Deliverables

For this assignment you need to complete the implementation of a linked list in C. Your GitHub repository will become your submission, so commit your work as you go. You will also need to upload a screenshot demonstrating the use of the VSCode debugger.

Partners and Groups

You are required to work with your assigned partner on this assignment. You are responsible for ensuring that you and your partner both fully understand all aspects of your submission; you are expected to work collaboratively and take responsibility for each other's learning.

You have also been assigned another group with whom you are expected to meet for planning. Schedule this meeting early in the week and use it to make sure that everyone understands the project's parameters and to sketch a high-level plan for implementation.

After the initial submission, you will also be conducting a peer code review. This will be done with a different group that will be assigned later, but you should keep this code review in mind and make sure you understand all of your code and that it is readable to people outside your group.

Objectives

The goal of this project is to refresh your memory on some of the core topics from the prerequisite classes:

1. CSC221: implementing data structures,
2. CSC250: programming in C, and using the GDB debugger (now in VSCode!).

Implementing data structures can be tricky in C since it's not an object-oriented language (that's why we use Java in 221). You'll need to work with structs and allocate/free memory manually. Working with generic types in C is possible, but inconvenient, and has been omitted from this assignment in favor of storing only integers in the list (we'll definitely see generics in C later).

Linked Lists

Your task is to implement a doubly-linked list (with `prev` and `next` pointers) that is also circular (the last element points back to the head). The file `linked_list.h` defines two structs that make up the linked list data structure and provides the the interface for its methods. You will implement those methods in `linked_list.c`. It is assumed that you know the basics of how a linked list works. Slightly more detailed guidelines are provided in the comments of `linked_list.h`, but it's entirely reasonable that you will have questions about how the data structure is expected to operate. If so, please post them on Slack!

Some basic testing code has been provided in `list_tester.c`, but you should modify it to suit your own purposes. Start by commenting out the tests for things that you haven't implemented yet, and then add more tests of your own to drill down on the issues you're having. You should use GDB (through the VSCode interface) and Valgrind to debug as you go. Your submission should not have any memory leaks (unless your tests deliberately crash the program by attempting invalid operations). Compile with `make` and run with `./list_tester`.

Submitting

You will submit your code by committing to your repository and pushing to GitHub. Remember that you will have a code review with another group next week, so be sure to clean up your code and add comments to ensure that they (and you!) can easily understand how it works.

In addition, you should take a screenshot proving that you have set up the VSCode debugger and used it to set a breakpoint and inspect the execution of your program. This screenshot should be renamed to make clear what it is. It can then be added, committed, and pushed to your Git repository just like any other file.