве k -й по счету минимальный элемент ($k = 0..n - 1$), то есть элемент, который после сортировки массива по неубыванию окажется на k -м месте от начала массива (индексация элементов начинается с нуля). Решение должно иметь сложность $O(n)$ в среднем.

Элементы массива a_i задаются при помощи псевдослучайного генератора по формуле: $a_i = (1103515245a_{i-1} + 12345) \bmod 2^{31}$, то есть все элементы массива задаются одним начальным значением a_0 . Для заполнения элементов массива начальными значениями следует использовать следующую функцию:

```
void fill(vector<int> & a, int n, int start)
{
    a.resize(n);
    a[0] = start;
    for (int i = 1; i < n; ++i)
        a[i] = (1103515245LL * a[i - 1] + 12345) % (1LL << 31);
}
```

Формат входных данных

Программа получает на вход три целых числа n , a_0 и k .

n — количество элементов в массиве, $1 \leq n \leq 2 \cdot 10^7$.

a_0 — значение первого элемента в массиве, $0 \leq a_0 < 2^{31}$.

k — индекс искомого элемента, $0 \leq k < n$.

Формат выходных данных

Программа должна вывести одно целое число — k -й минимум в данной последовательности.

Примеры

<code>kth.in</code>	<code>kth.out</code>
5 123456789 2	850994577

Замечание

В примере из условия массив (до сортировки) имеет вид $\{123456789, 231794730, 1126946331, 1757975480, 850994577\}$.

Задача Е. АСМ Марафон

Имя входного файла:	<code>contest.in</code>
Имя выходного файла:	<code>contest.out</code>
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. Марафон 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 послыки 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 входных данных

В первой строке входного файла находится время начала олимпиады в формате $hh : mm : ss$, где двухразрядное целое число hh ($0 \leq hh \leq 23$) означает час, а двухразрядные целые числа mm и ss ($0 \leq mm, ss \leq 59$) — 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 контекста.

Далее через пробел идет заглавная латинская буква — номер задачи. Последние два символа в строке — результат посылки. Результат посылки может быть один из следующих:

OK — OK

WA — Wrong answer

PE — Presentation error

TL — Time limit

ML — Memory limit

CE — Compilation error

CS — Code style violation

Формат выходных данных

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 — 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