

Содержание

Задача А. Простая сортировка [5 секунд, 256 МиБ]	2
Задача В. Тестирующая система [5 секунд, 256 МиБ]	3
Задача С. Коллекционер [5 секунд, 256 МиБ]	4
Задача D. Мультиграф [5 секунд, 256 МиБ]	5
Задача Е. Экзамен по истории [5 секунд, 256 МиБ]	6
Задача F. Incrementator [5 секунд, 256 МиБ]	7

Задача А. Простая сортировка [5 секунд, 256 МиБ]

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. Тестирующая система [5 секунд, 256 МиБ]

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

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 конテストе. 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

ejudge.in	ejudge.out
5	5 2 1 3 4
3 50	
5 720	
1 7	
0 0	
8 500	

Задача С. Коллекционер [5 секунд, 256 МиБ]

В Байтландии за всю её историю было выпущено 15000 различных почтовых марок. Известный коллекционер почтовых марок планирует собрать полную коллекцию марок Байтландии. Какое-то количество марок (возможно, с дубликатами) у него есть на данный момент). По заданному списку марок, имеющихся в наличии, вычислить, какое минимальное количество марок коллекционер должен докупить, чтобы коллекция стала полной.

Формат входных данных

Входной файл состоит из двух строк. В первой строке задано одно целое число n ($1 \leq n \leq 3\,000\,000$) — количество имеющихся на данный момент у коллекционера экземпляров байтландских марок. Во второй строке заданы n целых чисел, каждое из которых лежит в интервале $[1, 15000]$ — номер марки, представленной соответствующим экземпляром.

Формат выходных данных

Одно число — минимальное количество марок, которое коллекционер должен докупить, чтобы коллекция стала полной.

Примеры

collection.in	collection.out
8 3 6 2 2 4 6 3 7	14995

Задача D. Мультиграф [5 секунд, 256 МиБ]

Дан неориентированный невзвешенный граф. В графе возможны петли и кратные рёбра. Постройте такой новый граф без петель и кратных рёбер, что для любых двух вершин в нём расстояние равно расстоянию в исходном графе. Если вершины не связны, расстояние между ними бесконечность.

Формат входных данных

На первой строке число вершин n и число рёбер m ($1 \leq n, m \leq 100\,000$). Следующие m строк содержат пары чисел от 1 до n – рёбра графа.

Формат выходных данных

Новый граф в таком же формате. Рёбра можно выводить в произвольном формате.

Примеры

multigraph.in	multigraph.out
3 5	3 3
1 1	1 2
1 3	2 3
2 1	3 1
1 2	
2 3	

Задача Е. Экзамен по истории [5 секунд, 256 МиБ]

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 F. Incrementator [5 секунд, 256 МиБ]

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