

Содержание

Задача А. Простая сортировка [1 sec, 256 mb]	2
Задача В. Экзамен по истории [1 sec, 256 mb]	3
Задача С. incrementator [1 sec, 256 mb]	4

Примеры на питоне:

Сортировки на питоне	sort
Работа со словарями	dictionary
Работа со строками	strings

Задача А. Простая сортировка [1 сек, 256 mb]

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

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

Задача В. Экзамен по истории [1 sec, 256 mb]

Даны два списка дат.

Найти количество дат во втором списке, которые присутствуют в первом.

Формат входных данных

На первой строке длина первого списка N ($1 \leq N \leq 15\,000$).

Следующие N строк содержат целые числа от 1 до 10^9 – даты из первого списка.

Далее длина второго списка M ($1 \leq M \leq 10^6$).

Следующие M строк содержат целые числа от 1 до 10^9 – даты из второго списка.

Формат выходных данных

Одно число – ответ на задачу.

Примеры

history.in	history.out
2 1054 1492 4 1492 65536 1492 100	2

Задача C. incrementator [1 sec, 256 mb]

Ваша задача — написать программу, моделирующую простое устройство, которое умеет прибавлять целые значения к целочисленным переменным.

Формат входных данных

Входной файл состоит из одной или нескольких строк, описывающих операции. Строка состоит из названия переменной и числа, которое к этой переменной надо добавить. Все числа не превосходят 100 по абсолютной величине. Изначально все переменные равны нулю. Названия переменных состоят из не более чем 100 000 маленьких латинских букв. Размер входного файла не превосходит 2 мегабайта.

Формат выходных данных

Для каждой операции выведите на отдельной строке значение соответствующей переменной после выполнения операции.

Примеры

incrementator.in	incrementator.out
a 2	2
b 3	3
a -1	1
c 4	4
b 17	20
xyz 23	23