

Задача А. Золотой песок

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

Во время ограбления магазина вор обнаружил N ящичков с золотым песком. В ящичек под номером i песок имеет стоимость v_i и вес w_i . Чтобы унести награбленное, вор использует рюкзак. Требуется определить наибольшую суммарную стоимость песка, который может унести грабитель, если грузоподъемность рюкзака ограничена величиной W .

Из ящичков можно пересыпать любое количество песка, тогда отношение стоимости отсыпанного песка к стоимости всего ящичка будет равна отношению объема пересыпанного песка к объему всего ящичка.

Формат входного файла

В первой строке входного файла записаны два числа — N и W ($1 \leq N \leq 1000$, $1 \leq W \leq 10^6$). Далее следует N строк по два целых числа в каждой. В i -ой строке записана стоимость v_i и вес w_i песка в i -ом ящичке. Все числа неотрицательные и не превосходят 10^6 .

Формат выходного файла

Выведите искомую максимальную стоимость с точностью до 3 знаков после запятой.

Пример

dust.in	dust.out
3 50	180.000
60 20	
100 50	
120 30	

Задача В. ЛКОИ

Имя входного файла: lkoi.in
Имя выходного файла: lkoi.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайт

Лкшатская Командная Олимпиада по информатике (ЛКОИ) проходила согласно таким необычным правилам. Было N команд и T задач. Каждая задача

оценивалась с использованием всего лишь одного теста. Таким образом, каждую задачу можно было либо решить полностью, либо не решить совсем.

Количество баллов, назначенное каждой задаче, определялось после соревнования, и было равно количеству команд, которые не решили задачу. Баллы каждой команды подсчитывались как сумма баллов, назначенных задачам, решенным этой командой.

Бурундуки участвовали в соревновании, но они запутались в сложных правилах оценивания, и сейчас они, глядя на результаты, не в состоянии определить свое место в финальном протоколе. Помогите Бурундукам написать программу, которая подсчитает их баллы и место в финальном протоколе.

Перед соревнованием командам присвоили уникальные номера от 1 до N включительно. Номер Бурундуков обозначим P . В финальном протоколе команды перечислены в порядке невозрастания набранных ими баллов. В случае равенства баллов, первыми будут перечислены команды, которые решили больше задач. В случае равенства количества решенных задач, команды с одинаковыми результатами будут перечислены в порядке возрастания их номеров.

Напишите программу, которая по заданной информации о том, какие задачи были решены какими командами, определит количество баллов у Бурундуков и их место в финальном протоколе.

Формат входного файла

Первая строка содержит целые числа N , T и P ($1 \leq N, T \leq 1000$, $1 \leq P \leq N$), разделенные пробелами. Последующие N строк описывают, какие задачи были решены какими участниками. k -ая строка из них описывает, какие задачи были решены участником с номером k . Каждая такая строка содержит T целых чисел, разделенных пробелом. Первое из этих чисел обозначает, решил ли первую задачу участник с номером k . Второе число обозначает то же для второй задачи, и так далее. Эти T чисел могут быть только 0 или 1, где 1 означает, что участник с номером k решил соответствующую задачу, и 0 означает, что он ее не решил.

Формат выходного файла

Ваша программа должна записать в выходной файл одну строку с двумя целыми числами, разделенными одним пробелом. Первое число — количество баллов, которые Бурундуки получили на соревновании ЛКОИ. Второе число — место Бурундуков в финальном протоколе. Место — это целое число от 1 до N включительно, где 1 означает, что участник расположен сверху финального протокола (то есть имеет наибольшее количество баллов), а N означает, что он расположен внизу финального протокола (то есть имеет наименьшее количество баллов).

Пример

lkoi.in	lkoi.out
5 3 2	3 2
0 0 1	
1 1 0	
1 0 0	
1 1 0	
1 1 0	

Задача С. Простая сортировка

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

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания.

Формат входного файла

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 выходного файла

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

Пример

qsort.in	qsort.out
10	1 1 2 2 3 3 4 6 7 8
1 8 2 1 4 7 3 2 3 6	

Задача D. Разброс

Имя входного файла: `diff.in`
Имя выходного файла: `diff.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайт

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания. Разброс между минимальным и максимальным числами в массиве не превышает 107.

Формат входного файла

В первой строке входного файла содержится число N ($1 \leq N \leq 3\,000\,000$) — количество элементов в массиве. Во второй строке находятся N целых чисел, по модулю не превосходящих $2^{31} - 1$.

Формат выходного файла

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

Пример

diff.in	diff.out
3	3 5 8
5 3 8	

Задача E. Точки и отрезки

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

Дано n отрезков на числовой прямой и m точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка x считается принадлежащей отрезку с концами a и b , если выполняется двойное неравенство $\min(a, b) \leq x \leq \max(a, b)$.

Формат входного файла

Первая строка содержит два целых числа n ($1 \leq n \leq 10^5$) — число отрезков и m ($1 \leq m \leq 10^5$) — число точек. В следующих n строках по два целых числа a_i и b_i — координаты концов соответствующего отрезка. В последней строке m целых чисел — координаты точек. Все числа во входном файле не превосходят по модулю 10^9 .

Формат выходного файла

В выходной файл выведите m чисел — для каждой точки количество отрезков, в которых она содержится.

Примеры

segments.in	segments.out
2 2	1 0
0 5	
7 10	
1 6	