

## Задача А. Частотный анализ

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

Вам дан текст. Мы не спрашиваем вас, что хотел сказать автор; в течение отведенного вам времени выпишите все слова из текста в порядке убывания их частоты.

### Формат входных данных

Во входном файле содержится исходный текст. Текст состоит не более чем из 100 000 слов, разделённых пробелами и переводами строк. Все слова состоят из строчных латинских букв. Соседние слова разделены ровно одним пробельным символом. Длина любого слова не превышает 20 символов.

### Формат выходных данных

Выведите все слова, встречающиеся в тексте, по одному на каждую строку. Слова должны быть отсортированы по убыванию их количества в тексте, а при равенстве — по алфавиту.

### Примеры

<code>frequency.in</code>	<code>frequency.out</code>
<code>hi hi what is your name my name is bond james bond my name is damme van damme claudio van damme jean claudio van damme</code>	<code>damme is name van bond claudio hi my james jean what your</code>
<code>oh you touch my tralala mmm my ding ding dong</code>	<code>ding my dong mmm oh touch tralala you</code>

## Задача В. Продажи

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

Дана база данных о продажах некоторого интернет-магазина. Каждая строка входного файла представляет собой запись вида `покупатель товар количество`, где `покупатель` — имя покупателя (строка без пробелов), `товар` — название товара (строка без пробелов), `количество` — количество приобретённых единиц товара.

Создайте список всех покупателей и для каждого покупателя подсчитайте количество приобретённых им единиц каждого вида товаров.

### Формат входных данных

Во входном файле записано не более 100 000 строк в указанном формате. Имена покупателей и названия товаров представляют собой строки из заглавных и строчных латинских букв не длиннее 10 символов. Количество товара в каждой покупке — натуральное число, не превышающее  $10^6$ .

### Формат выходных данных

Выведите список всех покупателей в лексикографическом порядке, после имени каждого покупателя выведите двоеточие, затем выведите список названий всех приобретённых данным покупателем товаров в лексикографическом порядке, после названия каждого товара выведите количество единиц товара, приобретённых данным покупателем. Информация о каждом товаре выводится в отдельной строке.

### Примеры

<code>sales.in</code>	<code>sales.out</code>
<code>Ivanov paper 10</code>	<code>Ivanov:</code>
<code>Petrov pen 5</code>	<code>envelope 5</code>
<code>Ivanov marker 3</code>	<code>marker 3</code>
<code>Ivanov paper 7</code>	<code>paper 17</code>
<code>Petrov envelope 20</code>	<code>Petrov:</code>
<code>Ivanov envelope 5</code>	<code>envelope 20</code>
	<code>pen 5</code>

## Задача С. Англо-латинский словарь

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

Однажды, разбирая старые книги на чердаке, школьник Вася нашёл англо-латинский словарь. Английский он к тому времени знал в совершенстве, и его мечтой было изучить латынь. Поэтому попавшийся словарь был как раз кстати.

К сожалению, для полноценного изучения языка недостаточно только одного словаря: кроме англо-латинского необходим латинско-английский. За неимением лучшего он решил сделать второй словарь из первого.

Как известно, словарь состоит из переводимых слов, к каждому из которых приводится несколько слов-переводов. Для каждого латинского слова, встречающегося где-либо в словаре, Вася предлагает найти все его переводы (то есть все английские слова, для которых наше латинское встречалось в его списке переводов), и считать их и только их переводами этого латинского слова.

Помогите Васе выполнить работу по созданию латинско-английского словаря из англо-латинского.

### Формат входных данных

Во входном файле содержатся несколько описаний английских слов. Каждое описание содержится в отдельной строке, в которой записано сначала английское слово, затем отведёнными пробелами дефис (символ номер 45), затем разделённые запятыми с пробелами переводы этого английского слова на латинский. Переводы отсортированы в лексикографическом порядке. Порядок следования английских слов в словаре также лексикографический.

Все слова состоят только из маленьких латинских букв, длина каждого слова не превосходит 15 символов. Общее количество слов на входе не превышает 100 000.

