Имеется файл с данными по успеваемости абитуриентов. Он представляет из себя набор строк, где в каждой строке записана следующая информация:

Фамилия;Оценка\_по\_математике;Оценка\_по\_физике;Оценка\_по\_русскому\_языку

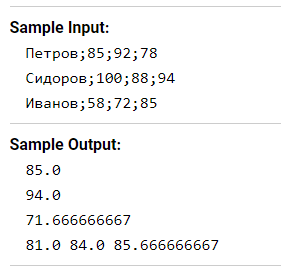
Поля внутри строки разделены точкой с запятой, оценки — целые числа.

Напишите программу, которая считывает файл с подобной структурой и для каждого абитуриента выводит его среднюю оценку по этим трём предметам на отдельной строке, соответствующей этому абитуриенту.

Также в конце файла, на отдельной строке, через пробел запишите средние баллы по математике, физике и русскому языку по всем абитуриентам:

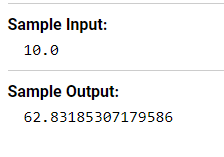
Примечание. Для разбиения строки на части по символу ';' можно использовать метод split следующим образом:





**Модули, подключение модулей**

Напишите программу, которая подключает модуль math и, используя значение числа π из этого модуля, находит для переданного ей на стандартный ввод радиуса круга периметр этого круга и выводит его на стандартный вывод.



Напишите программу, которая запускается из консоли и печатает значения всех переданных аргументов на экран (имя скрипта выводить не нужно). Не изменяйте порядок аргументов при выводе.

Для доступа к аргументам командной строки программы подключите модуль sys и используйте переменную argv из этого модуля.

Пример работы программы:



Напишите программу, которая принимает на стандартный вход список игр футбольных команд с результатом матча и выводит на стандартный вывод сводную таблицу результатов всех матчей.

За победу команде начисляется 3 очка, за поражение — 0, за ничью — 1.

Формат ввода следующий:

В первой строке указано целое число n — количество завершенных игр.

После этого идет n строк, в которых записаны результаты игры в следующем формате:

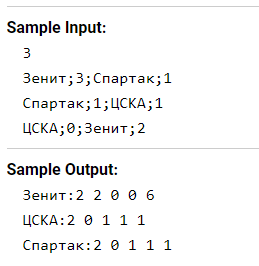
Первая\_команда;Забито\_первой\_командой;Вторая\_команда;Забито\_второй\_командой

Вывод программы необходимо оформить следующим образом:

Команда:Всего\_игр Побед Ничьих Поражений Всего\_очков

Конкретный пример ввода-вывода приведён ниже.

Порядок вывода команд произвольный.



В какой-то момент в Институте биоинформатики биологи перестали понимать, что говорят информатики: они говорили каким-то странным набором звуков.

В какой-то момент один из биологов раскрыл секрет информатиков: они использовали при общении  [подстановочный шифр](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%BF%D0%BE%D0%B4%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B8), т.е. заменяли каждый символ исходного сообщения на соответствующий ему другой символ. Биологи раздобыли ключ к шифру и теперь нуждаются в помощи:

Напишите программу, которая умеет шифровать и расшифровывать шифр подстановки. Программа принимает на вход две строки одинаковой длины, на первой строке записаны символы исходного алфавита, на второй строке — символы конечного алфавита, после чего идёт строка, которую нужно зашифровать переданным ключом, и ещё одна строка, которую нужно расшифровать.

Пусть, например, на вход программе передано:

abcd

\*d%#

abacabadaba

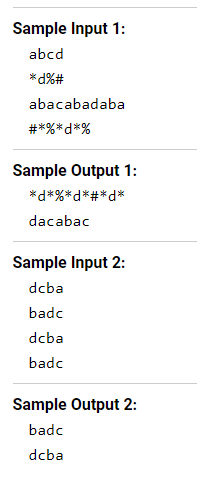
#\*%\*d\*%

Это значит, что символ a исходного сообщения заменяется на символ \* в шифре, b заменяется на d, c — на % и d — на #.

Нужно зашифровать строку abacabadaba и расшифровать строку #\*%\*d\*% с помощью этого шифра. Получаем следующие строки, которые и передаём на вывод программы:

\*d\*%\*d\*#\*d\*

dacabac

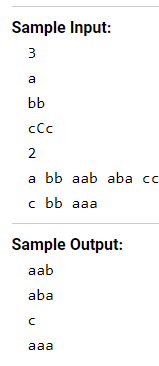


Простейшая система проверки орфографии основана на использовании списка известных слов. Каждое слово в проверяемом тексте ищется в этом списке и, если такое слово не найдено, оно помечается, как ошибочное.

Напишем подобную систему.

Через стандартный ввод подаётся следующая структура: первой строкой — количество d записей в списке известных слов, после передаётся d строк с одним словарным словом на строку, затем — количество l строк текста, после чего — l строк текста.

Напишите программу, которая выводит слова из текста, которые не встречаются в словаре. Регистр слов не учитывается. Порядок вывода слов произвольный. Слова, не встречающиеся в словаре, не должны повторяться в выводе программы.



﻿

Группа биологов в институте биоинформатики завела себе черепашку.

После дрессировки черепашка научилась понимать и запоминать указания биологов следующего вида:

север 10

запад 20

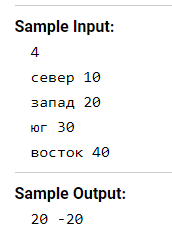
юг 30

восток 40

где первое слово — это направление, в котором должна двигаться черепашка, а число после слова — это положительное расстояние в сантиметрах, которое должна пройти черепашка.

Но команды даются быстро, а черепашка ползёт медленно, и программисты догадались, что можно написать программу, которая определит, куда в итоге биологи приведут черепашку. Для этого программисты просят вас написать программу, которая выведет точку, в которой окажется черепашка после всех команд. Для простоты они решили считать, что движение начинается в точке (0, 0), и движение на восток увеличивает первую координату, а на север — вторую.

Программе подаётся на вход число команд n, которые нужно выполнить черепашке, после чего n строк с самими командами. Вывести нужно два числа в одну строку: первую и вторую координату конечной точки черепашки. Все координаты целочисленные.



Дан файл с таблицей в формате TSV с информацией о росте школьников разных классов.

Напишите программу, которая прочитает этот файл и подсчитает для каждого класса средний рост учащегося.

Файл состоит из набора строк, каждая из которых представляет собой три поля:

Класс Фамилия Рост

Класс обозначается только числом. Буквенные модификаторы не используются. Номер класса может быть от 1 до 11 включительно. В фамилии нет пробелов, а в качестве роста используется натуральное число, но при подсчёте среднего требуется вычислить значение в виде вещественного числа.

Выводить информацию о среднем росте следует в порядке возрастания номера класса (для классов с первого по одиннадцатый). Если про какой-то класс нет информации, необходимо вывести напротив него прочерк, например:

﻿