CS201 - Spring 2018 - Sabancı University
Homework #7 – Due: May 18, Friday, 23:55 (Sharp Deadline)

FOOTBALL LEAGUE POINT TABLE GENERATOR

Description

In this homework, you will write a program to generate a football league point table. A point table consists of information about performance status of football teams. Your program will read results of matches among various teams from an input file. Moreover, for each team, your program will compute the number of wins, losses, draws, games played, goals scored, goals conceded and more importantly points, goal difference, and the position. Then you will ask a team name from the user, and display related information regarding this team on the console.

Input File

Your program should read match results from an input file. In this file, any match result will be represented in one line. The format of a match result is as the following.

team1name score1 score2 team2name

team1name: the name of the first team
score1: the number of goals scored by the first team
score2: the number of goals scored by the second team
team2name: the name of the second team

For example:
TOTTENHAM_HOTSPUR 4 1 WEST_HAM_UNITED

As it said above, there is only one match result on each line of the input file. There might be one or more blanks or tab characters before/after the team names and the scores.

You may assume that there is not any empty line in the input file and the score values are always integers. Thus, you don't need to perform an input check to make sure that score values are integers. One other assumption is that each team name consists of only one string. A name of a team may only contain uppercase letters and/or underscore character '_'. You do not have to check whether a team name is a valid, since they are already valid in the input file.

However, you cannot make any assumption about the number of teams and the number of matches in the input file.

Processing

When you start the program, the name of the input file will be entered by the user through the console as the first input of the program. If the user enters a wrong file name that does not exist, your program should display an appropriate message and ask for the name of a new input file from the user until a correct and existing file name is entered.

Use a `struct` data structure to encapsulate one team data, like team name, team points, number of wins, etc. It is up to you to decide which fields are to be included in this `struct`. Note that the number of fields may vary depending on your algorithm.

You can use a `vector` to store several teams' data. Of course, the type of this vector will be the previously defined team struct. Here please remark that you CANNOT make any assumption about the number of teams in the league. The input file in this zip package is retrieved from Barclays Premier League in 2015; so there are 20 teams. However, your program should work for different leagues which contain different number of teams.

Probably most of you already know, but for those of you, who have no idea about football match results, let us give some basic information:

- In a match, if one team scores more goals than the other team, the team that scores more goals wins the game. The winner gets 3 points, the loser gets nothing.
- If the goals scored in a match are the same, then a draw occurs. In this case, both teams get 1 point.

Flow of the program may be as follows:

- Your program may read the input file line by line. The line format is described above.
- For each line:
  - Parse that line and decide if the teams exist in the vector by searching it.
  - If a team exists, your program should modify that team's entry in the vector according to the new match result.
  - If it is a new team, your program should create a new instance of the struct you declared and add it to the vector (probably to the end of the vector).
- After your program finishes reading the match results from the input file and processing them, it should sort the vector in ascending manner.
  - You may modify and use one of sorting algorithm provided by the book and/or discussed in the lecture. If you want to develop your own sorting algorithm, of course you may, but this will be more difficult.
  - You should sort the vector by *points*; teams with higher points must be at higher positions in the table. If multiple teams have the same *point*, then you should use *goal difference* (total goals scored by the team - total goals conceded by the team) values; in this case the team with higher *goal difference* value must be at a higher position. If multiple teams have both the same *point* and *goal difference* values, then they should be sorted by their name; teams with alphabetically smaller names (in string comparison terms) must be at a higher position in the table.

Output

After the teams are sorted, your program should ask for a team name. Then the following information should be printed on different lines:

- the rank of the team
- the number of matches played by the team
- the number of wins by the team
- the number of draws by the team
- the number of losts by the team
- the number of goals scored by the team
- the number of goals conceded by the team
- the goal difference for the team
- the number of points collected by the team

A sample of the output as follows:

Rank: 3
Matches played: 14
Wins: 8
Draws: 4
Losts: 2
Goals scored: 20
Goals conceded: 10
Goal difference: 10
Points: 28

If there is no such team in the input file, then you should print:

"There is no such team!"

However, the output format is very important. You should strictly follow the format of the output as in the sample runs. Please check the sample runs carefully.

In the zip package of this homework, we provide a sample input file: *input.txt*. You may examine this sample to understand the homework and output format in detail.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. You should follow the input order in these examples and the prompts your program will display must be **exactly the same** as in the following examples.

