

Problem 3 – Handball Statistics

The international handball federation has a job opening at their software development department. They have decided to create a mobile app for their fans and thus are looking for a new developer to join their team. As a huge fan of handball and a great developer, you are extremely motivated to get that spot, because while working for the IHF, you will feel closer to your favorite teams. You have decided to show your skills and motivation by creating a small program, which will help you stand out from the other candidates.

Write a program that aggregates information about **wins** for each team, as well as who their **opponents** were.

You will keep receiving data in the following format until the word 'stop' is received:

```
<team1> | <team2> | <team1 points>:<team2 points> | <team2 points>:<team1 points>
```

Once you have all of the information, you need to start processing it. **Team's names may contain spaces.**

You must store **each team's name** and its **opponents' names**. You must also store the **wins** for every team. The wins are determined by comparing which team has more points, and if both teams have the same amount of points, the overall winner is the one who had more points on away soil. (see example)

Note: There will be no games with a score such as **2:2 | 2:2**, where the winner must be decided by a penalty shootout. All pairs will be unique.

Multithreading: Create a simple producer-consumer setup. Create two threads - one to read the input line by line and to store it in a data structure of your choice, and a second thread that takes the input lines one by one from the same structure and makes the needed calculations.

Input

You will receive several lines in the following format:

```
<team1> | <team2> | <team1 points>:<team2 points> | <team2 points>:<team1 points>
```

- The input ends with the command 'stop'
- There is no invalid input

Output

Print the information for each team in the following format:

```
<team name>  
- Wins: <total wins>
```

- Opponents: <opponent_1, opponent_2, ... opponent_N>

- The **teams** must be ordered by **total wins** in **descending order**. If two teams have the same number of wins, keep in **alphabetical order** by **team name**.
- The **opponents** must be printed in **alphabetical order**.

Constraints

There are no more than 50 lines of input.

The team names will consist of several words containing only English alphabet letters.

The points are integers in range of [0 ... 10]. Examples

Examples

Input	Output
Denmark Belgium 0:0 1:1 Belgium Austria 2:0 0:2 Latvia Monaco 2:0 0:0 Bulgaria Italy 2:1 3:2 stop	Belgium - Wins: 1 - Opponents: Austria, Denmark Bulgaria - Wins: 1 - Opponents: Italy Denmark - Wins: 1 - Opponents: Belgium Latvia - Wins: 1 - Opponents: Monaco Austria - Wins: 0 - Opponents: Belgium Italy - Wins: 0 - Opponents: Bulgaria Monaco - Wins: 0 - Opponents: Latvia

Explanation:

In the first input line '**Denmark | Belgium | 0:0 | 1:1**' Denmark have scored **1 point** in total, Belgium have also scored **1 point** in total. However, Denmark is the **overall winner** because they have scored their point on away soil. This means that **Denmark** has **1 win** and its **opponent** is **Belgium**.

On the second input line '**Belgium | Austria | 2:0 | 0:2**' Belgium has **4 points** and Austria **0 points**. Thus, **Belgium** has

1 win now and its **opponents** are **Denmark** (from the first input line) and **Austria**.

On the third input line 'Latvia | Monaco | 2:0 | 0:0' happens the same as in the second input line.
Input line 4 is similar to the first input.

Scroll down for more examples.

Input	Output
Brazil Germany 1:1 7:0 stop	Germany - Wins: 1 - Opponents: Brazil Brazil - Wins: 0 - Opponents: Germany

Input	Output
Montenegro Cyprus 0:0 1:1 Montenegro Bosnia 0:0 1:1 Montenegro South Africa 0:0 1:1 stop	Montenegro - Wins: 3 - Opponents: Bosnia, Cyprus, South Africa Bosnia - Wins: 0 - Opponents: Montenegro Cyprus - Wins: 0 - Opponents: Montenegro South Africa - Wins: 0 - Opponents: Montenegro

