Homework 9

Released Friday

Due Friday 11:59pm

Goals:

- Memory allocation:
    - Using malloc() and free() appropriately

Notes:

You will be implementing malloc() and free() in a linked list. Here are some rules of thumb for this assignment:

1. For every malloc, there should always be one free.
2. For every fopen, there should always be one fclose.
3. Freeing a pointer does not set the pointer to null, it has to be done manually.
4. GDB and Valgrind is your friend.

Task: Best Basketball Team

In this task, you will implement a program that aims to builds the best basketball team by choosing best players for each position. In order to build the best team, you need to sort out the players from the big chunk of data which is given in a file.

Given:

- **main.org** - reference executable file showing how the program should run.
- **best_team.c** - implement your program of building the best team here.
- **best_team.h** - header file of all the functions that should be completed.
- **Makefile** - to compile the source files to the executable file **main**
- **roster** - given file includes player's information in the format of "**_shirtNum, _age, _pos, _pass, _shoot, _speed, _block, _height, _name**" (without spaces)
- **test** - test script to test if your program generates the expected result. (will be released over the weekend)

➔ Note: for **position**, 0 stands for PG, 1 stands for SG, 2 stands for SF, 3 stands for PF and 4 stands for C.

Recommended workflow:

1. Complete the function **initializeTeam**.
2. Using gdb, see what the function **inputFile** returns (so that you have an idea what to do with it).
3. Now that you see the data returned by **inputFile**, you should be able to complete the function **parsePlayer**.
4. After that, the function **loadData** will populate all the data and store them to **allPlayers** (you will be using linked list for **allPlayers**). Obviously, you will be calling **parsePlayer** and **inputFile** in this function.
5. You can now work on **findBestPlayer**. Iterate through **allPlayers** and look for the best player for the position specified by parameter.
6. The function **buildBestTeam** should call **findBestPlayer** for each position. See the comment on the given code skeleton to see how it's structured.
7. You obviously called malloc a couple of times above, now it's time to clean it up. Call free appropriately on **freePlayer** and **freeTeam**. (Hint: 1 free for 1 malloc, strdup counts as malloc).
8. Your program is expected to run without memory leaks. Run memleak and see the report at the bottom.

Submission:

These are the files that need to be in the Vocareum "work" directory for this assignment:

- best_team.c