COP3502C Computer Science 1
Dr. Andrew Steinberg
Spring 2022

Programming Assignment 5

The Unfair Game

Max Points 100

Due: 4/22/2022 at 11:59pm

Background Story of this Assignment

You and 1999 friends are playing this new strategic battle game. The game requires two teams only. This means you and 999 people (1000 total) will be on one team together. In order to get the game started, teams need to be formed. The group votes you and Ron to be team captains. Ron is a brand new person who just moved into town and doesn't know anyone or anything about them (including strengths and weaknesses). You however know everyone (including strengths and weaknesses)! You plan to use this to your advantage to pick the best players. Your objective is to pick the best players for the team so you can beat Ron.

Have fun and start early (seriously start early)! Make sure to see the TAs and ULAs for help EARLY! DO NOT PROCASTINATE!

Assignment Details

In this assignment you are going to create the best team and make the game unfair. In order to find out if you made the game truly unfair, take the average ranking of both teams. The results should be completely different and not even close. Dr. Steinberg has provided a test script for a scenario to test out. Each player has a name (only first name) and an associated rank that determines their capabilities. The rank ranges from 1 to 5 (1 being the best and 5 being the worst). Each player can be represented as a typedef structure as follows.

```c
typedef struct{
	char * name; //dynamic string
	int rank;
}player_t;
```

1. The name component is a dynamic string that represents the first name of the player.
2. The rank component represents the rank (which can be 1, 2, 3, 4, or 5).

The Provided Skeleton File

For this assignment you were provided with a skeleton (program5_Skeleton.c) file that contains the following content. This section discusses the content of the provided skeleton file that you may not be familiar with.

Line 5 defines the macro constant representing the number of players total.

Line 13 shows a function prototype for an implemented function that reads player names from a text file and assigns them a rank.

Lines 19-39 shows the main function. Inside the main function 3 dynamic arrays are created. One will contain the roster of ALL players and the other two are the teams.

Lines 44-62 shows the user defined function scanRoster that collects the name and assigns the ranking of the player randomly. Do not change this code!

The Function Prototypes

For this assignment, the function prototypes are not provided for you. At this point in the course you should have the necessary skills to apply function decomposition. Do not write your entire solution in the main function or one/two user defined function(s). Use good practice!

Requirements

Your program must follow these requirements.

- The output must match exactly (this includes case sensitivity, white space, and even new lines). Any differences in the output will cause the grader script to say the output is not correct. Test with the script provided in order to receive potential full credit. Points will be deducted!
- Do not remove ANY content of the skeleton that was provided for you. Your code will be tested through a script that relies on this main function. Any changes to this will result in your program not working fully which will lead to point deductions that will not be fixed! You will need to add to the skeleton file to get it to work.
- Name your C file `program5_lastname_firstname.c` where `lastname` and `firstname` is your last and first name respectively. Please make sure it matches the spelling exactly how it is registered in Webcourses. Points will be deducted if the file is not named correctly.
- **You will need to make sure there are no memory leaks!**
- Do not change the typedef structure that was provided for you.
- **Your solution must run in $O(nlogn)$ time where n is the number of players. If the solution doesn't satisfy this requirement, points will be deducted.**
- Use function decomposition! If you don't use function decomposition, then points will be deducted.

Tips in Being Successful

Here are some tips and tricks that will help you with this assignment and make the experience enjoyable.

- Do not try to write out all the code and build it at the end to find syntax errors. For each new line of code written (my rule of thumb is 2-3 lines), build it to see if it compiles successfully. **It will go a long way!**
- After any successful build, run the code to see what happens and what current state you are at with the program writing so you know what to do next! If the program performs what you expected, you can then move onto the next step of the code writing. If you try to write everything at once and build it successfully to find out it doesn't work properly, you will get frustrated trying find out the logical error in your code! **Remember, logical errors are the hardest to fix and identify in a program!**
- Start the assignment early! Do not wait last minute (the day of) to begin the assignment.
- Ask questions! It's ok to ask questions. If there are any clarifications needed, please ask TAs/ULAs and the Instructor! We are here to help!!! You can also utilize the discussion board on

Webcourses to share a general question about the program as long as it doesn't violate the academic dishonesty policy.