Soccer Tournament

`soccer.c`, `soccer.C`, `soccer.p`, `soccer.java`

ACM decided to run a big advertising campaign. Beside others, it will sponsor soccer tournaments by providing them with a system for computing and displaying game results. You are to write that system.

Input Specification

The input contains description of several tournaments. Each tournament begins with a line containing single positive integer N: the number of participating teams, $N \leq 80$. Then there are N lines, each containing a unique name of one team. The name is case-sensitive and may be composed only from letters, digits, dots ("`.`"), and dashes ("`-`"). No name will be longer than 100 characters.

After the team names, there will be a line with a single non-negative integer M: the number of games that have already been played. Each of the next M lines describes one game and contains the host team name, space, dash, space, guest team name, space, and the game result. The result will always appear as a single digit (0-9), colon ("`:`"), and another single digit (0-9). The digits specify the number of goals scored by the host and guest teams.

Then, the next tournament is described. The last tournament is followed by a zero on a separate line. No output should be produced for this zero.

You may assume that all team names have been listed among the N teams. All games will have distinct opponent pairs. This also means that any two teams may play at most two games with each other — in such a case, each of the two teams will be a host in one game and a guest in the other.

Output Specification

For each tournament, output a line containing the word "`RESULTS`" and a colon ("`:`"), then a result table, empty line, a line with the word "`STANDINGS`" and a colon, a line with 10 dashes ("`-`"), then current standings, and another empty line.

The result table is "drawn" using dashes ("`-`"), pipes ("`|`"), and plus signs ("`+`"). The table has $N + 1$ rows and $N + 1$ columns. The upper left cell is always empty (filled with spaces). Otherwise, the first column contains full team names, the width of the column must exactly fit the longest name. The first row contains first three characters of team names (followed by one or two spaces if the name was shorter than 3 characters). All teams must be given in the same order as they appeared in the input list of teams.

Diagonal cells (except the first row and column) contain centered uppercase letter "`X`". All other cells are either empty (three spaces) or contain game results for the games that have already been played. Hosting team is in the row, guest team in the column.

The standings consist of N output lines, each of them representing one team. The lines form 8 aligned columns separated with a single space character. The columns must be perfectly aligned, which means that the column occupies the same number of characters in each row. The number of characters must be equal to the longest data in that column. Other rows will be padded with spaces on the left (columns are right-aligned). The only exception is the second column (team names), which is padded from right (left-aligned).

The contents of individual columns are as follows:

1. Rank of the team (number of the line, starting with one) and a dot ("`.`").
2. Team name.
3. Number of games played by the team.
4. Number of games won (the team scored more goals than the opponent).
5. Number of tied games (both teams scored the same number of goals).
6. Number of games lost (the other team scored more goals).
7. Score consisting of three parts: number of goals scored by the team in all its games, colon ("`:`"), and number of goals scored by opponents in all games of the team.
8. Number of points achieved. Each team gets one point for a tie and three points for a victory. There are no points for lost games.

The standings must be sorted by the following criteria, in the given order:

1. Points, the team with more points is listed first.
2. Score difference (number of goals scored minus the number of goals scored by opponents), the team with a higher difference is given first.
3. Number of goals scored by the team, the one that scored more is listed first.
4. Number of games won. The team with more victories will be first.
5. *Reversed* order in the input. The team that was given later in the input list will be ranked better.

Refer to the sample output below whenever the format is ambiguous or unclear.

Sample Input

```
4
Slavia
Arsenal
Steaua
FCSevilla
4
Slavia - Arsenal 2:0
FCSevilla - Steaua 1:1
Steaua - Slavia 1:2
Arsenal - Slavia 0:0
0
```

Output for Sample Input

```
RESULTS:
+---------+---+---+---+---+
|         |Sla|Ars|Ste|FCS|
+---------+---+---+---+---+
|Slavia   | X |2:0|   |   |
+---------+---+---+---+---+
|Arsenal  |0:0| X |   |   |
+---------+---+---+---+---+
|Steaua   |1:2|   | X |   |
+---------+---+---+---+---+
|FCSevilla|   |   |1:1| X |
+---------+---+---+---+---+

STANDINGS:
----------
1. Slavia    3 2 1 0 4:1 7
2. FCSevilla 1 0 1 0 1:1 1
3. Steaua    2 0 1 1 2:3 1
4. Arsenal   2 0 1 1 0:2 1
```