

# CSE 240 Spring 2021 Homework 6, Linked Lists (50 points)

Due Saturday, February 27, 2021 at 11:59PM, plus a 24-Hour grace period

---

## Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures, including enumeration type, linked list, pointer operations, memory leak, parameter passing mechanisms, and recursion. By the end of the assignment, you should have understood and exercised

- Enumeration type and its operations
- Pointer operations accessing structured data
- Linked list of structures, with complex manipulations of pointer and structure
- Memory management and leak prevention
- Parameter passing and return values of different types.
- Recursive program.

**Reading:** Chapter 2, Textbook Section 2.5.4 on linked list and Section 2.6 on parameter passing; Section 2.10 a case study on linked list operations; and sample code in lecture slides.

**Preparation:** Complete the multiple choice questions in the textbook exercise section. The answer keys can be found in the course Web site. These exercises can help you prepare for your weekly quiz and the exam. You are encouraged to read the other exercise questions and make sure you understand these questions in the textbook exercise section, which can help you better understand what materials are expected to understand after the lectures and homework on each chapter.

You are expected to do the majority of the assignment outside the class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or the TA during their office hours.

You are encouraged to ask and answer questions on the course discussion board. However, **do not share your answers and code** in the course discussion board.

## Programming Assignment (50 points)

1. Based on the previous homework, you should implement hw06q1.c **from scratch**. The structure of this homework is similar to previous homework. In this homework, to do the same work in the previous homework, you must use **a linked list** instead of an array.
2. You should use a struct 'libraryRecord' with elements for Book Title, Author's Name, Book Type, Book Id, Aisle number in which the book should be kept. You will be completing a program that creates a list of books (a linked list of structs). It is a menu-driven program where the user is given the following options:

- a) Add a new book to the list. When adding a new book to the list, the user is prompted for Book Title, Author Name, Book Type, Book Id and Aisle number of the book. The book should be added at the end of the list. If the book (name or ID) already exists in the list, then you should not add to the list. The book type is an enum type.
- b) Display the list of books from the linked list.
- c) Sort the list of books numerically by ID. The sorting should happen within the list. You should not create a new list (linked list of structs) of books having sorted books.
- d) Delete a book from the list. When deleting a book from the list, the user is prompted for Book ID only. The book should be removed from the list and the linked list structure should be preserved / restored afterwards. Attempting to delete a book who is not in the list should return 0. Otherwise, return 1 upon a successful removal of the record.

The behavior of save() in this assignment is the same as that of the previous homework.

However, you should modify it to new save() that handles a linked list instead of an array.

Please refer to the specification of save() in the previous homework. You need to implement load() which is called at the start of the program. This function will read the saved file and fill in the linked list of structures. You need to read the file in the same order and manner that it is saved in save().

Expected output of each function:

**Add (10 points):**

```
Book_List.txt not found.

Enter your selection:
    a: add a new book
    d: display book list
    r: remove a book from list
    s: sort book list by ID
    q: quit
a

Enter book title: Sapiens
Enter author name: Yuval
Enter book type (Fiction/ Nonfiction): Nonfiction
Please enter book ID number: 1
Please enter aisle number: 1

Book successfully added to the list!
```

**Display (10 points):**

(after adding 3 book details)

```
Enter your selection:
    a: add a new book
    d: display book list
    r: remove a book from list
    s: sort book list by ID
    q: quit
```

```
d
```

```
Book Title: Sapiens
Author Name: Yuval
Book Type: Nonfiction
Book ID: 1
Aisle number: 1
```

```
Book Title: A Promised Land
Author Name: Barack Obama
Book Type: Nonfiction
Book ID: 3
Aisle number: 3
```

```
Book Title: The Alchemist
Author Name: Paulo
Book Type: Fiction
Book ID: 2
Aisle number: 2
```

**Sort (10 points):**

```
Enter your selection:
    a: add a new book
    d: display book list
    r: remove a book from list
    s: sort book list by ID
    q: quit
s

Book list sorted! Use display option 'd' to view sorted list.

Enter your selection:
    a: add a new book
    d: display book list
    r: remove a book from list
    s: sort book list by ID
    q: quit
d

Book Title: Sapiens
Author Name: Yuval
Book Type: Nonfiction
Book ID: 1
Aisle number: 1

Book Title: The Alchemist
Author Name: Paulo
Book Type: Fiction
Book ID: 2
Aisle number: 2

Book Title: A Promised Land
Author Name: Barack Obama
Book Type: Nonfiction
Book ID: 3
Aisle number: 3
```

The books seen in display() output above are sorted in sort(). Use 'd' option to verify sorted result.

