Lab9: LinkedList

In this lab, we will build upon the knowledge we learned from previous labs. Additionally, from now on, we will compile our files by creating makefiles similar to the ones we learned about from the previous labs. You will also used the gdb in case you needed to debug your code.

The outcomes from the lab are:

- Dynamic memory allocation, making sure that you correctly allocate and free the memory once you do not need to use it.
- Separating programs into client, interface, and implementation files
- Working with linked-lists
- Practicing makefiles.

Lab details:

In lab7, we created a linked list of books. In this lab, we will create a linked list of members. You do not have to finish lab7 to finish lab9. However, in later labs, you will have to use both linked list. You will have to create the following:

1. A struct member representing a member user to the system. You will need to store the name of the member and the username of the member. The name of the member does not have to be unique but the username must be unique.
2. A linked list representing the list of members in the system.
3. You will create a main program to allow adding a new user through the command line.
4. Through your main program, you will provide log in and log out functionality in your program. You will need to keep track of which members are currently logged in.
5. You might need to edit the provided linked list to add any extra functionalities to make sure your program works correctly. e.g., a function to search for a certain username.

You will need to submit the following:

- README file explaining what the program does and how you compiled it, which command line options you used. Also, since you are allowed to work with a partner, write down your name, your partner name, what you did in the lab, and what the program does as a whole. You are going to submit your lab separately from your partner.
- Submit your program. The file must be fully documented.
    - All programs must include a comment section at the top of the program as outlined below:

```
/*************************************************************************
Program: Author: Date:
Time spent: Purpose:
<name of program>
<your name>
<date you finish the program>
<total amount of time spent on the project>
The purpose of this program is to blah blah blah
*************************************************************************/
```

    - For the function, it must include a comment section giving information about the pre and post conditions

//Precondition:
//Postcondition:

- Submit the makefile used to compile the code
- Submit your main program and the linked list files.
- If valgrind was used, submit also the report showing that you did not face any problems.
- The files must be compressed together and named using the following naming format: lab5-<firtinitialLastName>.zip. For example, lab4-jJohn.zip
- Submit the above compressed folder using the submit command to your lab TA account.