

Background:

An extremely common activity for Computer Scientists is to store, sort and print data. That's what you will be doing in this project. You will simply use a struct to store information about 25 lacrosse games. The information you will read from the file will be the names of the two teams and the final score of the game. You will need to compute the margin of victory.

Details:

You will read all the game information from a file called, LacrosseInfo.txt. You will write the output to a file called, LacrosseWin.txt. The input file will contain information about 25 games; which includes the two teams and the final score. Each item will be separated by a tab and each line, including the last one, will be terminated with a newline character. You will compute the margin of victory and store the information in an array of structs. Once you have finished reading in the data, you will print the array of data. After that you will sort the array in ascending order by margin of victory. Finally you will print the array of data again. All that is needed in the printout is the two teams and the margin of victory. You will also print a header for the table and one blank line between the tables. See the webpage soon for a sample input and output.

Implementation Details

- You must implement an array of structs for this project.
- You may use an array of size 25.
- The only fields that have to be store are the names of the teams and the margin of victory.
- If you want you may store other information as well.
- The sort and print operations must be implemented as a function.
- You must not have any global variables; therefore you will have to pass your array of structs into your functions.

Submitting your project

You will submit this assignment to the Curator System (read the Student Guide), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the Student Guide.

You will be allowed up to five submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted.

The Student Guide and submission link can be found at:

<http://www.cs.vt.edu/curator/>

Pledge

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you must include the following pledge statement in the header comment for your program:

```
// On my honor:  
//  
// - I have not discussed the C++ language code in my program with  
//   anyone other than my instructor or the teaching assistants  
//   assigned to this course.  
//  
// - I have not used C++ language code obtained from another student,  
//   or any other unauthorized source, either modified or unmodified.  
//  
// - If any C++ language code or documentation used in my program  
//   was obtained from another source, such as a text book or course  
//   notes, that has been clearly noted with a proper citation in  
//   the comments of my program.  
//  
// - I have not designed this program in such a way as to defeat or  
//   interfere with the normal operation of the Curator System.  
//  
// <Student Name>
```

Failure to include this pledge in a submission is a violation of the Honor Code.