Sample Run 1

```
Please enter a filename: input.txt
Please enter a team name: MANCHESTER_UNITED
Rank: 3
Matches played: 14
Wins: 8
Draws: 4
Losts: 2
Goals scored: 20
Goals conceded: 10
Goal difference: 10
Points: 28
Press any key to continue . . .
```

Sample Run 2

```
Please enter a filename: input.tx
Program cannot open input.tx
Please enter a filename: input
Program cannot open input
Please enter a filename: input.txt
Please enter a team name: WEST_HAM_UNITED
Rank: 8
Matches played: 14
Wins: 6
Draws: 4
Losts: 4
Goals scored: 25
Goals conceded: 21
Goal difference: 4
Points: 22
Press any key to continue . . .
```

Sample Run 3

```
Please enter a filename: input.txt
Please enter a team name: WATFORD
Rank: 12
Matches played: 14
Wins: 5
Draws: 4
Losts: 5
Goals scored: 15
Goals conceded: 18
Goal difference: -3
Points: 19
Press any key to continue . . .
```

Sample Run 4

```
Please enter a filename: input.txt
Please enter a team name: ARSENL
There is no such team!
Press any key to continue . . .
```

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all homeworks, unless otherwise noted.

How to get help?

You may ask questions to TAs (Teaching Assistants) of CS201. Office hours of TAs are at the class website. Recitations will partially be dedicated to clarify the issues related to homework, so it is to your benefit to attend recitations.

What and Where to Submit

Please see the detailed instructions below/in the next page. The submission steps will get natural/easy for later homeworks.

Grading and Objections

Careful about the full-automatic grading: Your programs will be graded using an automated system. Therefore you should follow the guidelines about input and output order; moreover you should also use same prompts as given in the Sample Runs. Otherwise automated grading process will fail for your homework, and you may get a zero, or in the best scenario you will lose points.

Grading:

- There is no late submission!

- **Having a correct program is necessary, but not sufficient to get the full grade. Comments, indentation, meaningful and understandable identifier names, informative introduction and prompts, and especially proper use of required functions, unnecessarily long program (which is bad) and unnecessary code duplications (which is also bad) will also affect your grade.**
- Please submit your own work only (even if it is not working). It is really easy to find out "similar" programs!
- For detailed rules and course policy on plagiarism, please check out http://myweb.sabanciuniv.edu/gulsend/su_current_courses/cs-201-spring-2008/plagiarism/ and keep in mind that

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your homework from the email address provided in the announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- Download the files you submitted to SUCourse and try to compile it.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

What and where to submit (IMPORTANT)

Submissions guidelines are below. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Add your name to the program: It is a good practice to write your name and last name somewhere in the beginning program (as a comment line of course). Do not use Turkish characters anywhere in your program (not even in the comments).

Name your submission file:

- Use only English alphabet lowercase letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
- Name your cpp file that contains your program as follows.

  "**sucourseusername_lastname_name_hwnumber.cpp**"
- Your SUCourse user name is actually your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:

  cago_ozbugsizkodyazaroglu_caglayan_hw7.cpp

- Do not add any other character or phrase to the file name.
- Make sure that this file is the latest version of your homework program.
- You have to submit all files that are used in your homework (ago_ozbugsizkodyazaroglu_caglayan_hw7.cpp, strutils.h, strutils.cpp, and all txt files) **without zipping them**.

Submission:

- Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).
    1) Click on "Assignments" at CS201 SUCourse (not the CS201 web site).
    2) Click Homework 7 in the assignments list.
    3) Click on "Add Attachments" button.
    4) Click on "Browse" button and select all file used in homework.
    5) Now, you have to see your file(s) in the "Items to attach" list.
    6) Click on "Continue" button.
    7) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Resubmission:

- After submission, you will be able to take your homework back and resubmit. In order to resubmit, follow the following steps.
    1) Click on "Assignments" at CS201 SUCourse.
    2) Click Homework 7 in the assignments list.
    3) Click on "Re-submit" button.
    4) Click on "Add/remove Attachments" button
    5) Remove the existing files by clicking on "remove" link. This step is very important. If you do not delete the old files, we receive both files and the old one may be graded.
    6) Click on "Browse" button and select the new files that you want to resubmit.
    7) Now, you have to see your new files file in the "Items to attach" list.
    8) Click on "Continue" button.
    9) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!
İnanç Arın and Gülşen Demiröz