### Формат выходных данных

Программа должна вывести количество латинских слов в словаре  $k$ . В следующих  $k$  строках программа должна вывести латинско-английский словарь, соответствующий входному словарю, в точности соблюдая формат входных данных. В частности, первым должен идти перевод лексикографически минимального латинского слова, далее — второго в этом порядке и т. д. Внутри перевода английские слова должны быть также отсортированы лексикографически.

### Примеры

dictionary.in	dictionary.out
apple - malum, pomum, popula	7
fruit - baca, bacca, popum	baca - fruit
punishment - malum, multa	bacca - fruit
	malum - apple, punishment
	multa - punishment
	popum - apple
	popula - apple
	popum - fruit

## Задача D. Снеговика

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

Для того, чтобы слепить снеговика, необходимо три снежных кома разного размера. В вашем распоряжении есть  $n$  снежных комов, которые представляют собой шары с радиусами  $r_1, r_2, \dots, r_n$ . Снеговика можно слепить из любых трех комов, радиусы которых попарно различны. Например, из комов с радиусами 1, 2 и 3 можно слепить снеговика, а из комов с радиусами 2, 2, 3 или 2, 2, 2 — нельзя. Определите, какое наибольшее количество снеговиков можно слепить из данных комов.

### Формат входных данных

В первой строке входных данных задано целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество комов. В следующей строке заданы  $n$  целых чисел — радиусы комов  $r_1, r_2, \dots, r_n$  ( $1 \leq r_i \leq 10^9$ ). Радиусы комов могут совпадать.

### Формат выходных данных

В первой строке выведите одно целое число  $k$  — наибольшее количество снеговиков. Следующие  $k$  строк должны содержать описания снеговиков. Описание каждого снеговика должно состоять из трех чисел, разделенных пробелами — радиуса большого кома, радиуса среднего кома и радиуса маленького кома. Снеговиков разрешается выводить в любом порядке. Если решений несколько, выведите любое.

### Примеры

<code>snowmen.in</code>	<code>snowmen.out</code>
7 1 2 3 4 5 6 7	2 7 6 5 4 3 2
3 2 2 3	0

## Задача Е. Огромный граф

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

В этой задаче вам нужно найти кратчайший путь между двумя вершинами в огромном неориентированном невзвешенном графе. Граф настолько огромный, что мы даже не знаем, сколько в нем вершин.

### Формат входных данных

В первой строке входных данных содержится одно целое число — количество ребер в графе  $M$  ( $0 \leq M \leq 10^5$ ). Во второй строке записано два целых числа — номер начальной и номер конечной вершины. В следующих  $M$  строках заданы ребра графа — номера двух вершин, соединенных ребром. Все номера вершин — целые числа от 1 до  $10^9$ .

### Формат выходных данных

Если пути между данными вершинами нет, программа должна вывести число -1. Если путь существует, то программа должна вывести число  $l$  — длину кратчайшего пути между данными вершинами. В следующей строке программа должна вывести  $l + 1$  число — номера вершин этого пути.

### Примеры

hugegraph.in	hugegraph.out
5 1 4 1 3 3 2 2 4 2 1 2 3	2 1 2 4
4 20 30 20 10 20 40 40 30 10 30	2 20 10 30
0 1 1000000000	-1

## Задача F. Минимальное остовное дерево

Имя входного файла: `mst.in`  
Имя выходного файла: `mst.out`  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 256 мегабайт

Дан связный неориентированный взвешенный граф. Необходимо выбрать в этом графе некоторое подмножество рёбер минимального суммарного веса таким образом, чтобы между любыми двумя вершинами графа существовал путь из выбранных рёбер.

Очевидно, что выбранное подмножество рёбер должно быть деревом (для минимальности суммарного веса ребер), и такое подмножество называется *минимальным остовным деревом* или *минимальным каркасом*.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $0 \leq m \leq 100\,000$ ). Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100\,000$ ).

Гарантируется, что данный граф является связным.

### Формат выходных данных

Программа должна вывести одно целое число — вес минимального остовного дерева.

### Примеры

<code>mst.in</code>	<code>mst.out</code>
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7