Delete (10 points):

```
Enter your selection:
    a: add a new book
    d: display book list
    r: remove a book from list
    s: sort book list by ID
    q: quit
r
Please enter ID number of book to be deleted: 2

Book deleted successfully!

Enter your selection:
    a: add a new book
    d: display book list
    r: remove a book from list
    s: sort book list by ID
    q: quit
d

Book Title: Sapiens
Author Name: Yuval
Book Type: Nonfiction
Book ID: 1
Aisle number: 1

Book Title: A Promised Land
Author Name: Barack Obama
Book Type: Nonfiction
Book ID: 3
Aisle number: 3
```

To verify that delete() worked as expected, use 'd' display option to display updated list.

Load (10 points):

```
Books record loaded from Book_List.txt.
```

```
Enter your selection:
```

```
    a: add a new book  
    d: display book list  
    r: remove a book from list  
    s: sort book list by ID  
    q: quit
```

```
d
```

```
Book Title: Sapiens  
Author Name: Yuval  
Book Type: Nonfiction  
Book ID: 1  
Aisle number: 1
```

```
Book Title: A Promised Land  
Author Name: Barack Obama  
Book Type: Nonfiction  
Book ID: 3  
Aisle number: 3
```

Notice the message given by load() "Books record loaded from Book\_List.txt" at the top. To verify that load() worked as expected, use 'd' display option to display loaded list.

## Grading and Rubrics

Each sub-question (programming tasks) has been assigned certain points. We will grade your programs following these steps:

(1) Compile the code. If it does not compile, 50% of the points given for the code under compilation will be deducted. Then, we will read the code and give points between 50% and 0, as shown in right part of the rubric table.

(2) If the code passes the compilation, we will execute and test the code using test cases. We will assign points based on the left part of the rubric table.

In both cases (passing compilation and failed compilation), we will read your program and give points based on the points allocated to each sub-question, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

Please notice that we will not debug your program to figure out how big or how small the error is. You may lose 50% of your points for a small error such missing a comma or a space!

We will apply the following rubrics to **each sub-question** listed in the assignment. Assume that points assigned to a sub-question is *pts*:

Rubric Table

Major	Code passed compilation				Code failed compilation		
Points	pts * 100%	pts * 90%	pts * 80%	pts * 70% - 60%	pts * 50% - 40%	pts * 30% - 10%	0
Each sub-question	Meeting all requirements, well commented, and working correctly in all test cases	Working correctly in all test cases. Comments not provided to explain what each part of code does.	Working with minor problem, such as not writing comments, code not working in certain uncommon boundary conditions.	Working in most test cases, but with major problem, such as the code fail a common test case	Failed compilation or not working correctly, but showing serious effort in addressing the problem.	Failed compilation, showing some effort, but the code does not implement the required work.	No attempt

## What to Submit?

This homework assignment will have multiple parts. You are required to submit your solutions in a compressed format (.zip). Make sure your compressed file is label correctly - lastname\_firstname6.zip. (All lowercase, do not put anything else in the name like "hw2".)

The compressed file MUST contain the following:

hw06q1.c

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded. (Even if the assignment is submitted late.)

**Submission preparation notice:** The assignment consists of multiple files. You must copy these files into a single folder for canvas submission. To make sure that you have all the files included in the zip file and they work after unzip operation, you must test them before submission. You must also download your own submission from the canvas. Unzip the file on a different machine, and test your assignment and see if you can open and test the files in a different location, because the TA will test your application on a different machine. If you submitted an empty project folder, an incomplete project folder, or a wrong folder, you cannot resubmit after the submission linked is closed! We grade only what you submitted in the canvas. We cannot grade the assignment on your computer or any other storage, even if the modification date indicated that the files were created before the submission due dates. The canvas submission may take a few minutes. Be patient and wait for it to complete.

## **Where to Submit?**

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

## **Late submission deduction policy**

- No penalty for late submissions that are received within 24 hours after the deadline (before Sunday midnight);
- 10% grade deduction for every day it is late after the grace period (After Sunday);
- No late submission after Tuesday at 11:59PM.

## **Academic Integrity and Honor Code.**

You are encouraged to cooperate in study group on learning the course materials. However, you may not cooperate on preparing the individual assignments. Anything that you turn in must be your own work: You must write up your own solution with your own understanding. If you use an idea that is found in a book or from other sources, or that was developed by someone else or jointly with some group, make sure you acknowledge the source and/or the names of the persons in the write-up for each problem. When you help your peers, you should never show your work to them. All assignment questions must be asked in the course discussion board. Asking assignment questions or making your assignment available in public websites before the assignment is due will be considered cheating.