

**Problem Statement:**

For this problem, you are to solve the same assignment given in Program 1, using the programming language Julia.

**Additional Requirements:**

Version 1 of this program (Go) required the report to be ordered by the player's names in alphabetical order. For this version of the program you write in Julia, print two versions of the report to the screen.

The first part of the report should print the results ascending by lastname (use first names to break ties when applicable, and your sort should be case insensitive), the same as in Program #1. The second portion should print the same table of results, except this time, ordered decreasing by passer rating.

You may assume all of the input in the test data file is valid and there are no blank lines in the data file.

**Libraries:**

Some libraries you may import/use include **Printf**, **FileIO**, **Statistics**. Please check with me before using other advanced library tools. I prefer you use the raw language, but Printf provides C style formatting, and FileIO may be useful, but is not always necessary.

Julia Download is available at: <https://julialang.org/>

Jetbrains tools allow you to install a Julia plugin to compile and run programs in the IDE. I believe I have installed it in my Goland IDE. I believe VScode also has a Julia plug-in.

**TURN IN:**

Please organize your program into a single Julia file to keep it simple for the grader to test.

Submit the electronic version of your project on CANVAS. Make sure your name and the name of the file is in the comments at the top of any program file, along with the system (windows 10, linux, etc.) you tested your program on. You should be using the current version available at the language download site.

See grading rubric on final page.

*Julia is an interesting language that was developed for the math domain, among other uses. That means it has some interesting array, matrix and other data manipulation. While you won't necessarily need the advanced features of Julia, try to explore the language to get a feel for what it can be used for. I found it much easier to organize my player data as a matrix where each row in the matrix is essentially an array containing the player's fields, instead of creating a player record like I did in the Go version. This exploits Julia's matrix manipulation strengths.*

## General Program Grading Rubric (out of 40 points)

I provide this as a general guideline for how each area of a program is to be rated for grading purposes.

Performance Element	5 - Excellent	4 – Very good	3 - Good	2 - Limited	1 - Inadequate
Specifications <b>(20 points)</b>	Program runs & meets all specifications	Runs, gets correct answers, output displayed correctly, meets most other specifications	Runs, gets correct answers, output is not displayed correctly or other specifications not met	Runs, get some correct results, does not meet most specifications	Program does not run, runs but gets no results or mostly wrong results
Readability <b>(5 points)</b>	Code is well organized and easy to follow	Most of the code is well organized & easy to read	Parts of the code are easy to read but the organization is not good	Code is readable if the reader knows what it is supposed to be doing	Code is poorly organized and very difficult to read
Documentation (i.e. comments) <b>(5 points)</b>	Documentation is clearly written & explains what the program as a whole should do; comment blocks introduce each function	Documentation is brief but helpful, includes header information for functions	Documentation consists of embedded comments and some header information for functions	Little or no documentation other than obvious embedded comments	Little or no documentation
Julia Features <b>(10 points)</b>	Program uses data organization and logic features supported by the Julia language.		Minor Usage of Julia features that are different from traditional programming languages.		Little or no usage of Julia specific features.