The European Football Championship is coming! From the 12th of June to the 4th of July, Portugal will be the sports center of the world. Everyone will do their best to make this event a memorable one.

However, an important detail is missing. The rules for the classification of the league stage have changed and everyone is a little bit confused. What would be really, really nice would be a computer program to calculate this classification, given the results of the game. Then it would also be possible to watch in real time the changes in the ranking!

Here is how the classification is made:

1. For a win 3 points will be awarded; for a tie, 1 point; for a defeat, 0 points

2. For establishing the final places, the following criteria will be applied, in descending order of priority:

   - Number of points
   - Goal-Average (difference between goals scored and given)
   - Number of wins (victories)
   - Number of goals scored

So after the games are taken in account, this parameters are all calculated. Things get interesting when there is more than one team with the same number of points. In that case, a sub-league is considered. You must now imagine that only the games between the tied teams count, and see the new sub-classification. If that does not break the tie (in points) for all the teams, you must do a sub-sub-league for all the teams that are still tied, and so on. There is only one case when a sub-league should not be partitioned. That is when all the teams in that sub-league have the same number of points and obviously, the partition would give the exact same group of teams and parameters. In that cases, teams should be ranked according to the four parameters calculated for that sub-league. If the parameters are not sufficient, then the teams should be considered to be in the same place, and they should appear in alphabetical order.

Your task is to write a program that given the results of several games, calculates the classification of the teams using the sorting algorithm defined above.

Of course that knowing how good programmer you are, the organization has asked you to make a program that could calculate the classification of thousands of teams more than the ones that will be present in Euro'2004, in order to use the program in any situation they want.

Input

The first line of input contains an integer T which is the number of test cases that follow.

Each test case starts with a number G ($1 \leq G \leq 10000$) indicating the number of games to consider.

Then G lines follow, each one with the format "$TEAM1\ TEAM2\ GOALS1\ GOALS2$", giving the result of a single game ($TEAM1$ scored $GOALS1$ goals and $TEAM2$ scored $GOALS2$ goals). Team names are made only by lower-case letters and have a maximum length of 20.

It is not necessary that games between all the teams have been made. Of course that you should only calculate the classification based on the games that you were given. Also, some teams may play against each other more than one time.

To make the classification you should consider all the teams that played at least one game. You may assume that the number of teams is ≤ 10000

Output

The output for each test case consists of lines in the form "$PLACE\ TEAM$", in ascending order of place, where $PLACE$ indicates the place the team got and $TEAM$ is the name of the team. Remember that all teams that played at least one game must appear.

Output of different test cases should be separated by a single blank line.

See the example output for a more detailed explanation of how the classification was obtained on that particular cases.

Explanation of Sample I/O:

- First sample case:

  Looking at the games, we see that "`portugal`", "`espanha`" and "`grecia`" made 6 points, and "`russia`" made 0 points (which automatically gives them the 4th place). A tie between the first three teams is achieved. A sub-league with only that three teams is then considered but in this sub-league all the three teams have 3 points. This tied group cannot be partitioned further and then the other parameters are considered. Since in that sub-league, "`portugal`" has the best goal-average, it achieves 1st place. Then comes "`espanha`" (2nd goal-average) and finally "`grecia`". If necessary, the other parameters would have been taken in account.

- Second sample case:

  Now, "`portugal`" and "`grecia`" are tied with 6 points, and "`espanha`" and "`russia`" have 3 points. The sub-league between "`portugal`" and "`grecia`" unties the two teams ("`portugal`" won against "`grecia`"), and in the same way the sub-league between "`espanha`" and "`russia`" unties them.

- Third sample case:

  The only two teams tied the game, so they are equal in all parameters. They are in the same place and they appear in alphabetical order.

Sample Input

```
3
6
portugal grecia 4 1
espanha russia 3 1
portugal russia 3 0
espanha grecia 1 2
portugal espanha 1 3
grecia russia 7 0
6
portugal grecia 4 1
espanha russia 1 3
portugal russia 3 0
espanha grecia 1 2
portugal espanha 1 3
grecia russia 7 0
1
brasil franca 0 0
```

Sample Output

```
1 portugal
2 espanha
3 grecia
4 russia

1 portugal
2 grecia
3 russia
4 espanha

1 brasil
1 franca
```