

Яндекс.Блиц

Квалификационный раунд

24 сен 2017, 16:00:48

старт: 24 сен 2017, 14:19:50

финиш: 24 сен 2017, 18:19:50

до финиша: 02:16:17

начало: 23 сен 2017, 06:00:10

конец: 24 сен 2017, 23:59:10

длительность: 04:00:00

Турнирная таблица

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Многие программисты любят играть в футбол. Некоторые даже любят проводить свои турниры. Но они не хотят следить за тем, кто сколько очков набрал и какое место занял, они хотят просто складывать результаты матчей в базу данных, после чего получать турнирную таблицу с количеством набранных очков и итоговым положением.

Одна из таких групп программистов попросила вас помочь им и, пока они собирают команды и проводят свой турнир, написать программу, которая будет строить итоговую турнирную таблицу.

Формат ввода

Входные данные представляют собой набор строк, каждая строка описывает ровно один сыгранный матч. В каждой строке записаны названия играющих друг с другом команд и результат матча. Названия и результат разделяются знаком тире, отбитым с обеих сторон пробелами. Каждое название состоит только из латинских букв, начинается с заглавной буквы, все остальные буквы строчные, гарантируется что длина каждого названия не превосходит 30 символов. Счет записывается в виде $A:B$, где A — количество голов, забитых первой командой, а B — количество голов, забитых второй командой. Победившей считается команда, забившая больше голов. Если забито одинаковое количество голов, результатом матча считается ничья. За победу команде присуждается три очка, за ничью — одно, за поражение — ноль.

Гарантируется, что нет ни одной пары команд с одинаковыми названиями, что ни одна пара команд не играла между собой более одного раза. Общее число команд-участников не превосходит 100. Ни в одном матче не было забито больше ста голов.

Формат вывода

Вам нужно построить турнирную таблицу с результатами.

Каждая строка таблицы — представление результатов каждой из команд, команды должны быть упорядочены в лексикографическом порядке. В первом столбце содержится порядковый номер команды, во втором — название. Далее следуют n столбцов, в каждом из

которых содержится информация об играх с остальными командами: в случае победы в ячейке должна присутствовать буква W , в случае поражения — L , в случае ничьей — D , если участники не играли друг с другом — пробел, если заполняется ячейка матча игрока с самим собой, то туда следует поставить символ X .

В последних двух столбцах должно быть выписано количество набранных командой очков и итоговое место. Команда A занимает более высокое место, чем команда B , если она набрала большее количество очков, или они обе набрали одинаковое количество очков, но команда A одержала больше побед, чем команда B . Если же число очков и число побед у команд одинаковое, они занимают одно и то же место. Для простоты награждения требуется присудить только места с первого по третье.

Все столбцы должны иметь минимально возможную ширину, чтобы вместить данные в каждой строке. Для столбцов, содержащих порядковые номера, количество набранных очков и занятые места, все данные выравниваются по правому краю, все названия

выравниваются по левому краю, после каждого названия в таблице должен гарантированно присутствовать пробел.

Оформляя таблицу, ориентируйтесь на примеры.

Пример 1

Ввод	Вывод
Linux - Gentoo - 1:0	+--+-----+--+--+--+--+
Gentoo - Windows - 2:1	1 Gentoo X L W 3 1
Linux - Windows - 0:2	+--+-----+--+--+--+--+
	2 Linux W X L 3 1
	+--+-----+--+--+--+--+
	3 Windows L W X 3 1
	+--+-----+--+--+--+--+

Пример 2

Ввод	Вывод
Cplusplus - C - 1:0	+--+-----+--+--+--+--+
Cplusplus - Php - 2:0	1 C X L D 1 3
Java - Php - 1:0	+--+-----+--+--+--+--+
Java - C - 2:2	2 Cplusplus W X W 6 1
Java - Perl - 1:1	+--+-----+--+--+--+--+
Java - Haskell - 1:1	3 Haskell X D 1 3
	+--+-----+--+--+--+--+
	4 Java D D X D W 6 2
	+--+-----+--+--+--+--+
	5 Perl D X 1 3
	+--+-----+--+--+--+--+
	6 Php L L X 0
	+--+-----+--+--+--+--+

Язык

1